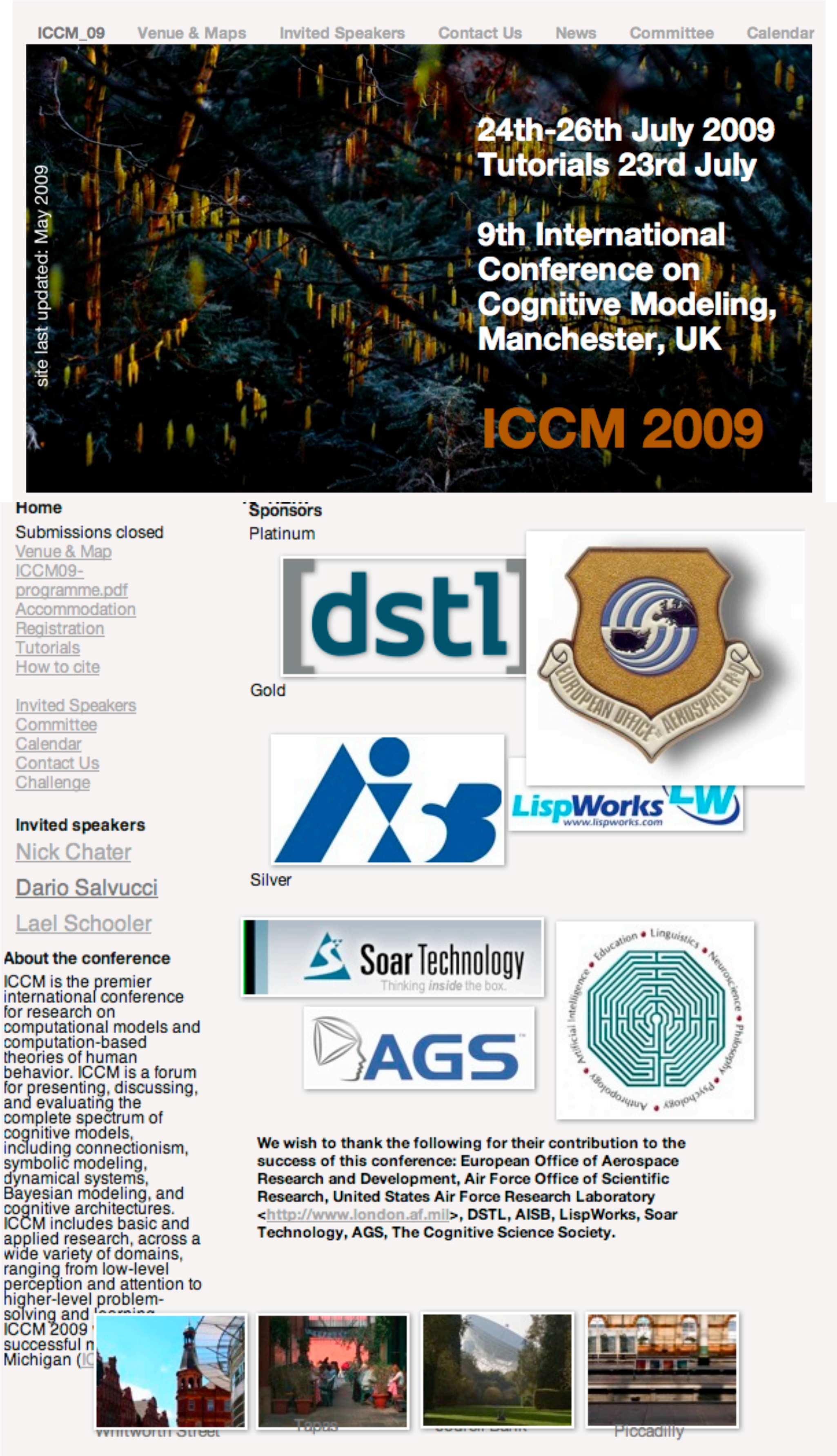


ICCM_09 Venue & Maps Invited Speakers Contact Us News Committee Calendar

site last updated: May 2009

24th-26th July 2009
Tutorials 23rd July

9th International Conference on Cognitive Modeling, Manchester, UK

ICCM 2009

Home

Submissions closed
Venue & Map
ICCM09-programme.pdf
Accommodation
Registration
Tutorials
How to cite

Invited Speakers
Committee
Calendar
Contact Us
Challenge

Invited speakers

Nick Chater

Dario Salvucci

Lael Schooler

About the conference

ICCM is the premier international conference for research on computational models and computation-based theories of human behavior. ICCM is a forum for presenting, discussing, and evaluating the complete spectrum of cognitive models, including connectionism, symbolic modeling, dynamical systems, Bayesian modeling, and cognitive architectures. ICCM includes basic and applied research, across a wide variety of domains, ranging from low-level perception and attention to higher-level problem-solving and learning.
ICCM 2009
successful m
Michigan (IC

Sponsors

Platinum

Gold

Silver

We wish to thank the following for their contribution to the success of this conference: European Office of Aerospace Research and Development, Air Force Office of Scientific Research, United States Air Force Research Laboratory <http://www.london.af.mil>, DSTL, AISB, LispWorks, Soar Technology, AGS, The Cognitive Science Society.

[dstl]
EUROPEAN OFFICE OF AEROSPACE R&D
LispWorks
www.lispworks.com
Soar Technology
Thinking inside the box.
AGS
Education • Linguistics • Neuroscience • Philosophy • Psychology • Anthropology • Artificial Intelligence
Whitworth Street
Tapas
Jodrell Bank
Piccadilly

Welcome!

Welcome to Manchester and the 9th International Conference on Cognitive Modeling. We hope you find the next three days enjoyable and stimulating. This year's conference attracted a large number of high quality submissions on a wide range of topics, making the task of categorising submissions into talks and posters extremely difficult for the program committee. After two days of intense deliberation, we settled on a program of 24 talks and 73 posters. However there was a class of submissions that we thought were of sufficient quality that, if the schedule had permitted, could also have been accepted as talks. To acknowledge this fact we have decided to allocate time before the two poster sessions for the authors of these distinguished posters to speak briefly and introduce their work.

In addition to the talk and poster sessions, this year's conference also features invited talks by three leading figures in the cognitive modelling community: Nick Chater, Dario Salvucci, and Lael Schooler, and two symposia. As usual, the conference is preceded by a number of tutorials on different approaches to cognitive modeling. All in all, we believe that this year's conference presents a strong program of research that reflects a growing, vibrant international cognitive modelling community. We hope that after participating in the conference you agree with this assessment.

The efforts of many people have gone into producing this conference. The chairs wish to thank the three invited speakers, Frank Ritter for organising the tutorials, the many reviewers for their invaluable comments on the submissions and the universities of Manchester, Huddersfield and Birkbeck, University of London for their financial and administrative support. In addition, we wish to thank the following for their contribution to the success of this conference:

Platinum Sponsors

Defence Science and Technology Laboratory

European Office of Aerospace Research and Development, Air Force Office of Scientific Research, United States Air Force Research Laboratory

Gold Sponsors

The Society for the Study of Artificial Intelligence and Simulation of Behaviour

LispWorks

Silver Sponsors

The Cognitive Science Society

Soar Technology

AGS TechNet

Andrew Howes, David Peebles and Rick Cooper

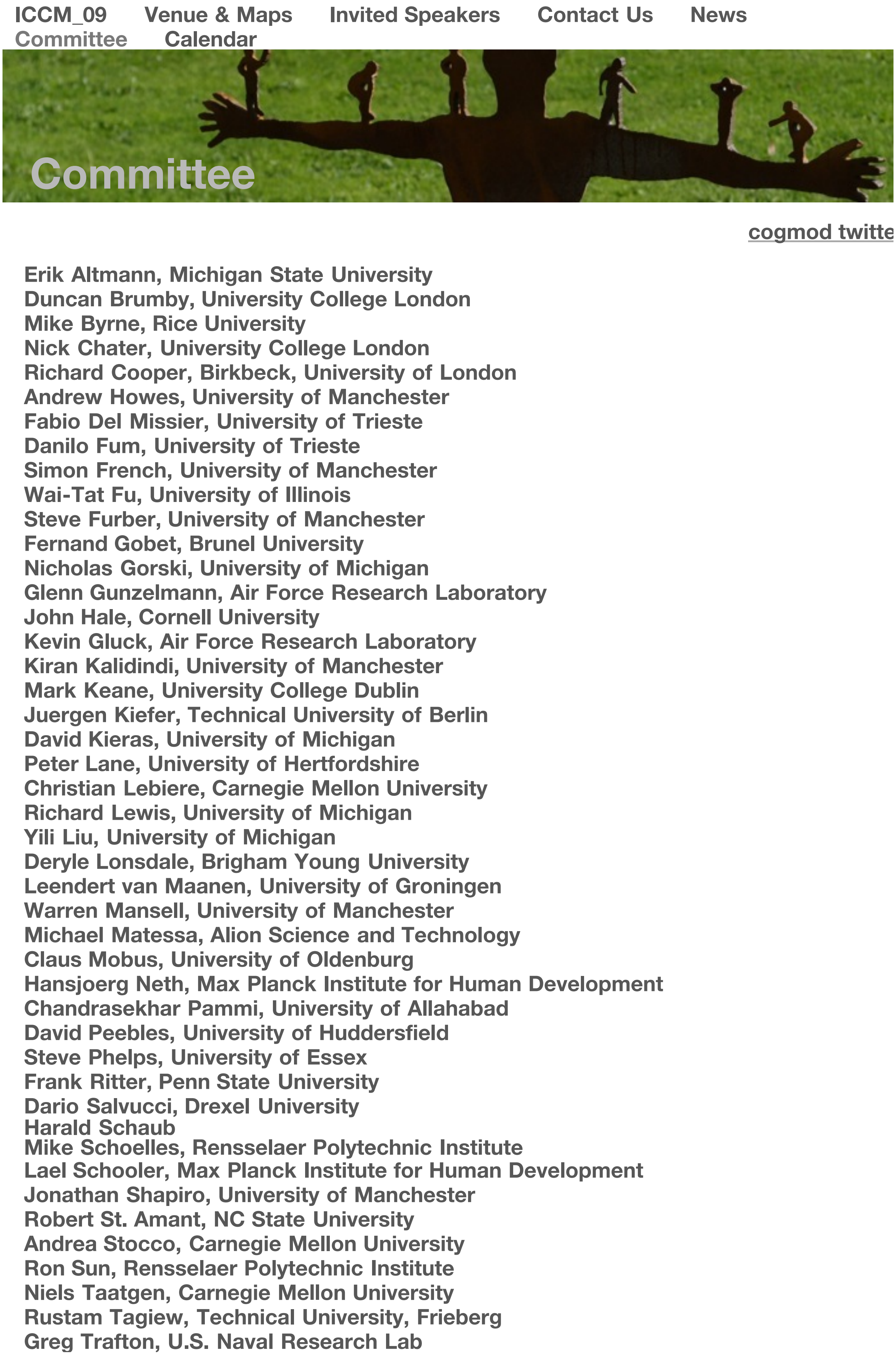
We are grateful to the following reviewers:

Erik Altmann
Ben Ambridge
Roman Belavkin
Jens Binder
Tibor Bosse
Henry Brighton
Duncan Brumby
Mike Byrne
Yun-Gyung Cheong
Morten Christiansen
Rick Cooper
Saskia van Dantzig
Eddy Davelaar
Fabio Del Missier
Julien Diard
Scott Douglass
Chris Eliasmith
Simon French
Wai-Tat Fu
Danilo Fum
Steve Furber
Gyslain Giguère
Kevin Gluck
Fernand Gobet
Nicholas Gorski
Wayne Gray
Alessandro Grecucci
Collin Green
Markus Guhe
Glenn Gunzelmann
John Hale
Anthony Harrison
Andrew Howes
Christian Huyck
Sudarshan Iyengar
Tiffany Jastrzembski
Gary Jones
Kiran Kalidindi
Mark Keane
Richard Kennaway
William Kennedy
David Kieras
Adam Krawitz
Peter Lane
Christian Lebiere
Frank Lee
Stelios Lelis
Sandro Leuchter
Rick Lewis
Yili Liu
Deryle Lonsdale
Leendert van Maanen
Warren Mansell
Michael Matessa
Ralf Mayrhofer
Stuart Mcgregor
John-Jules Meyer
Gareth Miles
Claus Moebus
Christopher Myers
Hansjoerg Neth
Tom Ormerod
Giovanni Pagliuca
V. S. Chandrasekhar Pammi
David Peebles
Steve Phelps
Julian Pine
Bill Powers
David Reitter
Hedderik van Rijn
Frank Ritter
Paul Rosenbloom
Elena Sakkalou
Dario Salvucci
Harald Schaub
Mike Schoelles
Lael Schooler
Holger Schultheis
Chris Schunn
Angela Schwering
J. Ignacio Serrano
Jonathan Shapiro
Patrick Simen
Robert St. Amant
Terry Stewart
Mark Steyvers
Andrea Stocco
Ron Sun
Niels Taatgen
Rustam Tagiew
Greg Trafton
Shravan Vasishth
Victor Vickland
Hongbin Wang
Gert Westermann
Janet Wiles
Richard Young
Shunan Zhang

Return to the previous page

Nicholas Gorski Yili Liu Harald Schaub

Return to the previous page



Committee

Erik Altmann, Michigan State University
Duncan Brumby, University College London
Mike Byrne, Rice University
Nick Chater, University College London
Richard Cooper, Birkbeck, University of London
Andrew Howes, University of Manchester
Fabio Del Missier, University of Trieste
Danilo Fum, University of Trieste
Simon French, University of Manchester
Wai-Tat Fu, University of Illinois
Steve Furber, University of Manchester
Fernand Gobet, Brunel University
Nicholas Gorski, University of Michigan
Glenn Gunzelmann, Air Force Research Laboratory
John Hale, Cornell University
Kevin Gluck, Air Force Research Laboratory
Kiran Kalidindi, University of Manchester
Mark Keane, University College Dublin
Juergen Kiefer, Technical University of Berlin
David Kieras, University of Michigan
Peter Lane, University of Hertfordshire
Christian Lebiere, Carnegie Mellon University
Richard Lewis, University of Michigan
Yili Liu, University of Michigan
Deryle Lonsdale, Brigham Young University
Leendert van Maanen, University of Groningen
Warren Mansell, University of Manchester
Michael Matessa, Alion Science and Technology
Claus Mobus, University of Oldenburg
Hansjoerg Neth, Max Planck Institute for Human Development
Chandrasekhar Pammi, University of Allahabad
David Peebles, University of Huddersfield
Steve Phelps, University of Essex
Frank Ritter, Penn State University
Dario Salvucci, Drexel University
Harald Schaub
Mike Schoelles, Rensselaer Polytechnic Institute
Lael Schooler, Max Planck Institute for Human Development
Jonathan Shapiro, University of Manchester
Robert St. Amant, NC State University
Andrea Stocco, Carnegie Mellon University
Ron Sun, Rensselaer Polytechnic Institute
Niels Taatgen, Carnegie Mellon University
Rustam Tagiew, Technical University, Frieberg
Greg Trafton, U.S. Naval Research Lab

Greg Trafton, U.S. Naval Research Lab
Hedderick van Rijn, University of Groningen
Shravan Vasishth, University of Potsdam
Richard M. Young, University Collegel London
Travis Seymour, UC Santa Cruz

Table of Contents

Pre-conference Tutorials

Symposia

Papers

Featured Posters

Posters

Poster Abstracts

Agent-Based Simulation: Social Simulation and Beyond

Bruce Edmonds (bruce@cfpm.org)
Emma Norling (norling@acm.org)
Centre for Policy Modelling
Manchester Metropolitan University
Aytoun Building, Aytoun St, Manchester, M1 3GH, United Kingdom

Abstract

Our interest in agent-based simulation is for *social simulation*, where the society-level outcomes emerge from the interaction of individuals. In this tutorial, we aim to introduce the core concepts of agent-based social simulation, illustrated by a range of examples, before walking through a specific example with the participants so that they can experience these issues at first hand.

Keywords: social simulation, social theory, agent-based modelling

What is Agent-Based Simulation?

Agent-based simulation (ABS) represents each individual with a separate encapsulated object in a simulation. Beyond this, the definition of an "agent" varies quite widely, but in general they are seen to be autonomous, pro-active entities. Simulation outcomes emerge from the interactions between these entities, and often even quite simple interactions can give rise to complex system dynamics.

The individuals that agents represent in a simulation need not be humans, and could be social actors of any type. Examples of entities that have been represented by agents in simulations range from individual cells and bacteria through to multi-national corporations. Typically though in social simulation we are interested in modelling each individual person with a single agent. At the same time, we are often interested in modelling the interactions of large numbers of individuals, and this forces a trade off between the detail of the individual models and the number of entities that can be modelled.

Thus, while it is desirable for the agents to include models of various aspects of cognition (such as decision making, learning, belief representation, autonomous goals), it is necessary to pare them down to the bare minimum required to model the social interactions of interest. By the standards of cognitive models many of the programs internal to each agent might be fairly simple, although some researchers in this area are investigating ways of including more detailed models of individuals within this type of simulation.

Simulating Societies

Our interest in ABS is to simulate how humans (or other social entities) might interact: for example, how complex coordination might be achieved through the interaction of essentially selfish agents (Edmonds, 2006). Some of these models can be very detailed, including many different aspects of a particular observed social situation. In this case the result is more like a dynamic description within a simulation – a distributed representation that may incorporate many different kinds of evidence.

Emergent Behaviour

At the same time, complexity science has repeatedly shown how the interaction of fairly simple agents can result in complex ("emergent") outcomes. Thus, one stream of research in ABS is looking at how social systems might be understood in this way. These tend to be quite abstract simulations with very simple agents, which are intended to encapsulate a general social theory, rather than to represent any particular observed social phenomena.

Applying Social "Rules" to Other Networks

One outcome of the study of emergent behaviour in human societies has been to transfer these principles to other social systems. For example, when systems of independently programmed computers interact in a network, many of the same issues (trust, reputation, coordination etc.) that occur in human societies are found to be important. The previously mentioned work on cooperation between self-interested individuals, for example, has been used to develop algorithms for peer-to-peer computing systems that are robust against "cheaters" (Hales, 2006).

Outline of the Tutorial

This tutorial introduces the main ideas of ABS, highlighting the difficulties as well as the strength of these issues, drawing on many examples of ABS, from complex specific simulations, up to highly abstract simulations that encapsulate social theories. In the second half of the tutorial, these ideas will be illustrated through the use of a concrete example. Depending on the existing skills of the participants, there will be opportunities to implement their own realisation of this example.

Acknowledgements

Some material in this tutorial is based upon a tutorial presented by Franziska Klügl and Emma Norling at the European Agent Systems Summer School, 2006.

References

Edmonds, B (2006). The Emergence of Symbiotic Groups Resulting from Skill-Differentiation and Tags. *Journal of Artificial Societies and Social Simulation* 9(1)10 <http://jasss.soc.surrey.ac.uk/9/1/10.html>.

Hales, D. (2006) Emergent Group-Level Selection in a Peer-to-Peer Network. *Complexus* 2006;3:108-118

Cognitive Modelling with the Neural Engineering Framework

Chris Eliasmith (celiasmith@uwaterloo.ca)
Terrence C. Stewart (tcstewar@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Abstract

The Neural Engineering Framework (NEF; Eliasmith and Anderson, 2003) provides a general methodology for developing efficient and realistic neural models that perform a specified task. The framework consists of three quantified principles, one for each of representation, computation, and dynamics in neural systems. Adopting these principles provides a method for generating connection weights between groups of neurons that represent and transform state variables. In short, the NEF provides a neural compiler: a method for taking a high-level description of a neural system and deriving a plausible organization of realistic neurons that realize this system. Our tutorial introduces the principles of the NEF and demonstrates how they apply to cognitive modeling. This is done through the use of Nengo, a GUI neural simulation system, which supports an adjustable level of neural accuracy, Python scripting, and the analysis of the resulting models.

Keywords: Nengo; neural engineering; neural representation; control theory; neural cognitive modelling

The Neural Engineering Framework

As cognitive models become more complex, there is an increased demand for details at both the low and high levels. Traditionally, focus in cognitive modeling has been on higher levels of abstraction. As a result, researchers typically posit a high-level organizational structure which allows them to consider the information that needs to be represented and the transformations that are required for implementing hypothesized algorithms. Ideally, however, a cognitive model should also make detailed predictions as to the firing rates of neurons implementing the model, their tuning curves, connectivity, neurotransmitters, and other properties.

The Neural Engineering Framework (or NEF; Eliasmith and Anderson, 2003) provides a novel approach to addressing this typical gap in cognitive modeling. It is based on three principles of neural engineering:

1. Neural representations are defined by the combination of nonlinear encoding (exemplified by neuron tuning curves) and weighted linear decoding.
2. Transformations of neural representations are functions of variables that are represented by neural populations. Transformations are determined using an alternately weighted linear decoding.
3. Neural dynamics are characterized by considering neural representations as control theoretic state variables. Thus, the dynamics of neurobiological systems can be analyzed using control theory.

Each of these principles is considered under the assumption that neural systems are subject to significant amounts of noise. Therefore, any analysis of such systems must account for the effects of noise.

The core idea of the NEF is to consider any cognitive system as containing a large number of representations which change over time. How these representations change is dependent both on the external stimuli and on the other representations within the system. A particular neural group can represent (via its spike pattern) a single scalar, a vector, or even a function. These representations are inherently noisy, and accuracy will be dependent on various neural properties (although representational error has been shown to be inversely linearly related to the number of neurons used).

To understand how these representations change (i.e. define a transformation of a representation), the NEF provides methods for defining weighted axonal projections. For instance, a given group might represent the product of the values being represented by two other groups which are projected to it (i.e. x(t) = y(t)*z(t), where each variable is represented by a neural population). Importantly, we can use the NEF to derive the linearly optimal connection weights to perform a wide variety of linear and nonlinear transformations. Doing so makes it clear that the accuracy of these transformations is intimately related to the observable tuning curves of the neurons involved. This leads to models that are orders of magnitude more efficient than other approaches to neural representation, and which are a closer match to observed neurological data (e.g. Conklin & Eliasmith, 2005; Fischer, 2005).

Applications

Initially, the main applications of this approach were in the domains of sensory and motor systems. This has included the barn owl auditory system (Fischer, 2005), rodent navigation (Conklin & Eliasmith, 2005), escape and swimming control in zebrafish (Kuo & Eliasmith, 2005), and the translational vestibular ocular reflex in monkeys (Eliasmith et al., 2002). However, these same principles are now being applied to higher-level cognitive models. A direct extension of the visual working memory model (Singh & Eliasmith, 2006) has led to a neural model of the ACT-R goal buffer (Stewart, Tripp, & Eliasmith, 2008). More crucially, the use of Vector Symbolic Architectures (Gayler, 2003) has allowed for the representation and manipulation of structured symbol trees by these neural models. This neurally realistic cognitive architecture (Stewart & Eliasmith, 2009a) resulted in a model of the Wason card task (Eliasmith, 2005) and ongoing work producing an efficient production system using realistic neural constraints (Stewart & Eliasmith, 2008; 2009b).

The NEF provides an exciting new tool for cognitive modelers as it provides a technique for producing direct neural predictions from a given high-level algorithmic description of a cognitive model. Furthermore, it leads to important theoretical results as to the relationships between neural properties and the high-level algorithms they are capable of implementing (e.g. the relationship between neurotransmitter re-uptake rate and the time constant of neural transformations).

These consequences are also very general, as the NEF provides techniques that can be applied to any cognitive model. It provides a structure for organizing the high-level description of a model, such that it can be implemented by realistic spiking neurons, providing meaningful data in terms of the expected spike patterns, time course, and accuracy. We have made use of it in a wide variety of contexts, and we have developed tools that support the creation and analysis of these models. These tools can be applied to many existing models to incorporate low-level neural details into existing modeling research.

Software and Simulations

We have developed Nengo <nengo.ca>, a freely available open-souce Java-based neural simulator that supports the NEF. This allows for hand-on examples of the theoretical concepts underlying the NEF. Using a point-and-click interface, we can create neural group, configure them to represent scalars and vectors, adjust their neural properties, and simulate their spiking activity over time. We can also connect these neural groups via synapses so as to perform linear and nonlinear transformations on these values, and store information over time. These are the basic mechanisms required for a wide range of algorithms, and form the basis for our models of sensorimotor systems and working memory. Nengo can also be programmed using a Python interface, allowing for quick creation of complex models (Stewart, Tripp, & Eliasmith, 2009).

Furthermore, these basic tools can be used to implement the theory of Vector Symbolic Architectures (Gayler, 2003) using NEF. This involves using high-dimensional fixed-length vectors to represent symbols and symbol trees. The nonlinear operation of circular convolution is used to manipulate these symbol trees. This can be seen as a non-classical symbol system, capable of performing the operations required for symbolic cognition. The result is a scalable and efficient neural cognitive architecture, constructed from these basic neural components.

References

Conklin, J. & Eliasmith, C. (2005). An attractor network model of path integration in the rat. *Journal of Computational Neuroscience 18*, 183-203.

Eliasmith, C. (2005). *Cognition with neurons: A large-scale, biologically realistic model of the Wason task.* 27th Annual Meeting of the Cognitive Science Society.

Eliasmith, C. & Anderson, C. (2003). *Neural Engineering: Computation, representation, and dynamics in neurobiological systems.* Cambridge: MIT Press.

Eliasmith, C., Westover, M.B., & Anderson, C.H. (2002). A general framework for neurobiological modeling: An application to the vestibular system. *Neurocomputing 46*, 1071-1076.

Fischer, B. (2005). *A model of the computations leading to a representation of auditory space in the midbrain of the barn owl*. PhD thesis. Washington University in St. Louis.

Gayler, R. (2003). *Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience.* ICCS/ASCS International Conference on Cognitive Science, 133-138.

Kuo, D. & Eliasmith, C. (2005). Integrating behavioral and neural data in a model of zebrafish network interaction. *Biological Cybernetics*. *93*(3), 178-187.

Singh, R., & Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *Journal of Neuroscience*, *26*, 3667-3678.

Stewart, T.C. & Eliasmith, C. (2008). *Building production systems with realistic spiking neurons.* 30th Annual Meeting of the Cognitive Science Society.

Stewart, T. C. & Eliasmith, C. (2009a). *Compositionality and biologically plausible models*. In Oxford Handbook of Compositionality, W. Hinzen, E. Machery, M. Werning, eds. (Oxford University Press).

Stewart, T. C., & Eliasmith, C. (2009b). *Spiking neurons and central executive control: The origin of the 50-millisecond cognitive cycle*. 9th International Conference on Cognitive Modelling.

Stewart, T. C., Tripp, B., & Eliasmith, C. (2008). *Supplementing Neural Modelling with ACT-R*. 15th Annual ACT-R Workshop.

Stewart, T. C., B. Tripp, & Eliasmith, C. (2009). Python scripting in the Nengo simulator. *Frontiers in Neuroinformatics. 3*(7), 1-9.

EPAM/CHREST Tutorial: Fifty Years of Simulating Human Learning

Peter C. R. Lane (peter.lane@bcs.org.uk)
School of Computer Science, University of Hertfordshire, College Lane, Hatfield AL10 9AB, UK

Fernand Gobet (fernand.gobet@brunel.ac.uk)
School of Social Sciences, Brunel University, Cleveland Road, Uxbridge, Middlesex, UB8 3PH, UK

Keywords: CHREST; EPAM; cognitive modelling; chunking

Overview

This tutorial covers a tradition of symbolic computational modelling known as EPAM/CHREST, with its first member, EPAM (Elementary Perceiver and Memoriser) developed by Edward Feigenbaum in 1959. EPAM was used to construct models of a variety of phenomena, providing the impetus to develop the chunking theory (Chase & Simon, 1973; Gobet et al., 2001), which has been an important component of theories of human cognition ever since.

The history of computational modelling includes a variety of approaches to describe human behaviour. The benefits of encoding a theory as a computational model include a precise definition of how the behaviour is to be explained, and a means of generating quantitative predictions for testing the theory. Examples include models of single phenomena (such as Sternberg's model of STM; (Sternberg, 1966)), integrated models covering a wide range of different phenomena (such as Soar (Newell, 1990) and ACT-R (Anderson & Lebière, 1998)), and over-arching principles, which guide the development of models in disparate domains (such as connectionist approaches (McLeod, Plunkett, & Rolls, 1998), or embodied cognition (Pfeifer & Scheier, 1999)).

The group of models to be studied in this tutorial emphasise learning phenomena, and learning at a symbolic level. EPAM was the precursor of the later CHREST (Chunk Hierarchy and REtrieval STructures) system, and both are typically developed from large quantities of naturalistic input. For example, in modelling expert perception of chess players, actual chess games are used (Gobet & Simon, 2000). Similarly, in modelling the acquisition of syntax, large corpora of mother-child interactions are employed to develop the model's long-term memory (Freudenthal, Pine, Aguado-Orea, & Gobet, 2007).

The tutorial is structured so that participants will:

1. Acquire a complete understanding of the EPAM and CHREST approach to computational modelling, and their relation to the chunking and template theories of cognition;
2. Explore some key learning phenomena supporting the chunking theory, based around experiments in verbal-learning, categorisation and the acquisition of expertise;
3. Be introduced to an implementation of CHREST which can be used for constructing models of their own data.

Further information about CHREST, supporting publications and implementations can be found at: `http://chrest.info`

Chunking and Template Theories

A *chunk* is a 'familiar pattern', an item stored in long-term memory. Chunks collect together more basic elements which have strong associations with each other, but weak associations with other elements (Chase & Simon, 1973; Cowan, 2001). Miller observed (Miller, 1956) that short-term memory typically contains a limited number of pieces of information, but the size of these pieces varies with context; this observation lies behind the chunking theory. Chase and Simon (1973) confirmed the presence of chunks in the recall of chess positions, and the EPAM model provides a means of learning, storing and retrieving such chunks.

The chunking theory has been extended to form the *template theory* (Gobet & Simon, 1996, 2000). The extensions include mechanisms to create retrieval structures, which use specific retrieval cues to store and obtain information rapidly. The template is a form of slotted schema, containing a *core*, of stable information, and *slots*, containing variable information. Where the chunking theory captures much of how the average person learns in tasks such as verbal-learning, the template theory further captures the way in which highly-trained human experts perceive and identify patterns in their domain of expertise.

A more detailed overview of the chunking and template theories is contained in Gobet et al. (2001).

Implementation

CHREST comprises three basic modules:

- Input/output module, which is responsible for feature extraction, passing the features to the long-term memory for sorting, and guiding the eye movements;
- Long-term memory, which holds information in the form a discrimination network; and
- Short-term memories, which hold pointers to nodes in the long-term memory.

The key feature which distinguishes EPAM/CHREST models is the discrimination network for storing and retrieving information in long-term memory. Information input to the models is assumed to form a list of subobjects, each of which is either a further list of subobjects or else a primitive. Once information has been stored within the network, it becomes a *chunk*, a 'familiar pattern'. Tests in the discrimination network check for the presence of individual primitive objects, or known chunks (which can be large lists of subobjects). The discrimination network is trained by exposing

CHREST to a large set of naturalistic data. A typical network for an expert in a complex domain will contain on the order of 100,000 nodes.

CHREST extends on EPAM by collecting chunks together when an internal node meets specific criteria relating to its connections with other nodes within memory. A template is then formed from the common information in the linked chunks, with slots created for the variable information. Just as EPAM was the computational embodiment of key aspects of the chunking theory, CHREST implements essential aspects of the template theory.

Input can be provided to CHREST in one of two ways. As a single pattern, which is provided in 'one go'. These patterns are input to the network and stored directly. The second way is to use the in-built attentional mechanism, by which CHREST scans an input array, such as a chess board, and stores parts of the input array into memory. Short-term memory will then hold a set of chunks, each of which may hold information about a different part of the chess board, and collectively holding information about most of the board. The attention mechanism in CHREST is described in Lane, Gobet, and Ll. Smith (2009).

CHREST is implemented in Lisp, and uses Tk to provide a graphical interface. A graphical environment enables users to create simple CHREST models by providing data within an input data file. The implementation also supports more complex tailored models which may be developed by writing special-purpose code using the packages within CHREST. Within the tutorial we will introduce participants to the graphical environment, walk them through a number of provided examples which will illustrate the workings of the architecture and some samples of successful applications, and finally describe the input data format for applying the environment to new domains. A library and manual is provided to assist users wishing to write more complex models.

Applications

The tutorial will cover a variety of experimental data to illustrate the theory and processes. We begin with human verbal-learning processes, which were behind the development of the first EPAM learning system. The interlinked learning operations, which alternately extend or elaborate information in the netwrok, are illustrated using applications in verbal learning (Feigenbaum, 1959; Feigenbaum & Simon, 1984). Further properties of the chunking network will be described with reference to results from categorisation (Gobet, Richman, Staszewski, & Simon, 1997), implicit learning and language learning (Freudenthal et al., 2007; Jones, Gobet, & Pine, 2007).

More elaborate models of expertise explore the interaction between the learner and its external environment. We illustrate this aspect of the theory with models of chess expertise, and in particular look at the recall task, which can reveal many details of expert memory. This application is used to describe CHREST's attention mechanisms (Lane et al., 2009) and how they relate to training the discrimination network.

References

Anderson, J. R., & Lebière, C. (Eds.). (1998). *The atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology, 4*, 55-81.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences, 24*, 87–114.

Feigenbaum, E. A. (1959). *An information processing theory of verbal learning.* The RAND Corporation Mathematics Division, P-1817.

Feigenbaum, E. A., & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science, 8*, 305–336.

Freudenthal, D., Pine, J. M., Aguado-Orea, J., & Gobet, F. (2007). Modelling the developmental patterning of finiteness marking in English, Dutch, German and Spanish using MOSAIC. *Cognitive Science, 31*, 311–341.

Gobet, F., Lane, P. C. R., Croker, S. J., Cheng, P. C.-H., Jones, G., Oliver, I., et al. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences, 5*, 236–243.

Gobet, F., Richman, H., Staszewski, J., & Simon, H. A. (1997). Goals, representations, and strategies in a concept attainment task: The EPAM model. *The Psychology of Learning and Motivation, 37*, 265–290.

Gobet, F., & Simon, H. A. (1996). Templates in chess memory: A mechanism for recalling several boards. *Cognitive Psychology, 31*, 1–40.

Gobet, F., & Simon, H. A. (2000). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science, 24*, 651–82.

Jones, G. A., Gobet, F., & Pine, J. M. (2007). Linking working memory and long-term memory: A computational model of the learning of new words. *Developmental Science, 10*, 853–873.

Lane, P. C. R., Gobet, F., & Ll. Smith, R. (2009). Attention mechanisms in the CHREST cognitive architecture. In L. Paletta & J. K. Tsotsos (Eds.), *Proceedings of the fifth international workshop on attention in cognitive science* (Vol. LNAI 5395, pp. 183–196). Berlin: Springer-Verlag, GmbH.

McLeod, P., Plunkett, K., & Rolls, E. T. (1998). *Introduction to connectionist modelling of cognitive processes.* Oxford, UK: Oxford University Press.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, 81–97.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Pfeifer, R., & Scheier, C. (1999). *Understanding intelligence.* MIT Press.

Sternberg, S. (1966). High speed scanning in human memory. *Science, 153*, 652–654.

A Summary of *Human-System Integration in the System Development Process*

Frank E. Ritter (frank.ritter@psu.edu)
College of Information Sciences and Technology
Penn State
University Park, PA 16802

Abstract

In a recent book, Pew and Mavor and the Committee on Human-System Design Support for Changing Technology (2007) proposed a revision to Boehm's Spiral Model for system development. This revision encourages considering the user within a system as a source of risk. Where these risks are significant, this approach suggests ways to reduce the risk through appropriate studies of the user. This tutorial provides a summary of this model and some of the insights and extensions of this model based on teaching it. These insights are related to learning: learning by the field through using this approach to organize methods and techniques, learning by system development managers that there are sometimes risks related to humans using their systems (and implications for how to teach this), learning about designers as stakeholders, and learning by designers as lessons from one design are applied to later designs. These insights and extensions suggest the importance of shared representations such as cognitive models for educating team members and for the system development process.

Keywords: Human-system design; user models; representation

Introduction

In a recent book, Pew and Mavor and the Committee for Committee on Human-System Design Support for Changing Technology (2007) propose a revision to Boehm's Spiral Model for system development. I present here a summary of this model for system design. This report argues that not understanding aspects of the user can be a risk in system design. Thus, where there are no user related risks, system designers do not need to worry about users. In other cases, where there are risks, the book presents approaches for reducing these risks. User models are a way to share knowledge about users across the design process.

Intended audience. This tutorial will be of interest to people interested in using models in industry as a shared representation, modelers interested in applications of models, and those interested in understanding the Committee's report as edited by Pew and Mavor.

Prerequisite knowledge: This tutorial does not presume any prerequisite knowledge. Attendees may wish to have skimmed the book (which is available on the web page-at-a-time for free), or have examined other work on system design.

The Spiral Model

The spiral model is an approach to system design that encourages increment development of systems in a spiral of requirements specification, technical exploration, and stakeholder commitment. The spiral model is shown in Figure 1, where movement around the spiral represents time and commitment and work on the project.

At each stage in development, the system development is accessed for risks to the system's success. The process is then targeted at reducing these risks. Some risks may be technical, can we build it or can we build it for that price? In these cases, technical work is performed to reduce the risk through technical understanding. Other risks can arise from historical events, which are hard to reduce, and from financial matters, which often can be reduced by setting up contracts at a known price. Risks can also occur due to not understanding user, their tasks, or their interaction with the system, which the report and this tutorial address.

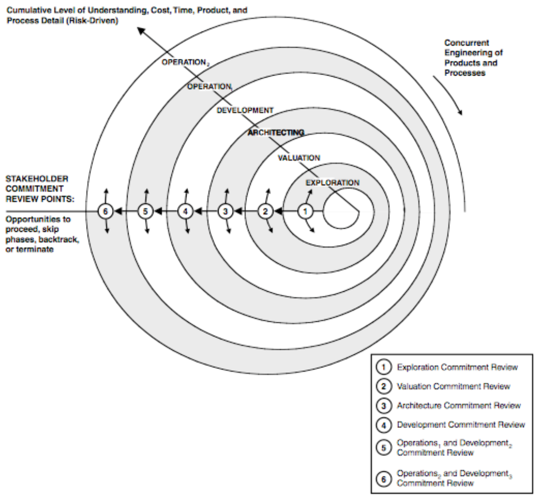

Cumulative Level of Understanding, Cost, Time, Product, and Process Detail (Risk-Driven)
Concurrent Engineering of Products and Processes
OPERATION$_2$
OPERATION$_1$
DEVELOPMENT
ARCHITECTING
VALUATION
EXPLORATION
STAKEHOLDER COMMITMENT REVIEW POINTS:
Opportunities to proceed, skip phases, backtrack, or terminate
6
5
4
3
2
1
1 Exploration Commitment Review
2 Valuation Commitment Review
3 Architecture Commitment Review
4 Development Commitment Review
5 Operations$_1$ and Development$_2$ Commitment Review
6 Operations$_2$ and Development$_3$ Commitment Review

Figure 1. The basic spiral model (Pew & Mavor, 2007).

This revised system design model in Pew and Mavor (2007) has several key features, as noted in the book: (a) Systems should be developed through a process that considers and statisfices the needs of stakeholders. This step is done in the Exploration and Evaluation stages.

(b) Development is incremental and performed iteratively. These related aspects are shown in Figure 1 by the multiple loops representing the increasing resources committed to design and implementation, and through the five stages (Exploration, Valuation, Architecting, Development, and Operation). These stages are incremental because movement from one stage to another depends upon a successful review

(c) Development occurs concurrently, that is, multiple steps may be performed simultaneously. Designers may implement one part of the system while testing another.

(d) The process is mindful of risks during system development and deployment. The level of risk is accessed repeatedly at milestones between stages. Risk is used to manage the project—the level of effort and level of detail of work are driven by the level of risk. Where there is no risk to system development, there is no need for effort to reduce risk. For example, if the system being developed is similar to a known product, there may be no reason to explore further how to support users or how to manufacture it.

Insights

The committee did not set out to create human-system integration (HSI) teaching materials, but the resulting book can be used to teach about HSI, human-computer interaction (HCI), and human factors. In teaching this material, the students and I found several extensions and insights.

(a) The revised spiral model provides a framework for organizing much of HCI and HSI. Most HCI methods can be cast as ways to reduce various types of risks, and most design processes cast as steps in the spiral.

(b) The revised spiral model is not just normative, it is also descriptive. That is, managers may already be working to reduce risk; it is just that they do not see the risks related to users because they do not understand users. This insight suggests that it is likely to be more important to create materials to teach about incipient risks than it is to teach about the revised spiral model process itself.

(c) Designers are stakeholders too. Tools and approaches to reduce risks must support their understanding. They are users of the process and their needs and capabilities are part of the development process.

(d) One of the major results of using shared representations and analyses of systems while being designed may be learning of the design team and application to later designs. Thus, work on creating shared representations should not just include integration across the team and across the design process for a single project (which the book calls for), but also across designs over multiple projects.

Summary

The risk-driven incremental concurrent development model, the later version of the spiral model, provides a useful and safer way to create systems. As a study aid, the model provides a new way to view HSI and HCI methods, design approaches, and development theories, and how to include them in system design.

So, in this new view, the decision to do user research, review, or studies is based on system design risks. If the system development is predicted to be smooth and not novel, then little or no usability studies are required, and little or no should be done. Where there is more risk, more work should be done given the resources. But, the user-related risk has be balanced against other risks. The technology may in fact be riskier, and thus require more resources. Or, as is often the case, the managers understand the technical risk.

There are several corollaries to this. The managers often must be educated about user risks, and we will need books and tutorials like this to help educate system designers about where and when their theories of users mismatch the world.

We will need improved representations of users (shared representations) to use in the design process, similar to how blueprints are used in buildings.

Acknowledgements

A graduate class on HCI at Penn State and a group at QinetiQ Malvern provided useful discussions that lead to the preparation of this tutorial and provided useful comments. Barry Boehm provided some slides and figures that I used to create my teaching materials. Olivier Georgeon, Jon Morgan, and Rick Koubek provided comments that improved the presentation.

References

Pew, R. W., & Mavor, A. S. (Eds.). (2007). *Human-system integration in the system development process: A new look*. Washington, DC: National Academy Press. books.nap.edu/catalog.php?record_id=11893.

Frank Ritter currently teaches at the College of IST at Penn State, previously he has worked at BBN Labs and taught at the U. of Nottingham, where he was the Director of the Institute for Applied Cognitive Science. He earned his PhD in AI and Psychology and a MS in psychology from Carnegie-Mellon. He has a BSEE from the University of Illinois at Urbana/Champaign. He was a member of the committee that helped prepare the book the tutorial is based on.£

Rational Explanations

Symposium

Organiser:
Andrew Howes[1]

Panelists:
John Anderson[2]*, Nick Chater*[3]*, Lael Schooler*[4]*, Richard L. Lewis*[5]*, Henry Brighton*[4]

University of Manchester[1]

Carnegie Mellon University[2]

University College London[3]

Max Planck Institute for Human Development[4]

University of Michigan[5]

Human behaviour is the product of two adaptive systems that generate and select actions beneficial to the organism. Through one of these systems, genetic selection, the species has acquired relatively stable psychological mechanisms. Through the other, learning, individuals acquire the knowledge that determines behaviour on a moment to moment basis. Together these systems generate the complex behaviours that cognitive science seeks to explain.

Focusing on behaviour as the product of adaptation opens up possibilities for deep explanations that answer questions not only about how people behave but also why they behave as they do. These *rational explanations* are grounded in theories of the constraints on adaptation, including constraints derived from the observable structure of the task environment (either evolutionary or local). They are also grounded in one, or other, assumption of rationality, which is sometimes defined in terms of optimality criteria. The assumption of rationality is the point of departure for a range of approaches to understanding cognition and perception, including rational analysis and related Bayesian approaches (Anderson, 1990; Anderson & Schooler, 1991; Oaksford & Chater, 2007), optimal motor control approaches (e.g. Maloney, Trommershäuser, & Landy, 2007), as well as signal detection theory and ideal observer analysis (Giesler, 2003). Others, notably Simon (1955) and Gigerenzer, ABC Research Group and Todd (2000), focus on the adaptive benefit of heuristics given that rationality is limited by psychological bounds.

The symposium will encourage discussion of relevant contributions made over the past 20 or so years and, further, will seek to expose the key unanswered questions. The remainder of this abstract provides brief descriptions of current contributors of the symposium speakers.

Anderson began to pursue the issue of how cognition might be adapted to the statistical structure of the environment in the late 1980s and soon published “The Adaptive Character of Thought" (Anderson, 1990). The fundamental idea was that to understand human cognition we do not need to develop a theory of its mechanisms but only need to understand the statistical structure of the problems it faces. This effort has had successes in developing theories of human memory and categorization. In the memory domain, Anderson and Schooler (1991) collected statistics on the information-retrieval demands made on human memory and showed that behavioral functions mirrored these. In the case of categorization this lead to a program which accounted for a wide range of human data and which did well on a number of machine-learning data sets. The rational analysis work played a major role in defining a better version of the ACT-R subsymbolic activation processes. Anderson realized that while these subsymbolic processes were tuned to the statistical structure of the environment, one needed an overall computational structure like ACT to understand how they interacted.

Furthering his earlier work with Anderson, Schooler is now pursuing a modeling and empirical effort that, in the context of David Marr's functional approach to understanding cognition, bridges two research programs grounded in an appreciation of the adaptive value of human cognition: The program on fast and frugal heuristics explores cognitive processes that use limited information to make effective decisions; and the ACT-R research program that strives for a unified theory of cognition. This work illustrates how a memory system that is tuned to automatically retrieve information can be exploited for a different purpose, namely making inferences about real objects in the world, based on meta-cognitive judgments about how the memory system responds to stimuli (Schooler & Hertwig, 2005). This work provides a good point of departure to discuss the kinds of cognition that yield to a rational analysis and those that might not.

Chater has argued that rationality is defined by the ability to reason about uncertainty. Although people are typically poor at numerical reasoning about probability, human thought, shaped through evolution, is sensitive to subtle patterns of qualitative Bayesian, probabilistic reasoning. In Bayesian Rationality (Oaksford & Chater

2007), the case is made that cognition in general, and human everyday reasoning in particular, is best viewed as solving probabilistic, rather than logical, inference problems. The psychology of "deductive" reasoning is addressed directly: It is argued that purportedly "logical" reasoning problems, revealing apparently irrational behaviour, are better understood from a probabilistic point of view. Data from conditional reasoning tasks, for example, are explained by recasting these problems probabilistically. The probabilistic approach makes a variety of novel predictions which have been experimentally confirmed.

Brighton's research, e.g. Brighton and Todd (2008), focuses on modeling the computational processes that underlie adaptive behaviour. With Gigerenzer, Brighton views heuristics as cognitive processes that gain efficiency by ignoring information. In contrast to the widely held view that less processing reduces accuracy, the study of heuristics shows that less information, computation, and time can in fact improve accuracy. Heuristics are ecologically rational when deployed in the right environment. The "adaptive toolbox" provides a systematic theory of heuristics that identifies their building blocks and the evolved capacities they exploit. According to this program, while people have biased minds and ignore part of the available information, they can handle uncertainty more efficiently and robustly than an unbiased mind relying on more resource-intensive and general-purpose processing strategies.

Lewis and Howes assume that individuals adapt rationally to a utility function given constraints imposed by their cognitive architecture and the *local* task environment (Howes, Lewis, Vera, accepted). This assumption underlies a new approach to modelling and understanding cognition —cognitively bounded rational analysis—that sharpens the predictive acuity of general, integrated, theories of cognition and action. Such theories provide the necessary computational means to explain the flexible nature of human behaviour, but in so doing introduce extreme degrees of freedom in accounting for data. The new approach narrows the space of predicted behaviours through analysis of the payoff achieved by alternative strategies, rather than through fitting strategies and theoretical parameters to data. Analyses of dual-task performance, and the development and analysis of a new theory of ordered responses, yield several novel results, including a new understanding of the role of strategic variation in existing accounts of dual-task performance, and the first predictive, quantitative, account showing how the details of ordered dual-task phenomena emerge from the rational control of a cognitive system.

References

Anderson, J.R. (1990). *Rational Analysis*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2, 396-408.

Brighton, H., & Todd, P. M. (2008). Situating rationality: Ecologically rational decision making with simple heuristics. In: P. Robbins & M. Aydede (Eds.) *Cambridge handbook of situated cognition* (pp. 322-346). Cambridge: Cambridge University Press.

Geisler, W. S. (2003). Ideal observer analysis. In L. Chalupa & J. Werner (Eds.), The visual neurosciences (pp. 825-837). Boston: MIT Press.

Gigerenzer, G., ABC Research Group, & Todd, P.M. (1999). *Simple Heuristics That Make Us Smart*. New York: Oxford University Press.

Howes, A., Lewis, R.L., Vera, A. (accepted). Rational behaviour under task and processing constraints: Implications for testing theories of cognition and action. *Psychological Review*, accepted.

Maloney, L.T., Trommershäuser, J., & Landy, M.S. (2007). Questions without words: A comparison between decision making under risk and movement planning under risk. In Gray, W.D. (Ed.) *Integrated Models of Cognitive Systems*. New York: Oxford University Press.

Oaksford, M., & Chater, N. (2007). *Bayesian rationality: The probabilistic approach to human reasoning*. Oxford: Oxford University Press.

Schooler, L. J., & Hertwig, R. (2005). How forgetting aids heuristic inference. *Psychological Review, 112*, 610–628

Simon, H.A., (1955). A behavioural model of rational choice. *Quarterly Journal of Economics, 69*,
99-118.

Increasing Generalization Requirements for Cognitive Models: Comparing Models of Open-ended Behavior in Dynamic Decision-Making

Christian Lebiere (cl@cmu.edu)
Department of Psychology, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213 USA

Cleotilde Gonzalez (coty@cmu.edu) and Varun Dutt (varundutt@cmu.edu)
Dynamic Decision Making Laboratory, Department of Social and Decision Sciences, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213 USA

Walter Warwick (wwarwick@alionscience.com)
Alion Science and Technology
4949 Pearl East Circle, Boulder, CO 80301

Keywords: Cognitive Architectures; Model Validation and Comparison; Dynamic Decision-Making.

Introduction

Model comparison is becoming an increasingly common method in computational cognitive modeling. The methodology is seemingly straightforward: model comparisons invite the independent development of distinct computational approaches to simulate human performance on a well-defined task. Typically, the benchmarks of the comparison are goodness-of-fit measures to human data that are calculated for the various models. Although the quantitative measures might suggest that model comparisons produce "winners," the real focus of model comparison is, or at least should be, on understanding in some detail how the different modeling "architectures" have been applied to the common task. And in this respect, the seemingly straightforward method of model comparison becomes more complicated.

The idea that a model comparison might be used to pick a winning approach resonates with common intuitions about model validation, namely, that a good fit is good evidence for the theory the model implements. But to the extent that model comparisons seek to illuminate general features of computational approaches to cognition rather than to validate a single theory of cognition, they depart from the familiar mode of good fit, good theory. Instead, a model comparison forces us to think about the science of modeling. A good fit is thus relegated to a necessary requirement rather than an end in itself, and the focus shifts toward a deeper understanding of the modeling approaches themselves. This shift brings into focus a host of new questions having to do with the relationship between model and architecture, theory and implementation, the relative contributions of the modeler and of the architecture to the final model, the role of parameter estimation in model development, the suitability of the simulated task to exercise features of the various architectures, the extensibility of the simulated task and the practical considerations that go into integrating disparate approaches within a common simulation environment. In this symposium, we address these issues in the context of our own model comparison. Our ultimate goal is to evolve a formal methodology to ensure the soundness of future comparison efforts and develop an infrastructure to make such efforts an ongoing process rather than one-off events.

Requirements

We have direct experience from a number of modeling comparisons projects, including the AFOSR AMBR modeling comparison (Gluck & Pew, 2005) and the NASA Human Error Modeling comparison (Foyle & Hooey, 2008). We have also entered cognitive models into multi-agent competitions (Billings, 2000; Erev et al, submitted) and organized symposia featuring competition between cognitive models as well as mixed human-model competitions (Lebiere & Bothell, 2004; Warwick, Allender, Strater and Yen, 2008). From these endeavors, we have gained an understanding of the required (and undesirable) characteristics of a task for such projects. While previous model comparison efforts did illustrate the capabilities of some modeling frameworks, the tasks were often ill suited to that purpose for a number of reasons:

- Some tasks demand a considerable effort just to model the details of task domain itself, which often results in a model whose match to the data primarily reflects the structure and idiosyncrasies of the task rather than the underlying cognitive mechanisms. This does not serve the primary purpose of a model comparison effort, which is to shed light upon the merits of the respective modeling frameworks rather than the cleverness and diligence of their users.
- Some tasks do not stretch model functionality beyond the conditions for which human data is available. The comparison effort can then be gamed by simply optimizing the model parameters to the data available, which puts frameworks that emphasize constrained, principled functionality at a disadvantage over those that permit arbitrary customization and parameterization.

- Likewise, some tasks are too specialized, emphasizing a single aspect, characteristic or mechanism of cognition and do not require the broad, integrated functional capabilities required of a general cognitive framework.
- If no common simulation or evaluation framework is provided, each team can focus on the aspects of the task most amenable to their framework, at the cost of making a direct comparison all but impossible.
- Finally, tasks for which no suitably comparable human data is available bias the effort toward a purely functional evaluation of model against model (rather than against data), which emphasizes performance at the expense of empirical fidelity.

This experience has taught us that the desirable characteristics of a task for a model comparison include:

- Lightweight, to limit overhead of integration, task analysis and knowledge engineering requirements.
- Fast, to allow efficient model development and collection of large numbers of Monte Carlo runs.
- Open-ended, to discourage over-parameterization and over-engineering of the model and test its generalization over a broad range of situations.
- Dynamic, to explore emergent behavior that is not predictable from the task specification.
- Simple, to engage basic cognitive mechanisms in a direct and fundamental way.
- Tractable, to encourage a direct connection between model and behavioral data.

Like other competitive benchmarks of human cognition (e.g. Robocup), the key is finding the right combination of simplicity, flexibility and emergent complexity.

Comparison Challenge

We believe the task we have selected, the Dynamic Stocks and Flows (Dutt & Gonzalez, 2007), meets these requirements and strikes the right combination between simplicity and complexity (Lebiere, Gonzalez, & Warwick, in press). The instructions to participate in this comparison challenge are on a web site[1], together with an executable version of the task, a text-based socket connection for models, and experimental data for a number of experimental conditions for model calibration. We collected data on additional conditions that were used to test the submitted model's generalization beyond the available conditions. Our focus in evaluating models was two-fold: quantitative measures of the models' fit to the data in the generalization conditions, and qualitative assessment of the generality and constraints of the underlying theories in meeting the demands of the task. The best entries under each criterion were invited to describe their model in this symposium.

Conclusion

A number of tests for a general theory of intelligence have been advanced (e.g. Cohen, 2005; Anderson & Lebiere, 2003). A key common aspect is to enforce generality in approach, in order to prevent special-purpose optimization to narrow tasks and force integration of capabilities. One can view that strategy as effectively overwhelming the degrees of freedom in the architecture with converging constraints in the data. However, precise computational specifications of those tests have to tread a tight rope between requiring unreasonable amounts of effort in modeling broad and complex tasks and falling back into narrow task specifications that will again favor engineered, optimized approaches. This model comparison challenge is our attempt at testing general cognitive capabilities in an open-ended manner by offering low barriers to entry in confronting different approaches with specific common problems that encourage integrated cognitive approaches.

Acknowledgments

This research was partially supported by the Advanced Decision Architectures of the Army Research Laboratory award number DAAD19-01-2-0009. Many thanks to Hau-yu Wong for her help in analyzing the data and to Mala Gosakan and Michael Matessa for their help in running the models.

References

Anderson, J. R. & Lebiere, C. (2003). The Newell test for a theory of cognition. *Behavioral & Brain Sciences 26*, 587-637.

Billings, D. (2000). The First International RoShamBo Programming Competition. *ICGA Journal*, Vol. 23, No. 1, pp. 42-50.

Cohen, P. (2005). If Not Turing's Test, Then What? *AI Magazine 26(4)*: 61–67.

Erev, I., Ert, E., Roth, A. E., Haruvy, E., Herzog, S., Hau, R., Hertwig, R., Stewart, T., West, R., & Lebiere, C. (submitted). A choice prediction competition, for choices from experience and from description. *Journal of Behavioral Decision Making*.

Foyle, D. & Hooey, B. (2008). *Human Performance Modeling in Aviation*. Mahwah, NJ: Erlbaum.

Gluck, K, & Pew, R. (2005). *Modeling Human Behavior with Integrated Cognitive Architectures.* Mahwah, NJ: Erlbaum.

Dutt, V., & Gonzalez, C. (2007). Slope of inflow impacts dynamic decision making. Paper presented at the Conference of the System Dynamics Society.

Lebiere, C., & Bothell, D. (2004). Competitive Modeling Symposium: PokerBot World Series. In *Proceedings of the Sixth International Conference on Cognitive Modeling*, Pp. 32-32.

Lebiere, C., Gonzalez, C., & Warwick, W. (in press). Convergence and constraints revealed in a qualitative model comparison. *Journal of Cognitive Engineering and Decision Making.*

Warwick, W., Allender, L., Strater, L., & Yen, J. (2008). AMBR Redux: Another Take on Model Comparison. Symposium given at the *Seventeenth Conference on Behavior Representation and Simulation*. Providence, RI.

[1] http://www.cmu.edu/ddmlab/modeldsf

A Model of Probability Matching in a Two-Choice Task Based on Stochastic Control of Learning in Neural Cell-Assemblies

Roman V. Belavkin (R.Belavkin@mdx.ac.uk)

Christian R. Huyck (C.Huyck@mdx.ac.uk)
School of Engineering and Information Sciences
Middlesex University, London NW4 4BT, UK

Abstract

Donald Hebb proposed a hypothesis that specialised groups of neurons, called cell-assemblies (CAs), form the basis for neural encoding of symbols in the human mind. It is not clear, however, how CAs can be re-used and combined to form new representations as in classical symbolic systems. We demonstrate that Hebbian learning of synaptic weights alone is not adequate for all tasks, and that additional meta-control processes should be involved. We describe an earlier proposed architecture (Belavkin & Huyck, 2008) implementing such a process, and then evaluate it by modelling the probability matching phenomenon in a classic two-choice task. The model and its results are discussed in view of mathematical theory of learning, and existing cognitive architectures as well as some hypotheses about neural functioning in the brain.

Keywords: Artificial Intelligence, Cognitive Science, Neuroscience, Decision making, Intelligent agents, Learning, Bayesian modeling, Computational neuroscience, Human experimentation

Introduction

There exists a variety of artificial systems and algorithms for learning and adaptation. Most of them can be classified as sub-symbolic (e.g. Bayesian and connectionist networks) or symbolic systems (e.g. rule-based systems). Known natural learning systems use neural networks, and therefore can be classified as using sub-symbolic computations. A distinguishing feature of the human mind, however, is the ability to use rich symbolic representations and language.

From an information-theoretic point of view, symbols are elements of some finite set that are used to encode discrete categories of sub-symbolic information. They enable communication of information about the environment or a complex problem in a compact form. One obvious benefit is that with language, one can learn not only from one's own experience, but also from experiences of others. The benefits of reading a guidebook before going abroad are obvious.

The duality between sub-symbolic and symbolic approaches has been studied in cognitive science. There exists sub-symbolic (i.e. connectionist), symbolic (e.g. SOAR, Newell, 1990) and hybrid architectures (e.g. ACT–R, Anderson & Lebiere, 1998) for cognitive modelling. These different approaches, however, have not yet explained where the symbols are in the human mind, or how the brain implements symbolic information processing.

It was proposed by Hebb (1949) that symbols are represented in the brain not by individual neurons, but by correlated activities of groups of cells, called *cell assemblies* (CAs). The CABOT project set out to test and demonstrate this idea in an engineering task by building an artificial agent, situated in a virtual environment, capable of complex symbolic processing, and implemented entirely using CAs of simulated neurons. Some of the objectives have already been achieved and reported elsewhere (e.g. Huyck & Belavkin, 2006; Huyck, 2007; Belavkin & Huyck, 2008). The architecture and some of these works will be discussed in the next section.

The work described in this paper is concerned with a particular aspect of the project — a stochastic meta-control mechanism that modulates Hebbian learning to allow for re-use and combination of CAs into new representations, such as learning logical implications (i.e. procedural knowledge). As will be discussed in this paper, this cannot be achieved by using a Hebbian learning mechanism alone. A unique contribution of this work is evaluation of the meta-control mechanism in a cognitive model of the probability matching phenomenon in a two-choice experiment (Friedman et al., 1964). The results suggest that a proposed mechanism is a plausible model. Some neurophysiological studies and hypotheses about the brain circuitry will be discussed supporting the biological plausibility of the architecture.

Cell-Assemblies as the Basis of Symbols

In this section, we outline some of the basic features of the CABOT architecture as well as the CA hypothesis.

Neural Information Processing in CABOT

It is widely accepted that human cognition is the result of the activity of approximately 10^{11} neurons in the central nervous system (CNS) that interact with each other as well as with the outside world via the peripheral nervous system (PNS). Biological neurons are complex systems, and they have been modelled with various levels of details. In our system, we use fatiguing, leaky, integrate and fire (fLIF) neurons.

The 'integrate and fire' component is based on the classical idea that the neuron 'fires' (or spikes) if its action potential, A, exceeds a certain threshold value θ: $y = 1$ if $A \geq \theta$; $y = 0$ otherwise. The action potential, A, is a function of the inner product (integrator): $\langle x, w \rangle = \sum_{i=1}^{k} x_i w_i$, where $x \in \mathbb{R}^k$ is the stimulus vector (pre-synaptic), and $w \in \mathbb{R}^k$ is the synaptic weight vector of the neuron. Here, $\mathbb{R}^k$ is a k-dimensional real vector space, where k is the number of synapses to the neuron. We use binary signals, and therefore x is a k-dimensional binary vector.

The ‘leaky’ property refers to a more complex (non-linear) dependency of the action potential on the pre- and post-synaptic activity:

$$A_{t+1} = \frac{A_t}{d_t} + \langle x_t, w_t \rangle \,, \quad d_t = \begin{cases} \infty & \text{if fired } (y_t = 1) \\ d \geq 1 & \text{otherwise} \end{cases}$$

Thus, the action potential is accumulated over several time moments if the neuron does not fire. Parameter $d \geq 1$ allows for some of this activation to ‘leak’ away. This is the LIF model (Maas & Bishop, 2001).

The ‘fatigue’ property refers to a dynamic threshold that is defined as follows:

$$\theta_{t+1} = \theta_t + F_t \,, \quad F_t = \begin{cases} F_+ \geq 0 & \text{if fired } (y_t = 1) \\ F_- < 0 & \text{otherwise} \end{cases}$$

where values F_+ and F_- represent the *fatigue* and fatigue *recovery* rates. Thus, if a neuron fires at time t, its threshold increases, and it is less likely to fire at time $t+1$.

The fatiguing and leaky properties of the neural model allow for a non-trivial dynamics of the system. Repetitive stimulation of excitatory synapses increases the probability of a neuron to fire, even if the weights have small (positive) values. On the other hand, if the neuron fires repetitively, its threshold increases reducing the chance of it firing again. Thus, frequencies of pre- and post-synaptic activities are important factors in our system.

The weights, w, of a neuron can adapt according to the compensatory learning rule (Huyck, 2007), which is an implementation of the Hebbian principle (Hebb, 1949), where w_{t+1} depends on the correlation between the pre-synaptic, x_t, and the post-synaptic, y_t, activities.

The above described properties are known characteristics of biological neurons, and our model is a compromise between computational efficiency and biological plausibility that is important for the emerging dynamics that we discuss.

Neural Cell-Assemblies

Networks of neurons can be used as general function approximators and applied in a variety of tasks including control, pattern recognition and classification. Our system, CABOT, uses recurrent, partially connected networks (a mesh) of fLIF neurons with a largely pre-defined topology. The non-linearity of the cells and the topology of the network leads to a complex dynamics of the system similar to that in attractor and recurrent nets (e.g. Hopfield, 1982), where some of the states are more probable. These more ‘stable’ states can be characterised by groups of neurons that remain significantly more active than the other cells in the system. According to Hebb (1949), we refer to such reverberating groups of cells as *cell assemblies* (CAs).

In our system, the formation of CAs depends on the topology of the network, and it is facilitated by the adaptation of the weights between connected cells. Therefore, CAs can be used for pattern classification of sensory stimuli (i.e. patterns from external connections). This leads to functional *specialisation* of neurons in the network based on CAs — two cells are functionally different if they belong to different CAs, even though they are similar architecturally. Such specialisation is observed in many neural networks, such as in self-organising maps (Kohonen, 1982) and particularly in the human brain. Note that CAs are not necessarily disjoint sets of cells. A single cell may be a member of several overlapping CAs. This feature can be used to encode hierarchies of patterns (Huyck, 2007).

An important property of CAs’ dynamics is their persistence. When enough neurons fire to start the reverberating circuit, the CA ignites. Once ignited, the activity within the cells in a CA may be sufficient to support itself. Many variables can contribute to this effect. In particular, the fatigue and recovery rate parameters in our system effect persistence.

A CA’s activity does not only depend on the external patterns, but also on the activity of other CAs in the system as they can ignite and extinguish each other. Thus, the activity of several CAs can be characterised by different patterns of ignition order and so on. It was demonstrated earlier that such state transitions in the system of CAs are sufficiently controllable to implement a broad range of tasks simulating symbolic processing that will be discussed below.

Symbols and Human Cognition

Many models of biological neurons suggest that synaptic weights may represent the memory for statistical and sub-symbolic information of the stimulus. In particular, in many algorithms for training artificial neural networks (e.g. Oja, 1982), the weight vector $w \in \mathbb{R}^k$ corresponds to one of the principal eigenvectors of the covariance matrix $E\{xx^\dagger\}$ of input vectors $x \in \mathbb{R}^k$ that have been observed. On the other hand, human cognition, and human knowledge in particular, is encoded using symbolic representations, and the link between the symbols and neural models is less clear.

It was proposed by Hebb (1949) that CAs may be considered as the neural basis of symbols. Indeed, as discussed in the previous section, CAs can be easily mapped to some discrete categories of the stimuli, and their activity patterns can model serial processing typical for symbolic algorithms. Testing this hypothesis experimentally is one of the main objectives of the CABOT project. However, many challenges had to be overcome to make a purely CA-based system performing some non-trivial symbol processing task.

Previously, we reported a system performing a counting task that consisted of 7 modules and 40 CAs (Huyck & Belavkin, 2006). A more recent system, CABOT 2, is an artificial agent functioning in a virtual 3D environment that has a model of visual information processing, and is capable of natural language processing and action selection (Belavkin & Huyck, 2008). One of the advantages of such a CA-based architecture is that neural CAs, that we associate with symbolic representations, integrate also all the sensory (i.e. sub-symbolic) information, which can be a natural solution to the *symbol grounding* problem. An associated phenomenon of symbolic processing is *grounding transfer* — combination and re-use of existing symbols to form new representations.

The re-use of symbols is also important for learning procedural knowledge. Indeed, a logical implication (i.e. a production rule) may use combinations of symbols both in the antecedent and the consequent, and generally there are many more possible combinations than the number of rules that are actually used. Hybrid architectures, such as ACT–R, rely on statistical (sub-symbolic) computations to 'filter' out the unwanted rules in the process called *conflict resolution*. In CABOT, associations between CAs are learnt due to the Hebbian learning mechanism. However, as will be pointed out below, this mechanism alone is not sufficient to implement learning of particular associations between CAs representing existing symbols. To resolve this problem, an additional stochastic meta-control mechanism, moderating the Hebbian learning, has been introduced (Belavkin & Huyck, 2008). Here, we use this mechanism to model the probability matching phenomenon in a classical two-choice experiment, and this way evaluate its plausibility.

Stochastic Meta-Control of Learning

Two-Choice Task

Let x, y_1 and y_2 be three symbols, where x represents a stimulus (antecedent), and y_1, y_2 represent two alternative responses (consequents). Thus, we have a conflict between two implications $x \to y_1$ and $x \to y_2$ shown on the diagram below

$$\begin{tikzcd}
 & x \arrow[dl] \arrow[dr] & \\
y_1 & & y_2
\end{tikzcd}$$

This is a simplest two-choice task (a more complex two-choice task may involve a set of different stimuli). The choice of y_1 or y_2 is followed by some reinforcement event E that may have different utility values (e.g. a success after choosing y_1 or a failure after choosing y_2). Learning the associations between the choices and the utility values, such as $u(x \to y_2) \le u(x \to y_1)$, leads to a preference $y_2 \lesssim y_1$, and therefore learning rule $x \to y_1$. If the reinforcement event is not deterministic, but occurs with some probability $P(E) = \pi \in [0,1]$, then the preference of y_1 to y_2 may also be stochastic. As demonstrated in many experiments with animals and human participants, the frequency of choosing y_1 adapts to probability π of reinforcement with high utility — a phenomenon referred to as the *probability matching*. This phenomenon can be explained based on the theories of optimal statistical decisions (Wald, 1950) and information value (Stratonovich, 1965).

Principles of Statistical Learning

Let us consider an abstract system with input $x \in X$ and output $y \in Y$. Any learning system can be characterised by some optimisation criteria and information constraints (Belavkin, 2009). Optimisation corresponds to some preference relation on the input-output pairs $(x,y) \in X \times Y$. In a deterministic setting, this preference relation can be represented by a utility function $u : X \times Y \to \mathbb{R}$, while in stochastic setting one considers conditional probability distributions $P(u \mid x,y)$ on values of utility $u \in \mathbb{R}$. If the utility function $u = u(x,y)$ or the joint distribution $P(u,x,y)$ is known (and hence $P(u \mid x,y)$), then given input x, the optimal output $\hat{y} \in Y$ maximises the expected utility:

$$\hat{y}(x) = \arg\max_{y} E_P\{u \mid x,y\}$$

where $E_P\{\cdot\}$ denotes the expected value with respect to distribution P (in the deterministic case, $E_P\{u \mid x,y\}$ coincides with $u = u(x,y)$). The *greedy* strategy of always choosing the optimal output can be expressed as follows:

$$P(y \mid x) = \begin{cases} 1 & \text{if } y = \hat{y}(x) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Information constraints mean that either the utility function $u = u(x,y)$ or the distribution $P(u,x,y)$ is not known. Instead, one has some data from past occurrences of $(u,x,y) \in \mathbb{R} \times X \times Y$ which can be used to estimate $\tilde{u}(x,y) \approx E_P\{u \mid x,y\}$. In this case, the greedy strategy for choosing the system's output is not optimal. The optimal policy is the following exponential ('soft-max') distribution (e.g. Belavkin, 2009):

$$\hat{P}(y \mid x) = Q(y \mid x)\,\exp\{\beta\tilde{u}(x,y) - \Psi(\beta,x)\} \tag{2}$$

where $Q(y \mid x)$ is the distribution corresponding to the minimum of information (e.g. no data), parameter β is related to the amount of information available in the data, and $\Psi(\beta,x)$ is defined from the normalisation condition (i.e. $\Psi(\beta,x) = \ln\sum_Y Q(y \mid x)\,\exp\{\beta\tilde{u}(x,y)\}$). Distribution (2) is obtained by solving the following variational problem

$$U(I) = \sup_P\{E_P\{u\} : I(P,Q) \le I\}$$

where $I(P,Q)$ is the Kullback-Leibler divergence of distribution $P(u,x,y)$ from $Q(u,x,y)$ representing information amount I contained in the data. Parameter β^{-1} appears in the solution as the Lagrange multiplier related to information constraint I by the derivative of $U(I)$:

$$\beta^{-1} = U'(I) \tag{3}$$

The function above is decreasing so that $\beta^{-1} \to 0$ (or $\beta \to \infty$) as information increases. Note that the exponential distribution (2) converges to the greedy strategy (1) as $\beta \to \infty$.

Exponential distributions are often used for selecting the output of a system in machine learning and stochastic optimisation algorithms. It is also used in the ACT–R cognitive architecture to model some stochastic properties of behaviour. In particular, it was used in the ACT–R model of the two-choice experiment, discussed below. However, the 'temperature' parameter β^{-1} is usually set to some constant value or determined from some arbitrary 'annealing' schedule. The relation of β^{-1} to entropy of success in ACT–R was proposed in (Belavkin, 2002/2003), and it was shown that it improves the match between the models and data. The derivation of optimal function $\beta^{-1} = U'(I)$ can be found in (Stratonovich, 1965) and more generally in (Belavkin, 2009).

Meta-Control of Hebbian Learning

The output of a neuron depends on its weight vector $w \in \mathbb{R}^k$, which, according to Hebb's hypothesis, adapts to the correlation between the pre- and post-synaptic activities x and y in the past. It is attractive to conclude, therefore, that Hebbian learning is a particular implementation of the statistical learning. However, the utility is clearly missing in this description of neural plasticity. What criteria does such a process of changing the weights optimise? If in a two-choice task the system accidentally chooses the 'incorrect' cell-assembly y_2, then the weights associating x with neurons in y_2 increase due to the correlation-based Hebbian learning. This can only increase the chance of $x \to y_2$ igniting in the future, even though the reinforcing event E following the choice of $x \to y_2$ has a low utility (i.e. a failure). Thus, some additional process should be involved to increase the chance of the 'correct' combination $x \to y_1$ after the reinforcing event E. Such a process appears to be especially useful if the CA-based symbolic representations, formed earlier, are to be re-used. Below we describe a neural implementation of such a meta-control of Hebbian learning based on the utility feedback (Belavkin & Huyck, 2008) following principles of statistical learning.

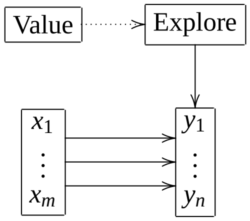


Figure 1: Components and connections of the Value and Explore modules controlling Hebbian learning of connections between CAs in modules X and Y. Solid and dashed arrows show excitatory and inhibitory connections respectively.

The meta-control process involves two specialised modules: Value and Explore. Their connections in the system are shown on Figure 1. Here, $X = \{x_1, \dots, x_m\}$ and $Y = \{y_1, \dots, y_n\}$ are sets of CAs representing m stimuli and n responses respectively. Initially, there are excitatory connections from every CA in X to all CAs in Y, which means that all pairs (x, y) (i.e. all rules $x \to y$) are equally preferred. Thus, given input $x \in X$, any response $y \in Y$ can be selected. However, due to Hebbian learning, the connection $x \to y$ is reinforced if a particular pair of CAs ignite together, giving the pair a higher chance to ignite together in the future. Thus, simply by virtue of Hebbian learning, the system can learn eventually to prefer some random pairs. The purpose of the Value and Explore modules is to make this process selective according to the utility value of the feedback.

The output activity of the Value module represents the utility values u associated with the pair (x, y) selected on the previous step. The input of the module can be configured according to the application (e.g. using sensory information).

The purpose of the Explore module is to randomise the activity of the response CAs (i.e. CAs in set Y). The Explore module contains cells that can be active without any external stimulation due to spontaneous activation. The cells in the Explore module send excitatory signals to all CAs in Y, and the weights of these connections do not change. Thus, the activity in the Explore module can trigger randomly any response CA, and this process does not have a memory. The Explore module implements the effect of parameter β^{-1} in the exponential distribution.

The Value module sends inhibitory connections to the Explore module, so that high activity of the Value cells may shut down the activity in the Explore module. As a result, any response CA that has been ignited in set Y will persist longer because it is less likely to be shut down by another CA. Such a connectivity implements the following learning scheme: If a particular pair (x, y) results in a high utility value, then high activity of the Value module inhibits the Explore module, and the responsible (x, y) pair is allowed to persist longer, and the $x \to y$ connection increases relative to others due to Hebbian learning.

Learning the 'correct' rules (subset $R \subset X \times Y$) contributes to a better performance of the system (i.e. higher expected utility). As a consequence, the average activity of the Value module increases with time, while the activity of the Explore module decreases. This dynamic also corresponds to a decrease of parameter β^{-1} as information increases making the system less random and more deterministic.

Modelling Probability Matching

To test how adequately the above mechanism can represent properties of human cognition, we evaluate its performance against data from a classic two-choice experiment due to Friedman et al. (1964). The choice of this dataset was motivated not only by its quality and detailed description of the procedures, but also because it was used to 'calibrate' stochastic properties of other cognitive architectures, such as ACT–R (Anderson & Lebiere, 1998). The complete description of the experiment and data can be found in the original paper (Friedman et al., 1964). Here we give a basic outline.

Experiment Description and Previous Work

In this experiment, participants were asked to select one of two responses on presentation of a stimulus. After the response was selected, a reinforcement event E occurred with probability $P(E) = \pi$ that did not depend on the response. Each participant had to perform this task in three sessions, each session consisting of 8 blocks, each block consisted of 48 trials. The probability $P(E) = \pi$ changed between each 48–trial block. This paper will report only simulations of results in Sessions 1 and 2. In these two sessions, blocks 1, 3, 5 and 7 had $P(E) = .5$, and blocks 2, 4, 6, and 8 were with $P(E) \in \{.1, .2, .3, .4, .6, .7, .8, .9\}$ that was assigned according to a random pattern. Thus, probability $P(E) = \pi$ was alternating between .5 and some value above or below .5 between

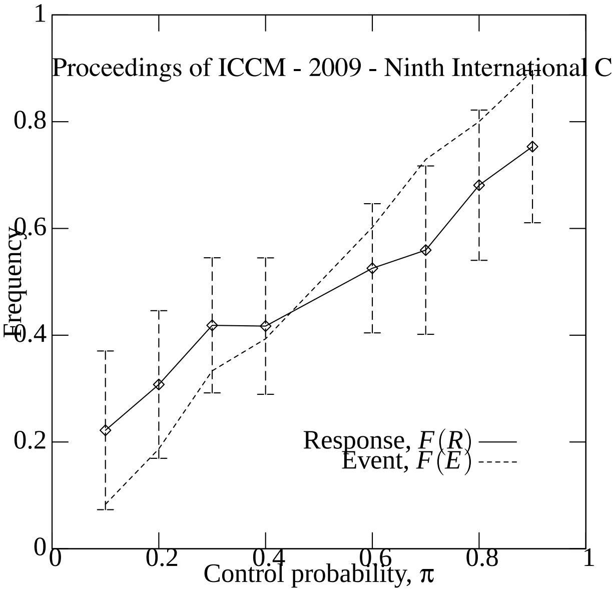


Figure 2: Frequency of response (ordinates) as a function of the probability of reinforcing this response (abscissae). Points and error bars represent average response and standard deviations in 48–trials of two-choice task from 80 participants, reported in (Friedman et al., 1964). Dashed line shows frequency of the reinforcing event itself.

48-trial blocks. The data recorded the number of times Response 1 was chosen in each 48-trial block.

Figure 2 shows the results of these experiments, reported by Friedman et al. (1964). The charts show frequencies of Response 1, $F(R)$, and reinforcement events, $F(E)$, as functions of the control probability $P(E) = \pi$. One can see that the frequency of the reinforcement event $F(E)$ approximates the the control probability $F(E) \approx P(E)$. The response frequency $F(R)$ also matches the probability $P(E)$, but it differs significantly at the lower and higher ends of the range: When $P(E)$ is low ($\pi = .1$), the participants overestimate the probability ($F(R) \geq P(E)$); when $P(E)$ is high ($\pi = .9$), the participants underestimate it ($F(R) \leq P(E)$). Thus, the response appears to be less certain than the reinforcing event.

As suggested by Anderson and Lebiere (1998), this experimental evidence indicates against using the greedy strategy (1) for choosing the response. The data was modelled in ACT–R by sampling responses from exponential distribution with some $\beta^{-1} > 0$. This agrees with equations (2) and (3), where $\beta^{-1} \to 0$ only when information $I \to \sup I$. We now describe a model of this experiment implemented in CABOT.

Model Description

The model used the architecture shown on Figure 1, where module X consisted of CAs representing one or more stimuli, and module Y contained two CAs representing two alternative responses. There were excitatory connections with low weights from module X to all CAs in module Y. The weights on these connections, however, could adapt according to a Hebbian rule increasing associations $x \to y$ between active CAs. The fatigue and leak parameters of the Y network were set in such a way that CAs ignite only when an external stimuli are present. The CAs in Y inhibited each other so that only one of the CAs in Y was active at any moment. The Explore module had excitatory connections with a small proportion of cells in module Y. These connections were distributed uniformly, and the weights did not adapt. Spontaneous activation in the Explore module could randomly trigger any of the two response CAs in module Y. The activity of the Explore module could be inhibited by the output activity from the Value module that was triggered in each trial according to probability $P(E) = \pi$ of the reinforcement event, controlled by the experimental sequence.

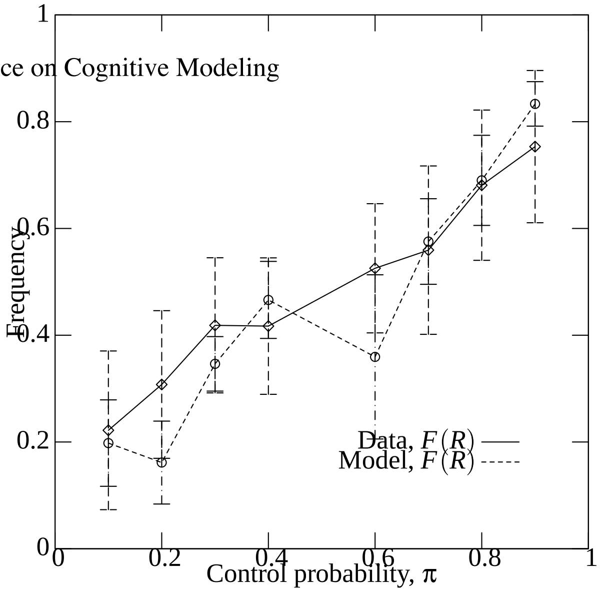


Figure 3: Comparison of response frequency produced by the CABOT model with response frequency by participants in (Friedman et al., 1964). RMSE=8.937%.

When the Explore module is inhibited by the reinforcing activity of the Value module, the active pair (x,y) is allowed to persist longer, strengthening the connections $x \to y$ relative to other connections. We found that the robustness of this effect depends on the time (i.e. number cycles) these CAs are allowed to persist. In this model, it takes approximately between 10–20 cycles for a response CA in Y to ignite, and if the Explore module is active, then the response CA may change during another 10–20 cycles. In this experiment, the system ran for 100 cycles per trial which was sufficient for the control of learning to have a robust effect. The complete code of the simulation is available online from the CABOT project website.

Results

The model was used to simulate Sessions 1 and 2 of eight 48-trial blocks each with variable control probabilities π (Friedman et al., 1964). The results comparing response frequency of the model with the data are shown on Figure 3. The model approximates the data fairly well (RMSE=8.937%) showing the probability matching effect that also overestimates and underestimates the low and high value of the control probability π respectively. Note that unlike the ACT–R model, where the estimated parameter β^{-1} in the exponential distribution was constant (Anderson & Lebiere, 1998), the activity of the Explore module randomising the response is dynamic.

Conclusions

In this paper, we discussed the CABOT architecture and some challenges associated with implementing the CA hypothesis of symbolic processing in the brain. The problem of re-use and combination of symbols, particularly in learning procedural knowledge, pointed at one significant shortcoming of the standard Hebbian learning mechanism — adaptation of weights based purely on correlations does not take into account the optimisation criteria that a system may have to satisfy. To resolve this problem, stochastic meta-control based on utility feedback was introduced into the system.

It is attractive to speculate about the existence of the Value and Explore modules in the brain. Some researchers have proposed that tonically active cholinergic neurons in the basal ganglia and striatal complex play an important role in conflict resolution and learning procedural knowledge (Granger, 2006). These neurons account for a small proportion of the connections that are quite uniform and non-topographic, and the activity of these neurons was suggested to play the role of stochastic noise, similar to the activity of cells in the Explore module (see Fig. 1). Interestingly, the activation of the tonically active cholinergic neurons is inhibited by the activation from the reward path, similar to the function of the Value module in our system. Other studies of mechanisms for exploratory behaviour in the brain are also in favour of the exponential distribution model (Daw, O'Doherty, Dayan, Seymour, & Dolan, 2006).

Setting these speculations aside, this work has demonstrated that the proposed mechanism can be used for controlling Hebbian learning in networks of relatively biologically faithful models of neurons. The mechanism allows for selective learning of connections between specialised groups of cells (CAs), and following Hebb's hypothesis it shows not only that CAs can indeed be associated with symbols, but also shows how such representations can be re-used and combined to learn new knowledge. Simulation of the probability matching effect has demonstrated that the mechanism is also a plausible cognitive model. We anticipate that the proposed architecture can also be used to model other psychological phenomena, such as the effect of reinforcement values on speed of learning, and this is one possible direction of our future research.

Acknowledgements

This work was supported by EPSRC grant EP/DO59720.

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.

Belavkin, R. V. (2003). *On emotion, learning and uncertainty: A cognitive modelling approach*. PhD thesis, The University of Nottingham, Nottingham, UK.

Belavkin, R. V. (2009). Bounds of optimal learning. In *2009 IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (pp. 199–204). Nashville, TN, USA: IEEE.

Belavkin, R. V., & Huyck, C. (2008). Emergence of rules in cell assemblies of fLIF neurons. In *The 18th European Conference on Artificial Intelligence*.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, *441*(7095), 876–879.

Friedman, M. P., Burke, C. J., Cole, M., Keller, L., Millward, R. B., & Estes, W. K. (1964). Two–choice behaviour under extended training with shifting probabilities of reinforcement. In R. C. Atkinson (Ed.), *Studies in mathematical psychology* (pp. 250–316). Stanford, CA: Stanford University Press.

Granger, R. (2006, July). Engines of the brain: The computational instruction set of human cognition. *AI Magazine*, *27*(2), 15–32.

Hebb, D. O. (1949). *The organization of behavior*. New York: John Wiley & Sons.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, *79*, 2554–8.

Huyck, C. (2007). Hierarchical cell assemblies. *Connection Science*.

Huyck, C., & Belavkin, R. V. (2006, April). Counting with neurons, rule application with nets of fatiguing leaky integrate and fire neurons. In D. Fum, F. D. Missier, & A. Stocco (Eds.), *Proceedings of the Seventh International Conference on Cognitive Modeling*. Trieste, Italy: Edizioni Goliardiche.

Kohonen, T. (1982). Self–organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Maas, W., & Bishop, C. (2001). *Pulsed neural networks*. MIT Press.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, Massachusetts: Harvard University Press.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*, 267–273.

Stratonovich, R. L. (1965). On value of information. *Izvestiya of USSR Academy of Sciences, Technical Cybernetics*, *5*, 3–12. (In Russian)

Wald, A. (1950). *Statistical decision functions*. New York: John Wiley & Sons.

Adaptive Mesh Refinement for Efficient Exploration of Cognitive Architectures and Cognitive Models

Bradley J. Best (bjbest@AdCogSys.com)
Caitlin Furjanic (cfurjanic@AdCogSys.com)
Nathan Gerhart (ngerhart@AdCogSys.com)
Adaptive Cognitive Systems, 1942 Broadway St., Suite 201
Boulder, CO, 80302 USA

Jon Fincham (Fincham@cmu.edu)
Department of Psychology, 5000 Forbes Ave.
Pittsburgh, PA, 15213 USA

Kevin A. Gluck (kevin.gluck@us.af.mil)
Glenn Gunzelmann (glenn.gunzelmann@us.af.mil)
Air Force Research Laboratory, 6030 S. Kent St.
Mesa, AZ, 85212 USA

Michael A. Krusmark (michael.krusmark@mesa.afmc.af.mil)
L-3 Communications, 6030 S. Kent St.
Mesa, AZ, 85212 USA

Abstract

The majority of cognitive models support some form of parameterization, either of the model itself, or through architectural mechanisms. In order to fully understand these models, it is important to understand the model's behavior as a result of parameter variation across a wide range of values. Even simple models become difficult to understand without a systematic method of exploring performance across parameter combinations, and scientists have turned to iterative methods to perform sweeps of these spaces. As an alternative to an exhaustive, homogeneous search, we examined adaptive mesh refinement (AMR) to explore simple and complex parameter spaces of several models developed within ACT-R. AMR allows for fewer model runs with minimal loss of information. We found that, with appropriate granularity, AMR methods can provide a sufficient computational exploration of a performance space with only 1% of the sampling of conventional, homogeneous parameter sweeps. The advantages of AMR for computationally efficient exploration of the performance predictions should be of benefit and interest to developers and users of cognitive architectures and cognitive models.

Keywords: Adaptive mesh refinement; Cognitive architecture; Cognitive model; ACT-R; Parameter sweeps; visualization

Introduction

Although many discussions of cognitive modeling focus on the degree of fit to human empirical data, the point has been compellingly made that what a cognitive model does outside of the best-fitting parameter combination is just as important as what it does at the best-fitting parameter combination, and perhaps even more so (Roberts & Pashler, 2000). Information about how a model performs outside the best-fitting parameter combination provides modelers with information about how likely it is that other parameter combinations result in a comparable fit. It also gives modelers information about the full range of behavior possible from the model and how different parameters interact to generate possibly complex behavioral dynamics. Both novice users of a cognitive architecture working to understand model dynamics, and expert users of a cognitive architecture testing modifications to the theories embedded in these architectures would stand to benefit enormously from a rapid analysis and visualization of the model performance spaces involved. However, cognitive modelers facing this problem are currently confronted with a lack of tools that support exploring that space. The de-facto approach to cognitive modeling is more often a focus on maximizing fit to human data. This is done through either hand-tuning based on the intuition and experience of the modeler or automated optimizing of the fit of a cognitive model through approaches such as genetic algorithms, the conjugate gradient methods, or any of a variety of other alternatives for optimization. Any of these approaches can be sufficiently successful, but they provide little data about the performance of the model outside of the ultimate parameter values used in presenting the final fit.

Cognitive modelers need techniques and tools to support the rapid exploration of parameter spaces in pursuit of understanding of both models and architectures, including methods that support visualization of complex spaces that illuminate model and architecture behavior in response to changes in parameters. We will describe an integrated approach to these explorations that we have developed across our previous research efforts (e.g., Best, Fincham, Gluck, Gunzelmann, & Krusmark, 2008). First, however, we will turn to a discussion of exploring parameter spaces in the context of cognitive modeling.

Our goal, in this case, is to understand how the architecture and model behave generally and at the best fitting point itself. To get a full understanding of how a model is behaving outside of the best-fitting parameter combination, one approach is to define the limits and step-sizes of a parameter space and then run a model some number of times at each parameter combination (an exhaustive, homogeneous search), where the selected number of runs is intended to provide convergence on the underlying prediction of the model and architecture. This method produces an evenly sampled space that describes the overall behavior of the model. However, resources (time, computation) are allocated evenly between informative and uninformative areas of the space. Informative areas are rich in detail relating the performance of the model or architecture to the underlying parameters. Uninformative (or less informative) areas of the space may take on a variety of different characteristics, such as a degenerate part of the space where a model produces no responses at all. The resources spent on uninformative areas are essentially wasted, as they provide little additional information. Furthermore, a reduction in granularity (step size) can result in oversampling of the parameter space; resources are wasted in this case as well. Even worse, if the model is a preliminary version or prototype, significant effort could be expended exploring a space that could quickly be deemed uninteresting (e.g., a model with a bug that produces spurious results). Adaptive mesh refinement is one technique that can be used to circumvent these issues and focus resources on high information value areas of the model and architecture space.

Adaptive Mesh Refinement

Adaptive mesh refinement (AMR) is a method that can differentially and intelligently allocate resources to areas of a parameter space that call for finer resolution in the modeling based on the presence of more local complexity (Plewa et al. 2005). Briefly, the entire n-dimensional parameter space, which is defined using some set of finite bounds, is initially divided into geometrically regular cells at a very coarse level. The value of each dependent measure the model produces at the midpoint of each cell is estimated based on the previously sampled value of the dependent measures produced at the corners. This estimated or expected value is then compared to the actual value sampled at the midpoint. If the expected and observed values at the midpoint are closer than a predetermined deviation threshold, changes in the dependent measure are estimated to change linearly across the parameter range within the cell, and the dependent values for all target parameter combinations within that cell are populated with linear interpolation based on the sampled corners and midpoint. Alternatively, if the difference between the estimated and measured values for the dependent measure(s) exceeds the threshold, the cell is divided more finely and the process is repeated with the children cells. Ultimately, this results in minimal sampling over linear portions of the space and maximal (bounded) sampling over areas that have more complex surface characteristics (e.g., curvature, variability). The stringency of the threshold chosen determines the amount of space sampled. For example, a small allowable deviation such as 1% will result in nearly complete sampling of the space, while a more lax criterion such as allowing up to 50% deviation before further refinement was pursued would result in almost none of the space being sampled. We have found that using AMR with a well chosen refinement threshold can result in a 100 fold reduction of resources expended without a corresponding reduction in the information value of the data gathered from the model parameter space, allowing for a rapid exploration of parameter spaces, thereby dramatically shortening the cognitive model revision cycle (Best et al. 2008).

AMR techniques, because they attempt to sample minimally, may produce local spikes in the data, especially when applied to stochastic models such as the ACT-R spaces described here (i.e., the means are less stable when using fewer model runs). We have found that the inclusion of smoothing as a post-process for AMR generally produces improved results, especially at lower sampling rates, since it uses information from the local neighborhood to cancel out noise present in the surface. We implemented smoothing, as is commonly done in digital image processing, by combining the AMR determined value of a dependent measure at a point in some proportion (e.g., ½ was useful in many of our experiments) with the average of its nearest neighbors on the AMR surface (Plewa et al. 2005).

As parameter spaces become larger and more complex (i.e., greater dimensionality and finer granularity), however, the required resources can prohibit exploration, even with the gains from AMR. The main reason for this is that the scaling of a parameter space is exponential, and thus even relatively simple models may easily exceed the capacity of available computational resources in a typical lab setting. In this situation, high performance computing (HPC) must be leveraged, in combination with AMR, if a timely exploration is to be performed. HPC computing typically involves a large network or cluster of computers that can perform model runs in parallel, resulting in a faster exploration of complex parameter spaces. This is especially useful in the case of cognitive model explorations, which can be described as “embarrassingly parallel”, a term used in the field of computational complexity that means that the processes to be parallelized (individual model runs) do not interact with each other (Dutra et al. 2003).

The remainder of our presentation will focus on applying AMR to a set of task models of increasing complexity, demonstrating the utility of AMR and the value of parameter exploration for understanding cognitive models. The three tasks we will describe are the Paired Associates Task (PAT), taken directly from the ACT-R tutorial units (ACT-R Tutorials, 2009), the Psychomotor Vigilance Test (PVT; Dinges & Powell, 1985), and the Walter Reed Serial

Addition and Subtraction Task (SAST; Thorne et al., 1985). We now turn to these models and an exploration of their parameter spaces using AMR and HPC.

Parameter Space Descriptions

The Paired Associates Task, as described in Anderson (1981) is a learning task that involves presentation of 20 nouns associated with the digits 0-9. The pairs are presented once during a study session and then presented 7 times during a testing session. The participant is scored on latency to correct response and proportion of correct responses out of the 20 pairs for each of the 7 presentations.

This task is used as the target of a modeling unit in the ACT-R tutorials where the focus is on understanding the interactions of parameters related to activation in producing the memory behavior of the ACT-R architecture (and its corresponding explanation of human memory). However, the modeling task itself poses a challenge to the novice cognitive modeler, and prospective modelers may leave the tutorial unit unsure of the interactions of the parameters, and possibly even somewhat frustrated. We thus chose this model as a target to see if the methods we have developed could be quickly applied to aid in understanding the behavior of the architecture and model of this task.

In the ACT-R architecture, the latency of a retrieval from declarative memory is impacted by the activation of chunks, where that activation is a product of its base level activation and a noise factor. The activation is also impacted by the rate of decay in declarative memory, while the ability to retrieve activated chunks is impacted by the retrieval threshold, which determines an activation level below which chunks cannot be retrieved. Of these parameters, the base level learning parameter is typically left at a default value, leaving us three parameters to choose from for this exploration. Their behavior is given by the following equations. The first equation relates the retrieval time to *A,* the activation of a chunk, and *F*, the latency factor, while the second equation relates the probability of recall for a chunk to the retrieval threshold, *τ*, the activation of the chunk, *A,*and the noise parameter of the system, *s*.

$$Time = Fe^{-A}$$

$$P(retrieval)_{Chunk_i} = \frac{1}{1 + e^{\tau - A_i / s}}$$

To allow for easy visualization, we chose to focus on only two of these remaining parameters, fixing the noise parameter *s* at 0.5, and exploring the PAT space by varying the parameters for the retrieval threshold (*τ*) and the latency factor (*F*), as suggested in the tutorial instructions (ACT-R Tutorials, 2009). We explored levels of *τ* from -3 to 0 with a step size of 0.25 and levels of *F* from 0 to 0.45 with a step-size of 0.025, resulting in a space with 13 levels of *τ*, 19 levels of *F*, and a total of 247 parameter combinations.

Our general approach to understanding the efficiency and effectiveness of AMR methods, which we also used below with the PVT and SAST models, is to first collect 100 model runs at each parameter combination, and then divide these into a “train” and “test” portion of the data. The comparison of these two halves provides a baseline estimate of how well the data fit themselves (model stability), which can be expressed as a baseline Root Mean Square Error (RMSE). AMR variants can then be compared against this baseline to see what additional error, if any, they produce.

Our exploration was conducted using software written to run the ACT-R models and collate the results automatically, allowing the experimenter to initialize experimental settings and then leave the software to continue unaided. The resulting data are then imported into R, which we used, or an alternative statistical analysis and visualization package.

Our focus is on AMR methods, but to demonstrate the efficiency gain these methods can produce, we also conducted an exhaustive homogeneous sweep of the parameter space for comparison. Our hypothesis is that the same scientific conclusions would be reached with either method, one using a fraction of the computational resources, and thus one source of evidence for this hypothesis will be in the quality of the conclusions a modeler might come to viewing the different diagrams. For this purpose, we will present an exhaustively sampled space, labeled “fully explored” (figure 1), and a minimally sampled space that uses AMR to the full extent possible to reduce computation, labeled “minimally explored” (figure 2). In addition, we also present a visualization of the results of the smoothing post-process (figure 3).

Figures 1-3 are of the latency for the 8th simulated recall trial during the PAT, labeled “t8lat DV”, which we selected for presentation based on the obvious interaction between *τ* and *F*. The gray spheres represent parameter combinations at which models were run.

These figures show that increasing the latency factor produces a predominantly linear increase in reaction times when the retrieval threshold is less than approximately -2, but that higher values of the retrieval threshold (closer to 0) produce an interaction with the latency factor. In particular, the latency for retrievals decreases at higher values of *τ*, since more active chunks are retrieved more quickly or, in cases when a failure to retrieve a chunk happens, the recognition that this is the case happens faster.

It is hard to imagine how a novice modeler might come to understand this space by manually entering parameter values and attempting to understand the rows of data that result, and thus for this reason alone we might suppose that the use of these methods is desirable. Further, the qualitative conclusion that can be reached comparing the smoothed AMR results (figure 3) to the exhaustive results (figure 1) is obvious: the smoothed AMR surface contains much of the qualitative detail of the exhaustively sampled surface, but at a fraction of the computational cost, having been produced using only 1% of the runs present in the exhaustive graph.

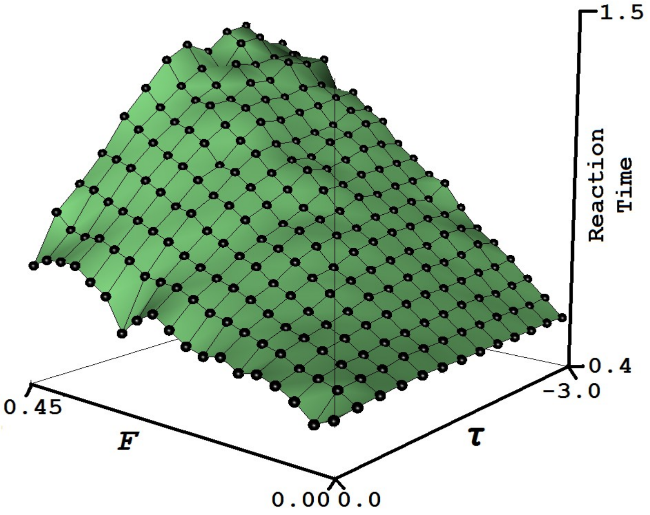


Figure 1: Fully explored parameter space

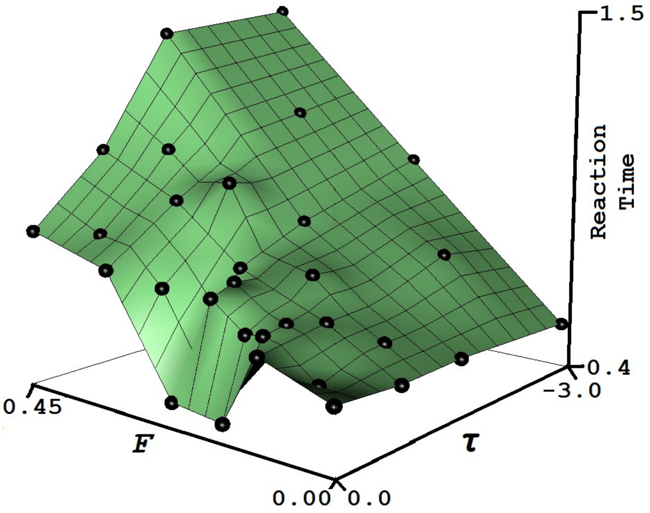


Figure 2: Minimally explored parameter space

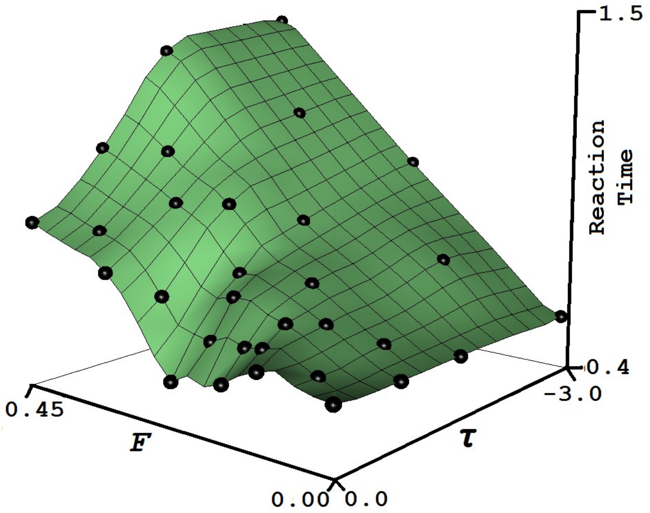


Figure 3: Minimally explored parameter space - smoothed

The question, then, is this: What is the gain of using AMR in terms of computational resources as it relates to any corresponding loss in fidelity? We will now attempt to answer that question both quantitatively and comprehensively in the context of the three tasks we have worked with: the PAT, PVT and SAST. First, however, we will provide a brief background on these two new tasks.

The Psychomotor Vigilance Test involves the presentation of a stimulus at known locations, but at random time intervals, and measuring the time it takes the subject to respond to that stimulus. Responses are binned into 20ms intervals with false starts defined as reaction times faster than 150 ms, lapses as reaction times slower than 500 ms, and sleep attacks as reaction times slower than 30 s. This task, due to its cognitive simplicity and sensitivity to the effects of sleep deprivation and circadian rhythm, is commonly used to assess the impact of fatigue (e.g., Dinges & Powell, 1985; Van Dongen & Dinges, 2005).

The Walter Reed Serial Addition/Subtraction Task involves presenting two single-digit numbers in sequence, followed by an operator – either a plus sign or minus sign. After performing the operation, participants respond with the ones digit of the answer, or the answer plus 10 if the result is negative. Time to correct responses and the percent of correct responses are measured.

As we did with the PAT, these tasks were evaluated within the framework of AMR to determine the impact of AMR methods on accuracy and reduction of computational demands. All of the AMR methods were compared to a corresponding exhaustive parameter sweep, where the exhaustive sweep used 100 model runs at each combination to establish a baseline: the exhaustive data were split in half and compared to determine how well the data fit themselves. This produced a baseline Root Mean Square Error (RMSE) for the model runs against which AMR runs were then compared. In addition, this allowed for an efficiency metric which was simply the percent of the "full space" that was explored by an AMR variant (% Space Sampled). The "full space" is one of the baseline halves and is composed of 50 model runs at each parameter combination. Finally, we also report the total number of model runs involved in each of the spaces and AMR variants. We tested several variations of AMR and smoothing using this methodology. In particular, we examined: 1) allowing the number of model runs to vary as a property of local variation or fixing them at some particular *n*, 2) using local error bounds based on one or all dependent measures, 3) determining local error in dependent measure prediction based on absolute, relative, or statistical criteria, 4) the impact of modifying the smoothing radius and intensity, and 5) the impact of using 4-neighbors vs. 8-neighbors in smoothing. Here we will only report specific instances due to space limitations.

The PVT and SAST spaces have been used to explore the ability of modifications to the ACT-R architecture to account for the pattern of deficits exhibited by people under conditions of extended wakefulness (e.g., Gunzelmann et al., 2007). These modifications include parameterized mechanisms, which require careful exploration to provide an understanding of their potential impacts. The PVT space was explored using 4 parameters, with the chosen granularity of these parameters resulting in a parameter space with 56,511 parameter combinations. Models were run at each combination for the exhaustive parameter sweep. Similarly, the SAST space was explored by varying 7 parameters, with a necessarily coarser granularity (to partially offset the higher dimensionality) that resulted in a parameter space with a total of 129,600 parameter combinations. Models were run at each of these combinations for the exhaustive parameter sweep.

Table 1: Algorithm Performance Summary

	Data Set	% Control RMSE	% Space Sampled	Total Model Runs
~ 1% Space Sampled	PAT	209.26%	1.07%	132
	PVT	226.45%	1.32%	37,335
	SAST	533.19%	1.09%	70,479
~ 10% Space Sampled	PAT	113.94%	10.20%	1,260
	PVT	154.81%	8.40%	237,350
	SAST	167.55%	9.72%	630,020
100% Space Sampled	PAT	100.00%	100.00%	12,350
	PVT	100.00%	100.00%	2,825,550
	SAST	100.00%	100.00%	6,480,000

In general, with only 10% of the space sampled, for the worst case additional error was only 67.55% beyond the error in the original data when compared to themselves. The granularity of the sampling, however, did interact, and the SAST model, despite having the largest parameter space, also had the coarsest minimum granularity. That is, the SAST has only 6 levels per IV, so not much processing can be skipped, and skipping removes information. The result of this was that, at very sparse sampling of ~1%, the AMR algorithm never proceeded much beyond the initial AMR corners, producing a very rough approximation for SAST. The PVT space granularity fell in the middle of the PAT and SAST spaces, and allowed for dramatic compression with very little loss of accuracy. In particular, in those spaces the error was approximately only doubled (~200% RMSE) when compared to baseline at a very minimal sampling of approximately 1% of the data sampled. This represents a two order of magnitude gain in time to get an answer that, while approximate, is most likely extremely useful (and might, in the case of faulty models, obviate the need for ever collecting the other 99% of the data).

Taken as a whole, algorithm performance is fairly similar across spaces despite dramatic differences in the size of the model spaces. That is, the SAST space is several orders of magnitude larger than the PAT space, but the error terms are within an order of magnitude.

For all three parameter spaces, we explored the effects of performing the homogeneous sweep with a reduced number of model runs. These data are not presented due to space limitations. In all cases, however, AMR methods provided superior results. For example, running 2 models at each parameter combination results in reducing the space sampled to 4%. AMR methods using only 2 model runs result in less space sampled and are more accurate as well.

We also explored adaptively changing the number of model runs at each parameter combination based on measures of local variation. This method ultimately results in focusing computational resources on portions of the space where the model returns spurious results. Increased model runs in these areas does not result in a superior understanding of the model; AMR methods predict these noisy areas more efficiently through linear interpolation.

Conclusions

In this paper, we have demonstrated the application of AMR to a variety of modeling contexts, showing both the visualizations that can be produced and the gains in computational efficiency achieved through this method. In the case of the PAT, the AMR exploration brought out a nonlinear interaction that would most likely not be obvious from a set of tabled values, and would almost certainly be missed by a novice modeler. However, through applying AMR to this task model, we were quickly able to visualize and understand the underlying model and architecture dynamics as a result of examining the impact of varying the parameters that control the model and architecture. This simply cannot be achieved by examining the fit of a model at a particular point in a parameter space.

The PAT could certainly be approached by hand modifying the models in a desktop environment, as it is during the ACT-R tutorials, or even through an exhaustive iterative sweep of the parameter space, but we make the case here that the AMR methods can produce superior understanding with little to no extra investment in computational resources, and thus they are clearly preferable to the alternatives.

As parameter spaces become larger and more complex (i.e., greater dimensionality and finer granularity), the resources required to enumerate or sample from them can become prohibitive, even with the gains from AMR. The reason for this is that the scaling of a parameter space is exponential, and thus even relatively simple models may easily exceed the capacity of available computational resources in a typical lab setting. It is evident that the number of model runs, as reported in Table 1, is a proxy for time. While Moore's Law was once considered a potential way out of computing bottlenecks – simply waiting for faster processors to arrive could solve some issues – that simply does not apply to problems that scale exponentially. Further, though processors are increasing in speed, our cognitive architectures, models, and the task domains we are interested in are also increasing in complexity, and these effects largely cancel each other out. Thus, it is necessary both to improve the efficiency of our methods through approaches such as AMR, and also to leverage resources that combine processors, such as High Performance Computing (HPC). HPC typically involves a large network or cluster of computers that can perform model runs in parallel, resulting in a faster exploration of complex parameter spaces. This is especially useful in the case of cognitive model explorations, which can be described as "embarrassingly parallel", a term used in the field of computational complexity that means that the processes to be parallelized (individual model runs) do not interact with each other (Dutra 2003).

Fortuitously, these methods also provide a natural gateway to solving harder computational problems: a problem formulated for AMR solution and visualization in the desktop environment is already formulated for HPC solution and visualization.

The techniques described here demonstrate effective ways for exploring large parameter spaces. Indeed, the work described here could not have been conducted without these techniques. This is not to say, however, that the underlying exponential nature of cognitive modeling problems has been tamed. Rather, the methods here provide a significant amount of leverage to a scientist who has managed to reduce the effectively infinite space of cognitive models to a manageable size.

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was funded by the Air Force Research Laboratory's Warfighter Readiness Research Division in Mesa, Arizona.

References

ACT-R Tutorials (2009). Retrieved from the ACT-R Web site: http://act-r.psy.cmu.edu/actr6/.

Anderson, J.R. (1981). Interference: The relationship between response latency and response accuracy. *Journal of Experimental Psychology: Human Learning and Memory*, 7, 326-343.

Berger, M., & Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53, 484-512.

Best, B., Fincham, J., Gluck, K., Gunzelmann, G., & Krusmark, M. (2008). Efficient Use of Large-Scale Computational Resources. In J. Hansberger (Ed.), *Proceedings of the Seventeenth Conference on Behavior Representation in Modeling and Simulation* (pp. 180-181). Orlando, FL: Simulation Interoperability Standards Organization.

Brooke, J.M., Marsh, J., Pettifer, S., and Sastry, L.S (2007). The importance of locality in the visualization of large datasets. *Concurrency and computation: practice and experience*, 19:195–205.

Dinges, D. F., & Powell, J. W. (1985). Microcomputer analyses of performance on a portable, simple visual RT task during sustained operations. *Behavior Research Methods, Instruments, & Computers 17(6)*, 652-655.

Dutra, I. C., Page, D. Santos Costa, V. Shavlik, J.W. and Waddell, M. (2003). Towards automatic management of embarrassingly parallel applications. In *Proceedings of Europar 2003*, Lecture Notes in Computer Science. Klagenfurt, Austria: Springer Verlag.

Gluck, K., Scheutz, M., Gunzelmann, G., Harris, J., & Kershner, J. (2007). Combinatorics meets processing power: Large-scale computational resources for BRIMS. *Proceedings of the Sixteenth Conference on Behavior Representation in Modeling and Simulation* (pp. 73-83). Orlando, FL: Simulation Interoperability Standards Organization.

Gunzelmann, G., & Gluck, K. A. (2008). Approaches to modeling the effects of fatigue on cognitive performance. In J. Hansberger (Ed.), *Proceedings of the Seventeenth Conference on Behavior Representation in Modeling and Simulation* (pp. 136-145). Orlando, FL: Simulation Interoperability Standards Organization

Gunzelmann, G., Gluck, K. A., Kershner, J., Van Dongen, H. P. A., & Dinges, D. F. (2007). Understanding decrements in knowledge access resulting from increased fatigue. In D. S. McNamara and J. G. Trafton (Eds.), *Proceedings of the Twenty-Ninth Annual Meeting of the Cognitive Science Society* (pp. 329-334). Austin, TX: Cognitive Science Society.

Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (in press). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*.

Plewa, T., Linde, T.J., and Weirs, V.G. (2005). *Adaptive mesh refinement, theory and applications: proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, Sept. 3-5, 2003.* Springer Verlag.

Rai, M. M., & Anderson, D. (1981). Grid evolution in time asymptotic problems. *Journal of Computational Physics*, 43, 327-344.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, *107*, 358-367.

Thorne, D. R., Genser, S. G., Sing, G. C., & Hegge, F. W. (1985). The Walter Reed performance assessment battery. *Neurobehaviora. Toxicology and Teratology*, 7, 415-418.

Van Dongen, H. P. A., & Dinges, D. F. (2005). Sleep, Circadian Rhythms, and Psychomotor Vigilance. *Clinical Sports Medicine, 24*, 237-249.

Willett, R. (2003, September 24). *Sampling Theory and Spline Interpolation*. Retrieved from the Connexions Web site: http://cnx.org/content/m11126/2.3/

Testing fMRI Predictions of a Dual-Task Interference Model

Jelmer P. Borst (jpborst@ai.rug.nl)[1,2]
Niels A. Taatgen (taatgen@cmu.edu)[1,2]
Hedderik van Rijn (d.h.van.rijn@rug.nl)[1]
Andrea Stocco (stocco@cmu.edu)[2]
Jon M. Fincham (fincham@cmu.edu)[2]
[1]Dept. of Artificial Intelligence, University of Groningen, The Netherlands
[2]Dept. of Psychology, Carnegie Mellon University, USA

Abstract

A previously developed ACT-R/threaded cognition model of dual-task interference (Borst, Taatgen & Van Rijn, 2009) was used to predict neuroimaging data in four brain areas. These predictions were tested in an fMRI experiment, which confirmed the predictions in three of the areas. The fourth area, the intraparietal sulcus, showed a different pattern than predicted. To account for this, a new mapping of an ACT-R module onto a brain area was introduced: It was assumed that activation in the intraparietal sulcus not only depends on the problem state module, as is customary, but also on the visual-location module. The resulting model fit well to the human data, confirming the model's assumptions of dual-task interference.

Keywords: fMRI, ACT-R, Problem State, Multitasking.

Introduction

Some tasks can be performed together effortlessly, like drinking coffee and listening to a talk, while other tasks interfere with each other, like talking to a colleague while writing a paper. The challenge for theories of multitasking is to explain why some tasks interfere with each other and some do not. Intuitively this is easy to explain: if tasks use the same cognitive resources they will probably interfere. This idea was formally implemented in the threaded cognition theory (Salvucci & Taatgen, 2008). In threaded cognition, multiple tasks (called 'threads') are active at the same time. Tasks can use several cognitive resources, like declarative memory and the visual system. These resources function in parallel (i.e., the visual resource can be used to perceive an object, while at the same time a fact can be retrieved from memory), but the resources themselves can only proceed in a serial fashion (i.e. the visual resource can only perceive one object at a time). Thus, if multiple tasks need the same resource, one of the tasks will have to wait for the other tasks, resulting in interference.

Salvucci and Taatgen (2008) have shown that, in addition to perceptual and motor resources, two central cognitive resources cause interference in multitasking: declarative and procedural memory. Additionally, we have shown that another central cognitive resource, the problem state, also causes interference in multitasking (Borst & Taatgen, 2007; Borst, Taatgen, & Van Rijn, 2009). The problem state is used to maintain mental representations necessary for performing a task. For instance, when solving '2*x*-7=6' the problem state is used to store the intermediate solution '2*x*=13'. In our previous research, we let participants perform a subtraction and text entry task concurrently. Both tasks were presented in two versions: an easy version in which no problem state was required to perform the task and a hard version in which it was. When both tasks required a problem state, significantly more interference was observed than in all other conditions: response times and error rate increased. To account for these results a cognitive model was developed using threaded cognition and ACT-R (Anderson, 2007).

In the current paper we set out to validate this model using neuroimaging data. First, the previously developed model was used to predict brain activation patterns in four brain regions. Subsequently, these predictions were tested in an fMRI experiment. Before we discuss these points, we will first explain how ACT-R models can be used to predict neuroimaging data.

Using ACT-R to predict the BOLD response

ACT-R (Anderson, 2007) describes human cognition as a set of independent modules that interact through a central production system. For instance, it uses a visual module for perception and a motor module to interact with the world. Besides these peripheral modules, there are several central cognitive modules: the procedural module that implements the central production system, the declarative memory module, the goal module, and the problem state module (sometimes called 'imaginal module'). All modules operate in parallel, but a module in itself can only proceed serially.

ACT-R models are usually tested on a behavioral level: if for instance reaction times and error patterns match the human data, it is concluded that a model gives a plausible account of the observed behavior. However, to find direct evidence for non-observable specifics of models, ACT-R has been extended to predict neuroimaging data (Anderson, 2005). To predict brain activation data, or to be more precise, the Blood Oxygenation Level-Dependent (BOLD) contrast, the modules of ACT-R have been mapped onto small regions in the brain (about 12x12x12mm). The most important modules and associated brain regions for this study are listed in Table 1.

The different modules are not constantly in use during the execution of an ACT-R model, but operate for short periods of time (in the order of hundreds of ms). It is assumed that when a module is active, it will drive a BOLD response in the associated brain region. This response is modeled by a gamma function, as is customary in fMRI research:

$$H(t) = m\left(\frac{t}{s}\right)^{a} e^{-(t/s)}$$

Table 1. ACT-R modules and associated brain regions.

ACT-R Module	*Brain Region*	*MNI Coordinates*
Manual	Precentral gyrus (BA 3)	-37, -28, 51
Visual	Fusiform gyrus (BA 37)	-22, -59, -15
Declarative Memory	Inferior frontal sulcus (BA 45/46)	-42, 22, 21
Problem State	Intraparietal sulcus (BA 7/39/40)	-23, -67, 36

where *m* determines the magnitude of the BOLD curve, *s* the time scale, and *a* the shape. If *D(t)* is a 0-1 demand function that indicates whether a module is active at time *t*, the BOLD function can be calculated by convolving *D(t)* with the gamma function:

$$B(t) = \int_0^t D(x)H(t-x)\mathrm{d}x$$

It should be noted that we do not assume that modules in ACT-R exclusively drive activation in these regions, nor that activation in these regions is only due to the associated ACT-R modules. However, these regions have been the best indicators of activation in the ACT-R modules over a series of studies (see also Anderson, 2007).

Predicting the BOLD response

In this section we will describe how we used the model of Borst et al. (2009) to generate BOLD predictions. We will first describe the task in detail, followed by the model and the predictions.

The task

In the experiment participants had to perform a subtraction and text entry task concurrently (Fig. 1). Both tasks had two versions, an easy version in which participants did not have to maintain a problem state between responses, and a hard version in which they were required to maintain a problem state. Participants had to alternate between the tasks: after entering a number, the subtraction task was disabled, forcing participants to subsequently enter a letter. After entering a letter, the text entry task was disabled and the subtraction task became available again, etc.

In the subtraction task, 6-digit column subtraction problems had to be solved in right-to-left order. In the easy, no problem state version, the upper term was always larger or equal to the lower term; these problems could be solved without 'borrowing'. In contrast, the hard version required participants to borrow 3 times (see Fig. 1). The assumption is that participants used their problem state resource to keep track of whether a 'borrowing' was in progress. Solved columns were masked with #-marks to prevent display-based strategies (i.e. reading previous columns again).

For the text entry task, 6-letter words had to be entered. In the easy version the words were presented one letter at a time. Participants had to click the corresponding button on the keypad, after which the next letter appeared. In the hard version, a word appeared at the start of a trial. When a participant clicked on the first letter, the word disappeared and had to be entered without feedback (participants could neither see the word they were entering, nor how many letters they had entered). It was assumed that participants needed a problem state to keep track of the word and their position within the word ('public, 4th position').

Before each trial, two colored circles were presented on the screen, one on the left and one on the right side, indicating whether the task on that side of the screen was going to be easy (green circle) or hard (red circle). Participants were instructed to act both quickly and accurately. The tasks were performed in all difficulty combinations: easy subtraction/easy text entry, hard/easy, easy/hard, and hard/hard.

Three changes were made with respect to the original task of Borst et al. (2009) to make it suitable for the fMRI scanner: a) letting participants respond using a mouse instead of the keyboard, b) changing the length of the stimuli from 10 to 6 numbers / characters, and c) making the interface more compact to minimize head movement.

The model

We will now describe the ACT-R/threaded cognition model that Borst et al. (2009) developed to account for the task above. Of particular importance for the tasks at hand is ACT-R's problem state module. This module can hold a problem state consisting of one chunk of information, which means that the module's contents have to be replaced frequently when it is required by multiple tasks. A problem state is accessible at no time cost, but replacing a problem state takes 200 ms. If the problem state is replaced, the previous problem state is automatically moved to declarative memory. Thus, the total time to replace a problem state is 200 ms plus the time it takes to retrieve the problem state from memory. Therefore, the problem state resource constitutes a bottleneck in multitasking: switching problem states incurs a considerable time cost.

The two tasks in the experiment were implemented as two threads in the model. Both threads use the visual module to perceive the stimuli and the manual module to operate the mouse and the keyboard. In the easy version of the

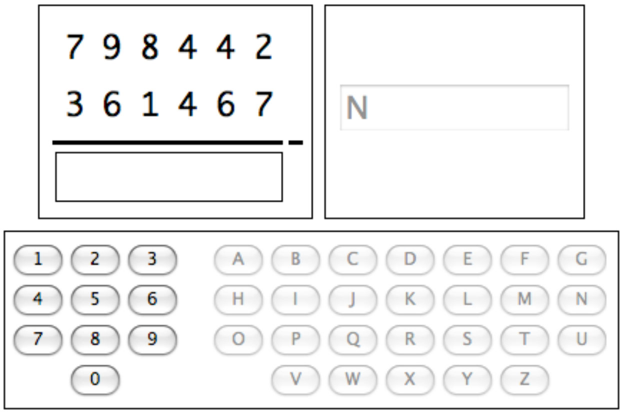


Figure 1. Screenshot of the experiment.

subtraction task, the model perceives the numbers, retrieves a fact from memory (e.g., 5–2=3) and enters the difference. In the hard version the model also starts by retrieving a fact from memory, if its outcome is negative (e.g., 3-6=-3) the model adds 10 to the upper term, stores in its problem state that a 'borrowing' is in progress, and retrieves a new fact (13–6=7). If the problem state indicates that a 'borrowing' is in progress, the model subtracts 1 from the upper term before the initial retrieval.

In the easy version of the text entry task, the model perceives the letter and clicks on the corresponding button. In the hard version, the model has to recall for each response what the target word is and what the current position is within the word: it uses the problem state resource to store the word and the current position ('public, 4th position'). If it is in the hard condition, the model does not look at the display, but uses the word and position in its problem state. However, before it can enter a letter, it first has to retrieve an order fact to determine what the next letter is. After entering a letter, the model updates its problem state to reflect that it is one position further in the word.

Because the model only needs multiple problem states in the hard/hard condition, and either zero (easy/easy) or one (easy/hard, hard/easy) in the other conditions, it predicts an over-additive effect of task difficulty on response times and accuracy. Constantly replacing the problem state in the hard/hard condition incurs a time cost, resulting in increased response times; furthermore, incorrect problem states are sometimes retrieved, resulting in errors. This model was used to generate BOLD predictions for the task, which we will describe next.

A priori BOLD predictions

As explained above, the different modules of ACT-R have been mapped onto brain regions. After changing the model to work with the new interface of the experiment (i.e. using the mouse instead of the keyboard), we generated predictions for four predefined regions. For these predictions we set the *a* and *s* parameters in the BOLD equation to 4 and 1.2, respectively. These are customary values in the literature, and as we did not fit our model to the fMRI data but predicted the data beforehand, there was no reason to alter these values. For the same reason the *m*-parameter was not used for scaling, but left at 1. We will discuss the four most important predictions of our model: the manual module, the visual module, the problem state module, and the declarative memory module. The results are displayed in Figure 2; each panel shows the BOLD response over a complete trial (entering 6 letters and numbers).

The predictions for the manual area, part of the precentral gyrus, are displayed in Figure 2A. While in all conditions the same number of responses has to be given, there are clear differences in the model predictions. This is caused by the fact that the individual responses in the more difficult conditions are spaced further apart in time (i.e., response times are higher). Consequently, the BOLD response has more time to decay between each response, resulting in longer but lower activation curves. This is in line with the fact that the area under the curve should be equal in all conditions, as it is proportional to the total time a module is active (Anderson, 2005), which is the same in each condition.

For the visual module a similar pattern can be observed (Fig. 2B). However, here the hard subtraction/easy text entry and the easy subtraction/hard text entry conditions are switched. This is caused by two things: first, when text entry is hard, the model does not have to look at the screen to see what it has to enter, but already knows the word it is entering. Therefore, less visual processing is required in the hard text entry conditions as compared to easy text entry. Second, in the hard subtraction conditions, the model does more visual processing: after noticing that it has to borrow (by reading the upper and lower terms), it reads the upper term again to process the borrowing, and afterwards reads the lower term again to come up with the final response.

Figure 2C shows the predictions for the problem state module. In the easy/easy condition the model does not use any kind of problem state, which accounts for the flat line. In both the easy/hard and the hard/easy conditions an intermediate activity level is predicted as a problem state has to be maintained for one of the tasks. In the hard/hard condition, the problem state has to be replaced on every step in a trial, because both tasks need to maintain a problem state. Thus, we expect much more activation in the hard/hard condition as compared to all other conditions: resulting in an over-additive interaction effect.

A related interaction effect can be observed for the declarative memory module (Fig. 2D). In the easy/easy condition, the model only needs to retrieve simple subtraction facts, which are extremely fast retrievals,

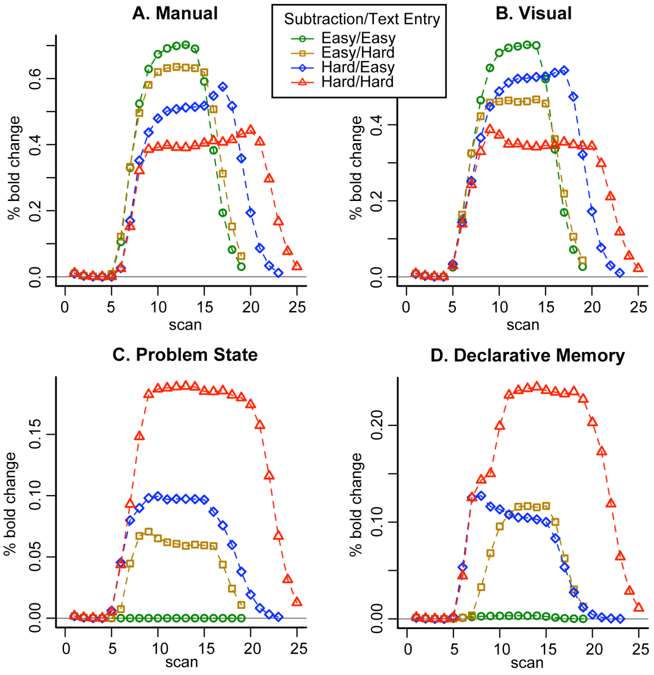


Figure 2. The BOLD predictions. 1 scan is 2 seconds.

resulting in almost no BOLD activity. In the easy subtraction/hard text entry condition, the model needs to retrieve both simple subtraction facts and facts about letter order in words, resulting in higher activation levels. In the hard subtraction/easy text entry condition the model needs to retrieve multiple subtraction facts on most of the steps in a trial, again predicting higher activation levels. In the hard/hard condition there is by far the most activation predicted, as not only the subtraction facts and letter order facts have to be retrieved, but also a problem state on each step.

To summarize, the model predicts *lower* but more persistent activation levels for the harder conditions in the visual and manual modules, and *higher* activation levels for the harder conditions in the problem state and declarative memory modules. We will now describe the fMRI experiment we carried out to test these predictions.

The Experiment

Ten students from Carnegie Mellon University participated in the experiment. Because one of them had abnormal brain anatomy, 9 datasets are left for analysis (2 female, average age 22, range 19-24, right-handed). Informed consent as approved by the Institutional Review Boards at Carnegie Mellon University and the University of Pittsburgh was given before the experiment. Participants received $65.

The 6-digit subtraction problems were generated anew for each participant. In the hard version, each subtraction problem featured 3 columns in which participants had to ‘borrow’, answers were always 6 digits long. The words in the hard text entry condition were handpicked from a list of high frequent 6 letter words (CELEX database) to ensure that similarities between words were kept at a minimum. These stimuli were also used in the easy text entry task, except that the letters within the words were scrambled to create nonsense letter strings, under the constraint that a letter never appeared twice in a row.

Each trial started with the presentation of a fixation cross, followed by two circles indicating the difficulty levels of the tasks, to avoid measuring ‘surprise-reactions’. The circles stayed on the screen for 5 seconds, after which the fixation cross was displayed again for 1 second. Afterwards, the subtraction and text entry tasks were presented. Participants had to start with the subtraction task, after which they had to alternate between the tasks. After entering the last response in each task, a feedback screen was shown for 3 seconds, indicating how many letters / numbers were entered correctly. Between trials there was a 13-17 second break, sampled from a uniform distribution. The start of the circles was aligned to the start of a scan, as was the start of the subtraction and text entry tasks.

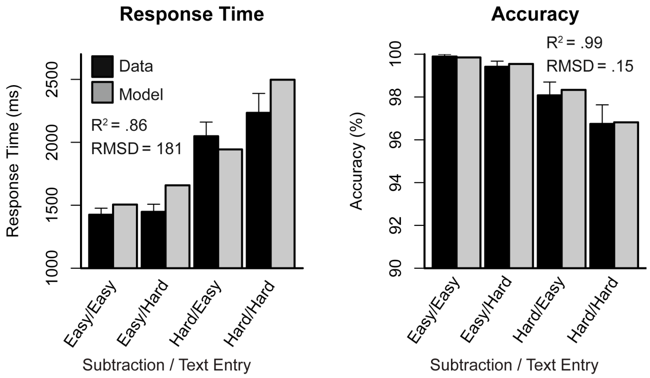


Figure 3. Behavioral results and model predictions.

The experiment consisted of one practice block and six experimental blocks. The practice block was administered during the structural scanning, to familiarize participants with performing the task in the scanner. All blocks consisted of 12 trials, 3 per condition, fully randomized. Thus, the complete experiment consisted of 72 trials. On the day before the scan day, participants practiced the experiment for approximately 30 minutes outside the scanner.

Results

Only the data of the experimental phase were analyzed. Outliers in response times faster than 250 ms and slower than 9000 ms were removed from the data, after which we removed data exceeding 3 standard deviations from the mean per condition per participant (in total, 2.2% of the data was removed). All *F*- and *p*-values are from repeated-measure ANOVAs, all error bars depict standard error.

The left panel of Figure 3 shows the average response time per condition; black bars depict experimental data, grey bars model data. Response times are measured as the time between two mouse-clicks, that is, the time it takes to give a response after having given the previous response. First responses of each task were removed. An ANOVA revealed a significant interaction effect of Subtraction and Text Entry Difficulty (*F*(1,8)=6.1, *p*=.04). A subsequent simple effects analysis showed significant effects of Subtraction Difficulty when text entry was easy (*F*(1,8)=12.04, *p*<.01), and of Subtraction Difficulty when text entry was hard (*F*(1,8) = 29.4, *p*<.001). The simple effects of Text Entry Difficulty did not reach significance. Thus, response times increase with subtraction difficulty, but even more when text entry was hard as well. The right panel of Figure 3 shows the accuracy data. No significant effects were observed, which is probably due to the low statistical power caused by the small number of participants, as such effects were observed in previous studies.

The results are in line with our previous findings (Borst, et al., 2009) and with our hypothesis. However, the effects are slightly smaller than observed previously.

The modeling results are displayed alongside the data in Figure 3. The model predicted an over-additive interaction effect because only one problem state can be maintained at a time. This was indeed observed in the data. However, the model predicted a slightly larger effect, as it was fitted on the data of the previous experiment.

Imaging data: confirmatory analysis

The results in the left precentral gyrus, associated with the manual module, are shown in Figure 4A. The data resemble the model closely: the easier the condition, the higher and broader the BOLD curve. This is explained by the fact that the responses are spaced further apart in the harder conditions, letting the activation decay between responses.

Figure 4B displays the BOLD responses in the fusiform gyrus, associated with the visual module. Again, higher activation levels were found for the easier conditions. The model predicted this, but it also predicted that the hard/easy and easy/hard conditions would switch position as compared to the manual module. While they are closer together, they did not switch completely. Presumably, the participants make less strict eye-movements than our model, and do more visual processing in the hard text entry conditions than predicted.

In Figure 4C the results of the intraparietal sulcus (associated with the problem state module) are shown. As the area under the curves is proportional to the total time a module is engaged (Anderson, 2005), most activation is observed in the hard/hard condition, as the model predicted. However, the model obviously predicted a much larger effect, with a clear interaction effect between conditions.

Finally, Figure 4D shows the activation in an area close to the inferior frontal sulcus, associated with the declarative memory module. Because four of our participants showed a negative BOLD response in the original area, we slightly changed the region to a nearby area where all our participants showed a positive BOLD response. This region, centered at *x*=-48, *y*=30, *z*=30, shows a response that roughly shows the same effects as our model: almost no activation in the easy/easy condition, and an increasing BOLD response with increasing difficulty. However, the effects were not as large as predicted.

To summarize, we confirmed our main predictions that there are higher activation levels in the easier conditions in the visual and manual regions, and that an opposite effect can be observed in the problem state and declarative memory regions. However, the BOLD response in the problem state region was different from the predictions, and the effect in the declarative memory module was less pronounced.

Imaging data: exploratory analysis

Besides the confirmatory analysis, we also performed an exploratory analysis of the fMRI data. The results are shown in Table 2. At the top, regions are shown that were more active in the hard subtraction condition as compared to the easy subtraction condition (uncorrected *p*-value < 0.001 and contiguous voxel size > 20). First of all, we found a region around the intraparietal sulcus to be active both in the left and the right hemisphere. This region corresponds to the horizontal segment of the intraparietal sulcus (HIPS), which is an important circuit for numeric processing. Next, we found two regions around the right middle frontal gyrus that responded more in the hard subtraction condition than in the easy condition. The more anterior region partly overlaps with ACT-R’s declarative memory region. These regions conform to our expectations of more memory retrievals in the harder subtraction condition. The largest active region was found in the medial frontal cortex. It is known that this region is involved in cognitive control and decision making. Not surprisingly, participants need more extensive cognitive control in the hard subtraction condition, as they have to keep track of steps in the borrowing process.

At the bottom of Table 2 regions are shown that are more active in the hard text entry condition as compared to the easy text entry condition (uncorrected *p*-value < 0.01 and contiguous voxel size > 20). More activation was found in the medial frontal cortex and the intraparietal sulcus; both regions partly overlap with the regions we found for the subtraction task. However, the region in the medial frontal cortex is more posterior and superior, and the parietal region is more central and was only found in the left hemisphere.

Table 2. Results of the exploratory analysis.

Region	*Size in Voxels*	*MNI coordinates (x,y,z)*
Hard Subtraction > Easy Subtraction (*p* < .001)		
Right Intraparietal Sulcus	102	36, -36, 33
Right Middle Frontal Gyrus	56	39, 36, 24
Medial Frontal Cortex	113	-3, 18, 48
Left Intraparietal Sulcus	41	-45, -42, 39
Right Middle Frontal Gyrus	49	27, 12, 57
Hard Text Entry > Easy Text Entry (*p* < .01)		
Medial Frontal Cortex	77	-3, 12, 57
Left Intraparietal Sulcus	35	-33, -48, 36

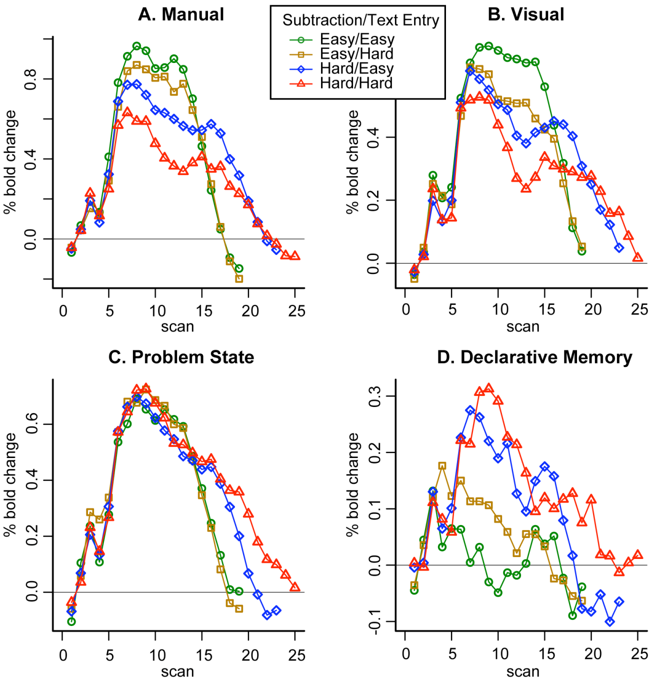


Figure 4. fMRI results for the four regions.

Posteriori Model Fit

One of the predictions of our model was an interaction effect in the posterior parietal cortex. However, instead of clear differences, the data show quite similar curves. While the area under the curves does give an indication of more total activation in the more difficult conditions, the data look very dissimilar from our model predictions.

From previous ACT-R/fMRI research it is known that activation in the problem state region often reflects visual processing (e.g., Kao & Anderson, personal communication; Sohn, et al., 2005), which is consistent with the literature on the posterior parietal cortex (e.g., Culham & Kanwisher, 2001). Figure 5A shows activation in the left fusiform gyrus and the left posterior parietal cortex in the predefined regions of ACT-R during a simple stimulus-response task (Kao & Anderson, personal communication). In this task participants had to press a key in response to the appearance of a stimulus, without any further processing. As can be seen, activation was observed in the posterior parietal cortex. Because in this task no problem states are involved, the activation in the parietal cortex cannot have been caused by problem state activity. On this basis, we argue that activation in ACT-R's parietal region is not only due to problem state related actions, but also to visual-spatial actions. This notion was operationalized by assuming that ACT-R's visual-location module (which represents spatial information and was not mapped onto a brain region before) and the problem state module both cause activation in the posterior parietal cortex.

To let our model make new predictions for the problem state region, we first calculated the influence of the visual system on the posterior parietal cortex in the data of Kao and Anderson. Linear regression showed that activation in the parietal cortex caused by the visual system was best predicted by taking .57 times the BOLD response of the fusiform gyrus. Next, we let the model predict activation in the parietal cortex by adding .57 times the activation of the visual-location module to the activation of the problem state module. The result can be seen in Figure 5B, showing a close fit to the data.

Discussion

In the current study we set out to confirm previous modeling results (Borst, et al., 2009) with an fMRI study. We used an existing experiment and cognitive model of the problem state bottleneck to generate a priori fMRI predictions. These model predictions turned out to be reasonably good indicators of activation in the visual, manual, and declarative memory regions of the brain. It should be noted that we did neither fit the model to the behavioral data, nor fit the model to the fMRI data. Usually, fMRI predictions are fitted to a model by calculating the best fitting *a*, *s*, and *m* parameters, but we thought it more informative to show our a priori predictions using default values.

In the posterior parietal cortex, associated with the problem state module, we found a different pattern than predicted by the model. To account for the BOLD response in the posterior parietal cortex, we let activation in this region depend both on activity of the problem state module, as is customary, and on the visual-location module, which was not mapped to a brain area before. While it is in accordance with the literature to assume visual-spatial influences in the parietal cortex (e.g., Culham & Kanwisher, 2001), the notion that the visual-location module influences the parietal cortex is tentative, and will have to be confirmed by new studies. Thus, while the resulting model outcome resembles the fMRI data, more experiments will be necessary to confirm the existence of a problem state bottleneck in the brain.

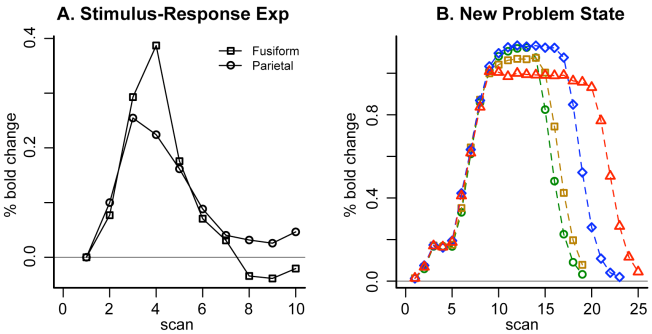


Figure 5. Results of a simple stimulus-response experiment and new problem state predictions.

Acknowledgments

This work was supported by Office of Naval Research grant N00014-08-10-541. Thanks to Katja Mehlhorn for comments on the manuscript.

References

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cog Sci, 29*, 313-341.

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.

Anderson, J. R., Fincham, J. M., Qin, Y., & Stocco, A. (2008). A central circuit of the mind. *Trends Cogn Sci, 12*(4), 136-143.

Borst, J. P., & Taatgen, N. A. (2007). The Costs of Multitasking in Threaded Cognition. *Proceedings of the 8th ICCM* (pp. 133-138). Ann Arbor, MI.

Borst, J. P., Taatgen, N. A., & Van Rijn, H. (2009). Modeling Triple-Tasking without Customized Cognitive Control. *Proceedings of the 31st Annual Conference of the Cognitive Science Society.*

Culham, J. C., & Kanwisher, N. G. (2001). Neuroimaging of cognitive functions in human parietal cortex. *Current Opinion in Neurobiology, 11*(2), 157-163.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological Review, 115*(1), 101-130.

Sohn, M. H., Goode, A., Stenger, V. A., Jung, K. J., Carter, C. S., & Anderson, J. R. (2005). An information-processing model of three cortical regions: evidence in episodic memory retrieval. *Neuroimage, 25*(1), 21-33.

Stocco, A., & Anderson, J. R. (2008). Endogenous control and task representation: an fMRI study in algebraic problem-solving. *J Cogn Neurosci, 20*(7), 1300-1314.

Comparing Human and Synthetic Group Behaviors: A Model Based on Social Psychology

Natalie Fridman, Gal A. Kaminka
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
{fridman, galk}@cs.biu.ac.il

Abstract

Existing models of group behavior, in a variety of fields, leave many open challenges. In particular, existing models often focus only on a specific phenomenon (e.g. flocking, pedestrian movement), and thus must be switched depending on the goals of the simulation. In contrast, we investigate a general cognitive model of simulating group behaviors, based on Festinger's Social Comparison Theory (SCT), a prominent social psychology theory. In previous work, we have show SCT covers a variety of pedestrian movement phenomena. In this paper we present evidence for SCT's generality by describing the use of the SCT model (using the Soar cognitive architecture) in generation of imitational behavior in loosely-coupled groups. Since the imitational behavior does not have clear standards of evaluation, we propose a method for such evaluation. Based on experiments with human subjects, we show that SCT generates behavior more in-tune with human crowd behavior.

Introduction

Models of crowd behavior facilitate analysis and prediction of the behavior of groups of people, who are in close geographical or logical states, and are affected by each other's presence and actions. Existing models of crowd behavior are often simplistic, and typically not tied to specific cognitive science theories or data. Moreover, existing computer science models often focus only on a specific phenomenon (e.g. flocking, pedestrian movement), and thus must be switched depending on the goals of the simulation.

We propose a novel model of crowd behavior, based on Social Comparison Theory (*SCT*) (Festinger, 1954), a popular social psychology theory that has been continuously evolving since the 1950s. The key idea in this theory is that humans, lacking objective means to evaluate their state, compare themselves to others that are similar. Similarity in SCT is very loosely defined—indeed, much of the literature on SCT addresses the exploration of different ways in which humans judge similarity.

In this paper we describe the implementation and adaptation of SCT the model in the Soar cognitive architecture. SCT was implemented as a secondary parallel thread within Soar. Whereas normally, operators are proposed (and selected) by Soar based on their suitability for a current goal, in our agent, operators were also proposed based on their suitability for SCT. We also briefly discuss mechanisms in the architecture, necessary for enabling SCT: a memory mechanism and an exploration mechanism.

We evaluate the use of SCT in generation of imitational behavior and show that SCT generates behavior in-tune with human crowd behavior. As the imitational behavior does not have clear standards of evaluation, we propose a method for evaluation of imitational behavior. The SCT model was evaluated in studies with human subjects. The subjects ranked SCT to be a middle-ground between completely individual behavior, and perfect synchronized ("soldier-like") behavior. Independently, human subjects gave similar rankings to short clips showing human crowds.

Background and Motivation

Social psychology literature provides several views on the emergence of crowds and the mechanisms underlying its behaviors. These views can inspire computational models, but are unfortunately too abstract to be used algorithmically. In contrast, computational crowd models tend to be simplistic, and focus on specific crowd behaviors (e.g, flocking). A common theme in all of them is the generation of behavior from the aggregation of many local rules of interaction, e.g. (Reynolds, 1987; Yamashita & Umemura, 2003).

Social psychology. A phenomenon observed within crowds, and discovered early in crowd behavior research, is that people in the crowd act similar to one another, often acting in a coordinated fashion which is achieved with little or no verbal communications.

There are several psychological theories that explained this coordinated behavior. For example, Le Bon (Le Bon, 1895) emphasized a view of crowd behaviors as "Collective Mind" that transform an individual who becomes a part of the crowd into becoming identical with the others in the crowd. Le Bon explains the homogeneous behavior of the crowd by two processes: *Imitation* and *Contagion*. Allport, (Allport, 1924) states that crowd behavior is a product of the behavior of like-minded individuals. According to Allport's theory, individuals become a part of the crowd behavior when they have a "common stimulus" with people inside the crowd. Additional explanation of coordinated crowd behaviors (Tajfel & Turner, 1986; Reicher, 2001) suggest that this coordination emerges because people in the crowd share a common social identity. Unlike Allport's individualistic behavior of people in crowds, Social Identity theory combines together the society aspects with an individual aspects.

Computational models. Work on modeling crowd behavior has been carried out in other branches of science, in particular for modeling and simulation. Reynolds (Reynolds, 1987) simulated bird flocking using simple, individual-local rules, which interacted to create coherent collective movement. There are only three rules: avoid collision with neighbors, match velocity with neighbors and stay close to the center of gravity of all neighbors.

Blue and Adler (Blue & Adler, 2000) used Cellular Automata (CA) in order to simulate collective behaviors, in particular pedestrian movement. The focus is again on local interactions: each simulated pedestrian is controlled by an automaton, which decides on its next action or behavior, based on its local neighborhoods.

Helbing et al. (Helbing & Molnar, 1997; Helbing, Molnar, Farkas, & Bolay, 2001) also focused on simulating pedestrian movement. Each entity moves according to forces of attraction and repulsion. Pedestrians react both to obstacles and to other pedestrians.

Yamashita and Umemura (Yamashita & Umemura, 2003) take a different approach in simulating group panic behavior. While inspired by Reynolds' model, they propose a model where each simulated person moves using three instincts: An escape instinct, a group instinct and an imitational instinct. According to Yamashita and Umemura, when a person is in panic, she acts based on these instincts, simplifying the decision making process.

Our work differs from those described above in that we aim to develop a general cognitive model of simulating group behaviors, one based on psychology. We have already shown that our model covers pedestrian movement phenomena as was presented in our previous work (Fridman & Kaminka, 2007), together with initial results on imitational behavior. Here, we present additional evidence for such generality by describing implementation in Soar, and evaluation of SCT model on imitational behavior in loosely-coupled groups. We discuss the full set of results, and the evaluation methodology, in detail.

A Model of Social Comparison

Our research question deals with the development of a computerized cognitive model which, when executed individually by many agents, will cause them to behave as humans do in groups and crowds.

We took Festinger's social comparison theory (Festinger, 1954) as inspiration for the social skills necessary for our agent. According to social comparison theory, people tend to compare their behavior with others that are most like them. To be more specific, when lacking objective means for appraisal of their opinions and capabilities, people compare their opinions and capabilities to those of others that are similar to them. They then attempt to correct any differences found.

We believe that social comparison theory may account for some characteristics of crowd behavior:

Imitation. Using social comparison, people may adopt others' behaviors. Festinger notes (Festinger, 1954): "The drive for self evaluation is a force acting on persons to belong to groups, to associate with others. People, then, tend to move into groups which, in their judgment, hold opinions which agree with their own“.

Contagion. One implication of SCT is the formation of homogeneous groups. Festinger writes (Festinger, 1954): "The existence of a discrepancy in a group with respect to opinions or abilities will lead to action on the part of members of that group to reduce the discrepancy".

To be usable by computerized models, social comparison theory must be transformed into a set of algorithms that, when executed by an agent, will proscribe social comparison behavior. A first step towards this goal has been take by Newell, who examined the axioms of social comparison (Newell, 1990), a subset of which appears here:

1. When lacking objective means for evaluation, agents compare their state features to those of others.
2. Agents compare themselves to those who are more similar; comparison increases with similarity.
3. Agents take steps to reduce differences to the objects of comparison.

Newell argued that these axioms are not social, in the sense of requiring active interaction between the agents. Rather, they utilize uni-directional observations and actions by the comparing agents.

We turn these abstract axioms into a concrete algorithm. The algorithm is described in (Fridman & Kaminka, 2007), and we provide only a brief description here. Each observed agent is assumed to be modeled by a set of features and their associated values. For each such agent, we calculate a similarity value $s(x)$, which measures the similarity between the observed agent and the agent carrying out the comparison process. The agent with the highest such value is selected. If its similarity is between given maximum and minimum values, then this triggers actions by the comparing agent to reduce the discrepancy:

```
1. For each known agent $x$ calculate similarity $s(x)$
2. $c \leftarrow \texttt{argmax}\, s(x)$, such that $S_{min} < s(c) < S_{max}$
3. $D \leftarrow$ differences between me and agent $c$
4. Apply actions to minimize differences in $D$.
```

SCT Implementation in Soar

We implemented SCT in the Soar cognitive architecture (Newell, 1990). Soar was connected to the GameBots virtual environment (Kaminka et al., 2002). Here, multiple agents, each controlled by a separate Soar process (each executing SCT) can interact with each other in a dynamic, complex, 3D virtual world (see Figure 1).

A detailed discussion of Soar's role as a cognitive architecture is beyond the scope of this paper. We provide a very brief overview here, and refer the interested reader to (Newell, 1990) for additional details.

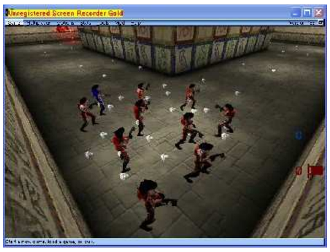


Figure 1: **Soar agents in the GameBots environment. Each agent has limited field of view and range, and may move about and turn.**

Soar has two components: A graph-structured working memory, and a set of user-defined production rules that test and modify this memory. Efficient algorithms maintain the working memory by executing rules that match existing contents. All the agent's knowledge, sensor readings, and decisions are recorded in the working memory. Soar operates in a classic sense-think-act cycle, which includes a decision phase in which all relevant knowledge is brought to bear to propose, and then select, an *operator*, that will then carry out deliberate mental (and sometimes physical) actions. Once the operator finishes its actions, it is automatically de-selected (terminated), and the cycle repeats. Unlike simple production rules, whose effects on working memory are temporary, operator-induced the actions of rule firings on working memory (and in turn, on physical actions) are persistent, even after the operator has been de-selected. Overall, a Soar agent's behavior is the result of the sequential selection of operators, each performing an action on the environment and/or internal memory.

For our experiments, several basic task-oriented operators were implemented, to allow the agents to move about, turn towards each other, measure distances to others, etc. Thus one thread of control, always running, is in control of the agent's actions towards whatever tasks it was given.

SCT was implemented as a secondary parallel thread within Soar (Figure 2). Whereas normally, operators are proposed (and selected) by Soar based on their suitability for a current goal (e.g., through means-end analysis), in our agent operators were also proposed based on their suitability for SCT. In other words, at every cycle, a Soar agent would consider operators that advance it towards its goal. In our implementation, it would also consider operators that seek to minimize perceived differences to other agents.

Thus SCT-proposed operators compete with the task-oriented operators for control of the agent. This may appear to contradict Festinger's theorizing that social comparison comes into play only when people are at an impasse. However, this is not the case. By setting Soar's decision preferences to prefer SCT-proposed operators only when no task-oriented operators are available, one gets the behavior predicted by Festinger's theory. Further exploration of this issue is beyond the scope of this paper.

The SCT thread proposed operators by following the algorithm described previously, though in a way that is adopted for Soar's decision cycle: At every cycle, for each observed agent and for each difference, the SCT process would propose an operator that would minimize the difference. Then, a set of preference rules is triggered that ranks the proposals based on feature weight. Additional rules prefer the most similar agent (that is still not sufficiently similar). Thus at the end, only one SCT operator is supported.

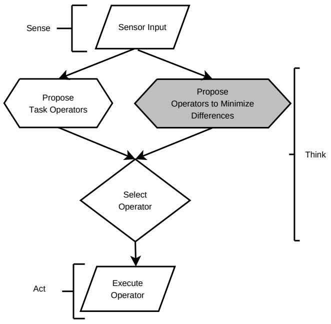


Figure 2: **The Soar sense-think-act decision cycle, SCT process highlighted.**

Here additional cognitive components became necessary. Suppose an agent X decided to turn towards the same angle as an agent Y that is next to it. Due to the limited field-of-view of X, it would lose track of Y once it makes the turn. From that point on, it could no longer keep track of Y, to minimize additional differences. This would cause it to become overly reactive, turning about immediately to seek Y again, or to select a different operator altogether (now that Y could no longer be imitated).

We thus found it necessary to utilize two mechanisms: (i) a memory mechanism that keeps track of the whereabouts of agents, once seen; and (ii) an exploration mechanism that occasionally would turn towards remembered agents, to provide an update on their state (for the purpose of comparison). Both of these mechanisms (memory and exploration) are of course present in many cognitive architectures, and are not necessarily linked to SCT. We thus leave discussion of such mechanisms outside of this paper.

Modeling Imitational Behavior

An attractive feature of social comparison is its hypothesized prevalence in human group behavior, i.e., its generality across different behaviors. Indeed, we believe that the SCT model we present in this paper is sufficiently general to account for a wide variety of group behaviors. This is in contrast to many existing computational models, that typically focus on specific tasks.

In previous work (Fridman & Kaminka, 2007) we evaluated the use of the SCT model in generation of pedestrian movement phenomena like bidirectional movement and movement in groups with and without obstacles. The SCT model accounts for group formation in pedestrians that are inter-related, a phenomenon not addressed by previous models. And where previous techniques apply, SCT shows improved results.

Here, we discuss in detail the implementation in Soar, and the evaluation methodology, providing additional evidence for such generality by describing the application of the SCT model to the problem of generating imitational behaviors in loosely-coupled groups. Unlike individual imitation, where one agent imitates a role model, crowd imitational behavior spreads across a group of individuals who dynamically select role models for imitation, from the level of observable actions to the level of unobservable internal mental attitudes (e.g., goals). Here, imitation occurs more loosely, as the role models do not necessarily intend to play their role, and indeed may not even know that they are being imitated. Also, the imitators potentially switch their role-model targets from one moment to the next. Psychology literature describes such imitational behavior as one of the keystones of crowd behaviors (Le Bon, 1895).

In order to simulate imitational behavior we used position and direction as the agents' feature set. For each observed agent and for every difference found, the SCT process proposes a corrective operator to be performed in order to minimize the difference in the selected feature. In this task, the corrective operators were 'move-to' (minimizing distance to the observed agent, correcting position differences) and 'turn-to' (imitating angle of the observed agent).

In addition to the proposed SCT operators, Soar also proposes operators based on their suitability for the current goal, and based on an exploration mechanism which proposes operators seeking new information. In this task, goal operators were 'turn-to' (a random angle); the exploration mechanism operators turned towards previously seen agents.

We used Soar preference rules to rank the feature weights such that the position feature gets higher priority than direction. This means that a closest agent is considered to be more similar, however the chosen feature for correction is direction. The S_{max} value was unbounded, which means that there is no such thing as too similar. In our case Soar can propose corrective operator with value equal to zero if there is no correction to make with respect to the observed agent. We used additional Soar preference rules to give higher priority to exploration mechanism operators than to goal operators. Thus, each agent prefers the SCT operators ('turn to') and in the case when there are no seen agents (i.e. there is no proposed SCT turn-to operator) an agent will prefer the exploration mechanism operators, and only afterwards the goal operators. The resulting simulated behavior has the agents standing in their initial locations, turning to some direction or doing nothing.

Evaluation of imitational behavior

We conducted experiments to evaluate whether SCT can indeed generalize to account for imitational behavior in groups. Unlike the pedestrian movement domain, where clear measures are available for objective measurement of the success of a model (e.g., flow, lane changes), imitational behavior does not have clear standards of evaluation.

We propose a method for evaluation of imitational behavior. We propose a questionnaire composed of general questions and specific tasks related questions. The general questions can be used as a common method for evaluation of all kinds of imitational behaviors. We rely on experiments with human subjects, which judged the human crowd behavior and the resulting SCT behavior in comparison to completely individual behavior (i.e., arbitrary decisions by each agent, independent of its peers), and to completely synchronized behavior (i.e., all agents act in complete unison).

The first hypothesis underlying the experiments was that groups controlled by SCT would generate behavior that would be ranked somewhere in-between the individual and perfect-coordination models, i.e., that SCT would generate behavior that would be perceived as coordinated, but not perfectly so. Another hypothesis is that human crowd behavior would also be ranked somewhere in-between the individual and perfect-coordinated behaviors.

To examine the first hypothesis, we created three screen-capture movies of 11 Soar agents in action. All movies were shot from the same point of view, and showed the agents in the same environment. In all screen-capture movies there is one blue agent that stands in front and turns up to 90° left or right. All others are red agents that act according to one of the models.

In one movie (*individual*), the red agents act completely independently of each other, randomly choosing an angle and turning to it. In another (*unison*), the red agents act in almost perfect coordination, turning towards the same angle as the blue agent almost instantaneously (small timing differences result from asynchronous responses of the simulated environment). Finally, in the *SCT* movie, the red agents act according to our model as described above.

These experiments were carried out using 12 subjects (ages: 18–40, mean: 28; male: 6; additional 4 subjects dropped due to technical reasons). Each subject was given a brief description of the appearance of the environment and agents, sometimes aided by a snapshot from a movie (e.g., as in Figure 1). The subjects were told that the purpose of the experiment was to evaluate the use of perception models embedded in the agents; that there was a red dot—visible to the agents but not to the subjects—that moves about on the walls surrounding the group. The agents' goal is to individually locate this dot, and then track it in place by turning around. The purpose of the cover story was to focus the attention of the subjects away from group behavior and imitation, so as to not bias the results. After the description, the movies were shown to the subject.

After each movie, the subjects were asked to fill a short questionnaire (described below) based on what they saw. Each movie was shown only once. The order of presentation of movies was randomly selected for each subject, to control for learning and order effects. The questionnaire included the following questions:

1. If there is only one red dot in the room, to what degree did

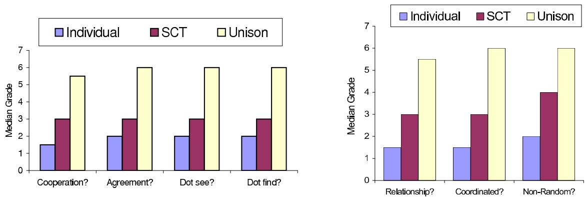
all agents see it? (1 - nobody saw the red dot; 6 - all agents saw it)

2. To what degree were the movements of the agents random? (1 - not random at all; 6 - very random)
3. To what degree was there cooperation between the agents? (1 - no cooperation at all; 6 - full cooperation)
4. To what degree was there agreement between the agents? (1 - no agreement at all; 6 - full agreement)
5. To what degree were the agents coordinated in terms of the direction of their movements? (1 -no coordination at all; 6 - fully coordinated)
6. How quickly did the agents find the red dot? (1 - dot not found at all; 6 - immediately found)
7. To what degree were the agents related to each other? (1 - no relation at all; 6 - tight relation)
8. Do you see any leaders? If so, how many? (1-11) (1- one leader; 11 - all agents are leaders, i.e., no leader).

In this experiment, the subjects were asked to grade the movies on an ordinal scale of 1–6, with 1 being a low score (typically associated with more individual behavior), and 6 being a high score (typically associated with perfect unison). In order to keep consistency in presentation of results, the scale of the second question (Non-Random) was reversed. The results of the last question (Number of leaders) are presented separately due to inconsistency in scale with other questions.

Agents results

In general, the responses to the questions in this experiment have placed SCT between the individual and unison models. Results are summarized in Figure 3(a) and 3(b). The questions in Figure 3(a) are associated with agents' performance on a given task. In the presented questionnaire the number of questions are 1, 3, 4 and 6. Figure 3(b) refers to more general questions (i.e. the same questions that were used in human crowd movie). In questionnaire the relevant numbers of questions are 2, 5, and 7. The categories in the X-axis correspond to questions given to the subjects. The Y-axis measures the median result. Each bar correspond to compared model and as explained above we compare SCT model to Individual and Unison models.



(a) (b)

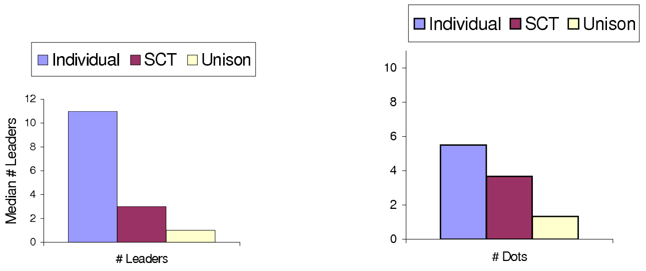
Figure 3: Results of questionnaire on agents performance.

The results clearly demonstrate that the SCT model lies in between the individual and perfect-unison model. While in some questions it appears to be somewhat closer to the individual model, it is significantly different from it at the $\alpha = 0.05$ significance level (t-test, one-tailed).

Figure 4(a) shows the results for the question on the number of leaders. The median result for the individual was 11 (i.e., every agent is a leader, or in other words, no leader). For the unison model, the median result was 1. For the SCT model, the median result was 3. In this question the SCT model result is very close to the Unison model. According to t-test (one-tailed) the SCT model significantly differs from the Individual model ($p = 0.02$). However, in comparison to Unison model there is no significance found ($p = 0.3$).

We conducted an additional experiment, in which static images—snapshots from the movies—were shown to subjects who were then asked how many red dots were present, based on the number of different directions in which agents were watching. The results of this experiment are summarized in Figure 4(b). Again the categories in the X-axis correspond to question given to the subjects. The Y-axis measures the average of median results that belong to each model.

Again the results demonstrate that the SCT model lies in between the individual and perfect-unison model and it significantly differs from the individual model ($p = 0.011$, t-test, one-tailed) and from perfect-unison model ($p = 0.012$, t-test, one-tailed).



(a) Number of leaders in screen-capture movies.

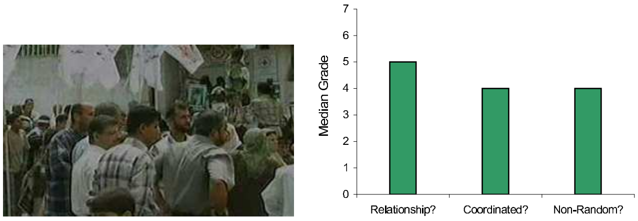
(b) Screen snapshot results.

Figure 4: Additional results for the simulated agents.

Human crowd experiment

Another hypothesis underlying the experiments is that human crowd behavior would also be ranked somewhere in-between the individual and unison models. To examine this, we search for a human crowd movie where individuals perform the same action as in simulated agents movies. We used a news clip movie which shows people, grouped together, standing and waiting for some event to occur. The only action they perform in the movie is to turn occasionally.

This experiment was carried out using 12 subjects different than in the screen-capture movies experiments. Each subject, after viewing a human crowd movie (Figure 5(a)) was asked to fill the same questionnaire as in previous experiments. However, since in the human crowd movie there was no cover story about red dot, there were some irrelevant questions that were dropped out. The remaining questions are more general and not tied to a specific task.



(a) A human crowd. (b) Results: News clip.

Figure 5: **Human crowd.**

Results are summarized in Figure 5(b). As in previous results, the categories in the X-axis correspond to questions given to the subjects and the Y-axis measures the median result.

We compare the human crowd results to the individual and perfect-unison models results. It appears to be significantly different from the individual model in all questions ($p = 0.000016$, $p = 0.000033$, and $p = 0.04$, respectively; t-test, one-tailed). However, in comparison to the perfect-unison model, the results of the coordination and non-random questions are significantly different($p = 0.0034$, and $p = 0.0003$, respectively; t-test, one-tailed). The results of the relationship question shows no significant different between the perfect-unison and the news-clip movie ($p = 0.44$).

In response to the question “Do you see any leaders? If so, how many?”, the median result in human crowd movie was 1.5. It appears to be significantly different from the individual model ($p = 0.001$, t-test, one-tailed) but not in comparison to the perfect-unison ($p = 0.374$). When the subjects were asked to qualitatively discuss their answer to this question, many subjects reported that they don’t see any leader, however “one must be present outside of the view of the movie, since the crowd is waiting for something or someone”. However, when they were asked to refer to only people seen in the movie, the answer was that there were several subgroups in the seen crowd. While this qualitative answer is similar to the answer we received in asking similar questions about the simulation movies, we do not believe that this necessarily suggests that the SCT model is completely accounting for realistic behavior. In the future, we will focus more explicitly on the issue of subgroups, by adding the following question to the questionnaire: “Are there any subgroups? If so, how many?”.

Summary and Future Work

This paper presented a model describing crowd behavior, inspired by Festinger’s social comparison theory (Festinger, 1954). The model intuitively matches many of the characteristic observations made of human crowd behavior. We presented an implementation of SCT model in Soar cognitive architecture, for experiments in imitational behavior. Though there is a lack of objective data against which the model can be evaluated, results of experiments with human test subjects are promising and seem to match intuitions as to observed behavior. The subjects ranked SCT to be a middle-ground between completely individual behavior, and perfect synchronized (“soldier-like”) behavior. Independently, human subjects gave similar rankings to a short news clip showing human crowds.

Acknowledgements. This research was supported in part by ISF grant #1357/07, and by IMOD.

References

Allport, F. H. (1924). *Social psychology*. Boston: Houghton Mifflin.

Blue, V. J., & Adler, J. L. (2000). Cellular automata microsimulation of bidirectional pedestrian flows. *Transportation Research Record*, 135–141.

Festinger, L. (1954). A theory of social comparison processes. *Human Relations*, 117–140.

Fridman, N., & Kaminka, G. A. (2007). Towards a cognitive model of crowd behavior based on social comparison theory. In *AAAI*.

Helbing, D., & Molnar, P. (1997). Self-organization phenomena in pedestrian crowds. In F. Schweitzer (Ed.), *Self-organization of complex structures: From individual to collective dynamics* (pp. 569–577). London: Gordon and Breach.

Helbing, D., Molnar, P., Farkas, I. J., & Bolay, K. (2001). Self-organizing pedestrian movement. *Environment and Planning B*, *28*, 361–384.

Kaminka, G. A., Veloso, M. M., Schaffer, S., Sollitto, C., Adobbati, R., Marshall, A. N., et al. (2002, January). GameBots: A flexible test bed for multiagent team research. *Communications of the ACM*, *45*(1), 43–45.

Le Bon, G. (1895). *The crowd: A study of the popular mind.* (1968 ed.). Dunwoody, Ga., N.S. Berg.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, Massachusetts: Harvard University Press.

Reicher, S. (2001). The psychology of crowd dynamics. In *Blackwell handbook of social psychology: Group processes.* Oxford: Blackwell.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on computer graphics and interactive techniques (SIGGRAPH-87)* (pp. 25–34). New York, NY, USA: ACM Press.

Tajfel, H., & Turner, J. C. (1986). The social identity theory in intergroup behavior. In *Psychology of intergroup relations.* Chicago: Nelson-Hall.

Yamashita, K., & Umemura, A. (2003). Lattice gas simulation of crowd behavior. In *Proceedings of the international symposium on micromechatronics and human science* (pp. 343–348).

Learning to Use Episodic Memory

Nicholas A. Gorski (ngorski@umich.edu)
John E. Laird (laird@umich.edu)
Computer Science & Engineering, University of Michigan
2260 Hayward St., Ann Arbor, MI 48109 USA

Abstract

This paper brings together work in modeling episodic memory and reinforcement learning. We demonstrate that is possible to learn to use episodic memory retrievals while simultaneously learning to act in an external environment. In a series of three experiments we investigate learning what to retrieve from episodic memory and when to retrieve it, learning how to use temporal episodic memory retrievals, and learning how to build cues that are the conjunctions of multiple features. Our empirical results demonstrate that it is computationally feasible to learn to use episodic memory in all three experiments, and furthermore, that learning to use internal episodic memory accomplishes tasks that reinforcement learning alone does not. These experiments also expose some important interactions that arise between reinforcement learning and episodic memory.

Keywords: Artificial Intelligence; Cognitive Architecture; Episodic Memory; Intelligent Agents; Reinforcement Learning.

Introduction

In this paper, we study possible mechanisms for learning to use the retrieval of knowledge from episodic memory. This unifies two important related areas of research in cognitive modeling. First, it extends prior work on the use of declarative memories in cognitive architecture where knowledge is accessed from declarative memories via deliberate and fixed cued retrievals (Wang & Laird, 2006; Anderson, 2007; Nuxoll & Laird, 2007) by exploring mechanisms for learning to use both simple and conjunctive cues. Second, it extends work on using reinforcement learning (RL) (Sutton & Barto, 1998) to learn not just control knowledge for external actions, but also to learn to control access to internal memories.

Earlier work has investigated increasing the space of problems applicable to RL algorithms by including internal memory mechanisms that can be deliberately controlled: Littman (1994) developed an RL agent that learned to toggle internal memory bits; Pearson et al. (2007) showed that an RL agent could learn to use a simple symbolic long-term memory; and Zilli & Hasselmo (2008) developed a system that learned to use both an internal short-term memory and an internal spatial episodic memory, which could store and retrieve symbols corresponding to locations in the environment. All three cases demonstrated a functional advantage from learning to use memory.

Our work significantly extends these previous studies in four ways: first, our representation is fully relational, which complicates both the structure of episodic memory and RL; second, our episodic memory system automatically captures all aspects of experience; third, our system learns not only when to access episodic memory, but also learns conjunctive cues and when to use them; and fourth, it takes advantage of the temporal structure of episodic memory by learning to advance through episodic memory when it is useful (this property is also shared by the Zilli & Hasselmo system, but for simpler task and episodic memory representations).

Our studies are pursued within a specific cognitive architecture, namely Soar (Laird, 2008), which incorporates all of the required components: perceptual and motor systems for interacting with external environments, an internal short-term memory, a long-term episodic memory, an RL mechanism, and a decision procedure that selects both internal and external actions. In comparison, ACT-R (Anderson, 2007) has many similar components but does not have an episodic memory. Its long-term declarative memory stores only individual chunks, and it does not store episodes that include the complete current state of the system. To do so would require storing the contents of all ACT-R's buffers as a unitary structure, as well as the ability to retrieve and access them, without having the retrieved values being confused with the current values of those buffers. Moreover, ACT-R's declarative memory does not inherently encode the temporal structure of episodic memory, where temporally consecutive memories can be recalled (Tulving, 1983). While the work presented in this paper is specific to learning to use an episodic memory, similar work could be pursued in the context of ACT-R by learning to use its declarative memory mechanism. However, we are unaware of existing work in that area, and even if there were, it would fail to engage the same issues that arise with episodic memory.

Background

Soar includes an episodic memory that maintains a complete history of experience (Nuxoll & Laird, 2007), implemented so as to support efficient memory storage and retrieval (Derbinsky & Laird, 2009). "Snapshots" of Soar's working memory, which is a relational graph structure, are automatically stored in episodic memory so that learning is not required to control how and when information is stored.

To retrieve an episode, a *cue* is created in working memory by Soar's procedural knowledge, which is encoded as rules. A cue is a relational structure that describes a subset of working memory elements that may exist in an episode. The cue is compared to the stored episodes, and the episode that best matches the cue is retrieved to working memory. If there are multiple episodes with the same degree of match, the most recent of those episodes is retrieved. Once an episode is retrieved to working memory, other knowledge (such as procedural knowledge) can access it.

After performing a cue-based retrieval, the agent can use the unique temporal structure of episodic memory and retrieve the next episode, providing a mechanism for the agent to move forward through its memories, recalling sequences of experiences, in addition to specific instances.

Although it is straightforward to create agents that use episodic memory for a variety of purposes (Nuxoll, 2007), this requires endowing the agent with knowledge as to when to access episodic memory and what structures should be used for cueing retrievals. In this research, we study the possibility of learning when to use episodic memory as well as learning which cues to use from experience using Soar's RL mechanism. Soar uses a type of RL called Q-Learning (Nason & Laird, 2005). Q-Learning learns the value for potential actions using temporal-difference updates of reward (Sutton & Barto, 1998) and in Soar this can be used to learn to control external actions as well as internal actions that retrieve information from episodic memory.

Well World

In order to explore how an agent might learn to use an internal episodic memory, we constructed several tasks within a domain we call "Well World." The domain is simple enough to be tractable for an RL agent, but rich enough such that episodic memory can potentially improve performance. The goal in Well World is to be safe when not thirsty, and to quench thirst as soon as possible when thirsty.

In Well World, the agent moves between objects and can consume resources, such as *water* or *shelter* if they are present. The agent perceives the object that is present at its current location, features of the object (including resources that are present), and adjacent objects that it can move to.

Figure 1 shows the base Well World environment. There are two wells which can provide the water resource ("r: water" in the Figure). Well 1 is currently empty, while well 2 has water available. There is also a shelter, which allows the agent to feel safe when the agent is not thirsty.

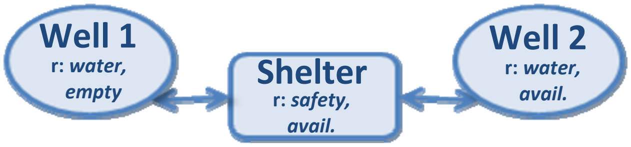


Figure 1: Objects, resources, and adjacency in Well World.

An agent in Well World possesses two internal drives: thirst and safety. When its thirst is quenched, an agent's thirst drive is 0; on every time step after it has been quenched, the thirst drive is incremented by a small amount. After passing a threshold, the agent is thirsty until it quenches its thirst, which requires that the agent move to a well object that contains water and consume water from it.

Only one well contains water at any given time; once water is consumed from a well, it is empty and water becomes available in the other well. In Figure 1, well 2 has water available while well 1 does not. Once the water at well 2 is consumed, well 2 will be empty while well 1 will have water available, and so on.

The agent's other internal drive is to feel safe. The agent satisfies this drive when not thirsty or when it consumes the safety resource from the shelter (which is always available).

Two of Well World's characteristics make it challenging for RL: the agent can only perceive the status of the object in its current location, and wells alternate in containing water and being empty. To perform optimally, an agent must maintain a memory of the environment (the status of the wells) – something a conventional RL agent lacks.

Reinforcement in Well World

The reward signal used by an RL agent in Well World is determined by the state of the agent's internal drives, as well as changes in the states of those drives. Reinforcement in Well World is internally calculated by the agent based on its internal drives, rather than determined by the environment as in a conventional RL setting.

The most important aspects of the agent's reward structure are that: there is a cost for taking external actions and it is greater than the cost of internal actions; there is a reward for not staying at the wells when the agent is not thirsty; there is a significant reward for performing the action (consuming water when thirsty) that is made possible by the episodic retrieval; and there is no explicit reward for using episodic memory, rather such control knowledge must be learned while seeking to satisfy thirst. The reward values are as follows. External actions result in -1 reward, while internal actions result in -0.1 reward. On every time step that the agent is thirsty, it receives -2. On every time step that the agent is not thirsty and consumes the safety resource, it receives +2. Finally, the agent receives +8 for satisfying its thirst. Concurrent rewards (e.g. the agent is thirsty and takes an external action) are summed together.

Experiments in Well World

Within the Well World domain, we developed a suite of three experiments to evaluate various strategies for using episodic memory. In the first experiment, we tested an agent's ability to learn to select a single cue for episodic memory retrieval. The second experiment tested an agent's ability to learn to use the temporal aspects of episodic memory retrievals. The third experiment investigated the agent's ability to create a conjunctive cue (i.e. a cue that contains more than one feature). This set of experiments investigated all of the ways retrievals can be used to access Soar's episodic memory. Before discussing the experiments and results, we present the details of our agent.

Agent Design and Implementation

To explore learning to use episodic memory, we created a Soar agent. In our agent, procedural knowledge determines what actions can be taken in the external environment as well as what actions can be taken to access the internal episodic memory. On each time step of the environment, the procedural knowledge proposes applicable actions based on the agent's current perception of the environment and its internal state. It proposes consuming resources that are present, and it proposes moving to any adjacent objects. There are two internal actions that it can propose for controlling episodic memory (depending on the experiment, as described below): create a cue to initiate a retrieval, or if there has been a retrieval, advance episodic memory forward in time. In experiments where the agent must learn which retrieval cue to use, multiple retrieval actions are proposed, one for each cue.

The decision procedure selects actions probabilistically, based on what has been learned by Q-learning. A central problem in RL is the exploitation vs. exploration trade-off (Sutton & Barto, 1998) - an agent must balance between choosing actions based on what it has already learned (exploitation), with choosing other actions to gain more knowledge about the effects of those actions (exploration). Our agent uses a linearly decaying exploration rate; initially, the agent selects a random action half of the time and the other half selects actions according to their learned values. As time goes on, the agent takes random actions less often.

Although the Well World is presented in terms of "water," "thirst," "empty," and "wells," the agent does not know the semantics of these terms. To the agent, consuming water is simply a possible action that it can take; it must learn that it is good to consume water when thirsty, that water is available at a particular well, and so on.

In contrast to many learning systems that are "reset" after a performance or learning trial, our agent has a continual existence and once it begins acting in the environment, it continues to move about Well World, performing actions, until the end of the experiment.

Instead of using episodic memory, the agent could have maintained task-specific events in working memory (such as which well the agent last consumed). This memory would provide the agent with sufficient knowledge to learn to act in the domain. However, this approach requires task-specific background knowledge while our approach is completely general and applies to any task without additional task-specific knowledge.

Results presented in this paper are the average of 250 trials, were smoothed with 4253 Hanning, and normalized so that an average reward of 0 per action is optimal.

Learning to Retrieve Episodic Memories

The first experiment tests the basic behavior of using RL to learn to use an internal episodic memory, and its purpose is to determine whether an RL agent can learn what to retrieve and when retrieval is appropriate. In this experiment, an agent must learn to retrieve information from memory using a single cue, where the retrieved episode provides sufficient information to perform the task. In one condition, there is only one cue available to the agent to use for retrieval; in another, the agent selects from six possible cues, only one of which is useful.

In Well World (Fig. 1), the optimal *policy* (where a policy is a mapping of every state, or situation, to an action) is for the agent to move to the shelter and consume the safety resource when it is not thirsty, and to move to the well that contains water and consume water there when it is thirsty. As agents are unable to perceive which well contains water, an agent that does not possess an internal memory does not know which well it must move to and wastes time while trying to find available water. An agent that possesses an internal memory, however, can retrieve the episode for the last visited well.

Figure 2 shows the results under the following conditions: only the correct cue is available to be learned (labeled "No distracters"); the correct cue and five distracters are available to be learned ("5 distracters"); and a baseline condition in which episodic memory is lesioned and all retrievals fail ("Lesioned ep. mem.").

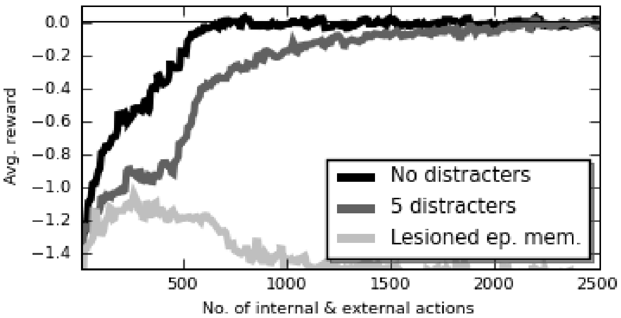


Figure 2: Performances of agents learning to retrieve episodic memories.

When only a single cue is available for retrieval, the agent quickly learns both to act in the environment and to use its internal memory so as to receive the maximum amount of possible reward (it follows the optimal policy). When distracter cues are present, the agent learns more slowly but also converges to the optimal policy. These results indicate that the agent can learn to use its internal memory while simultaneously interacting with its environment.

Learning to Retrieve What Happened Next

A unique aspect of episodic memory is that events are linked and ordered temporally. In Soar's episodic memory, memory retrievals can be controlled temporally by advancing to the next (or previous) memory after performing a cue-based retrieval, providing a primitive envisioning or planning capability where the agent can use its prior history to predict potential future situations. Through RL, the system has the potential of learning when and how to perform such primitive planning.

In the previous experiment, agents retrieved episodic memories of the last time that they had perceived the water resource, which was sufficient information to determine which well to move to in order to find water. An alternative strategy, explored in this experiment, is to retrieve a situation that closely resembles the agent's current situation and then advance to the next memory to remember what the agent did the last time that it was in a similar situation.

In this experiment, the agent has available the normal actions in the environment (moving and consuming resources). It also has two internal actions available to it: a cue-based episodic memory retrieval, which uses structures from its current perceptual state to retrieve the most recent situation that most closely resembled its current situation; and an action (called *advance*) that retrieves the next episode (the episode that was stored after the episode most recently retrieved). Thus, the agent must learn when to do a cue-based retrieval and when to advance its retrieval, and these actions are always competing with the other actions.

For this task, the optimal policy for the agent when it is not thirsty is to move to the shelter and consume the safety resource. When it becomes thirsty, the optimal policy is to perform a retrieval cued by its current state, which results in the agent remembering the last time it was thirsty at the shelter. The next step is to perform an advance retrieval, which results in the agent remembering where it moved to after it was last thirsty at the shelter. This is followed by moving to the other well, where the agent will find water, as the well that it previously visited will be empty.

An important characteristic of this task is that the agent must learn to use its memory while simultaneously learning to act in the world. The best policy for memory usage depends on the agent's prior actions in the environment; if the agent does not visit and consume resources in the appropriate order (i.e. follow the optimal policy for external actions), then the agent is not guaranteed to gain useful information from internal memory retrievals.

The performances of the agent under three conditions are plotted in Figure 3: using a fixed policy to automatically advance episodic memory after a cue-based retrieval, making only the initial cue-based retrieval open to learning; learning when to select both retrieval and advance actions; and a baseline comparison where episodic memory is lesioned.

There are several features of the results in Figure 3 worth further discussion. First, the performances of both agents that use episodic memory are very similar. This was unexpected. The agent that learns to use the temporal action has a larger action space, which implies that it would initially perform worse than the agent that had a fixed policy to advance to the next memory after retrieving. Second, the agents reach asymptotic performance after about 4,500 actions, but do not reach the optimal level of performance. Third, while the agents are exploring while selecting actions (until the 4,000th action), the agent that deliberately selects actions outperforms the agent that has a fixed policy to advance after retrieving. Fourth, there is a dramatic improvement in performance just after exploration ends. The agent retrieves episodes from memory that are similar to its current situation, and uses its past actions to determine how to act in the present situation. If the agent takes an exploratory action when it is thirsty or is not at the shelter when it becomes thirsty because of an exploratory action, then the behavior that results is no longer correct. In effect, although exploration of the problem space is necessary for the agent to learn, it hinders the agent's performance in the task and once there is no exploration the agent can perform significantly better.

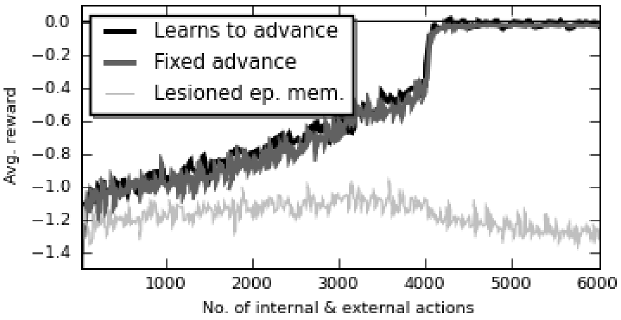


Figure 3: Performances of agents using temporal control of episodic memory after retrieval.

All four of these phenomena are explained by the difficulty of the learning problem that was identified above - for the agent to learn the optimal policy for using its internal memory, it must also learn a near optimal policy for acting in the environment. The learning problem is partially observable, in that the effects of the agent's memory actions depend on the history of the agent's actions in the environment, but the agent cannot perceive that history. The agent is faced with a conundrum: it must learn how to use its memory while settling on a good policy in the environment, but it must also settle on a good policy in the environment without knowing how to use its memory. Often the agent is successful in learning to simultaneously control both memory and external action, but occasionally the agent is unable to converge to the best policy.

The asymptotic behavior of the agent is very near to optimal, which demonstrates that the agent still learns to perform relatively well in the environment. In fact, in all trials the agent converged to one of two policies: the optimal policy, or a policy in which the agent uses episodic memory retrievals to toggle a conceptual bit, as in the agents in Littman (1994) and Pearson et al. (2007). In this second policy, when the agent becomes thirsty, it immediately moves to one of the wells (the same well every time). If the well contains water, it consumes it; if not, it performs a retrieval and moves back to the shelter. At the shelter, the agent now knows that it has performed a retrieval and instead of moving to the same well again (the one that it just visited and knows is empty), it moves to the other well and consumes water there. Essentially, the agent learns which well to move to when it is thirsty based on whether a retrieval has been performed, and not based on the contents of what was retrieved.

From Figure 3 it is also clear that the agent requires many more actions before converging to near-optimal behavior in comparison with the agents from the previous experiment. For the agent to converge to the optimal control policy, it must explore significantly longer than in the previous experiment; however, as noted above, this exploration can hinder the agent's performance in the task as well. We investigated how different exploration policies affected the agent's convergence to the optimal policy and the results are presented in Table 1. In all three cases, the rate of random action selection decays linearly over time. Table 1 presents data gathered when random action selection decayed over 500 steps, 5,000, and 50,000. These results suggest that there are important interactions between the exploration rate decay and learning that need to be pursued in future work.

Table 1: Percentage of trials that converged to optimal memory control policy when using temporal control for different periods of exploration.

Condition	500	5,000	50,000
Fixed	26%	60%	25%
Deliberate	36%	71%	38%

Learning To Construct a Retrieval Cue

In the first experiment, one condition involved the agent learning to select between multiple cues when retrieving from memory. In the second experiment, the agent used cues with more than one feature (multiple features of its current state) in order to retrieve from memory. The purpose of this third experiment is to investigate whether an agent can learn to select multiple features to use as cue, combining aspects of both previous experiments.

In order to test this capability, it was necessary to extend the base Well World configuration so that there were more wells and more features that could be used for retrieval. A third well was added to the environment, and a color feature was added to all objects; the modified environment is shown in Figure 4. As in the base environment, only wells 1 and 2 ever contain water, and they continue to alternate between full and empty as before. Well 3 is always empty and never contains water; it was added to the environment to serve as a distracter to the agent when it performs a cue-based retrieval with features not present on the other two wells.

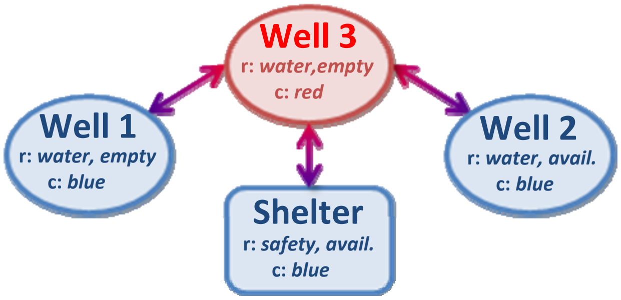


Figure 4: Well World modified with an additional well and an additional feature, color.

In this task, the optimal policy when the agent is not thirsty is still to navigate to the shelter and consume safety. When thirsty, the agent must construct a cue containing features corresponding to the two wells that can contain water in order to determine which well it visited last; these features are "resource: water" and "color: blue". After retrieving the memory of the last blue well that it visited, the agent must then navigate to the *other* blue well and consume water there to satisfy its thirst.

If the agent constructs a cue with some other combination of features, the result of its retrieval depends on its previous behavior – but the retrieved episode will not provide sufficient information for the agent to determine which well to visit next, because the agent must always visit the red well before visiting the shelter. As Soar's episodic memory mechanism is biased towards more recent episodes when multiple memories are perfect matches to the cue, building a cue that contains only "resource: water" or "color: blue" will not result in the agent remembering the last well that it visited (assuming that it has moved back to the shelter). Color: blue will lead to the retrieval of the shelter, while retrieval of resource: water will lead to retrieval of well 3.

The performances of the agent that constructs retrieval cues in the modified Well World are shown in Figure 5 for three conditions: learning to construct a cue from the two correct possibilities ("No distracters"), learning to construct a cue when two distracters are present, and a baseline where episodic memory is lesioned. In the two conditions, there are different sets of features with which an agent may construct the cue: the first has only the two correct features available (resource: water, and color: blue), while the other also has their complements (resource: water/shelter, and color: blue/red). Cues can contain any combination of features so the agent must learn to construct the cue from the correct combination in both cases.

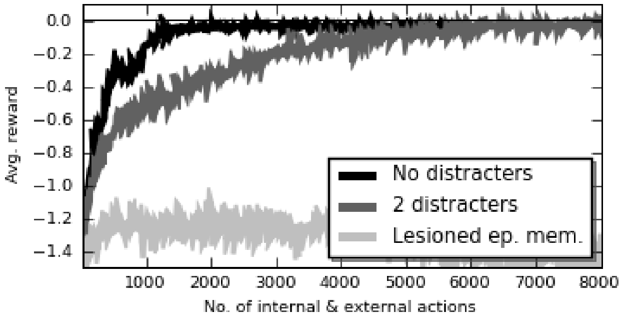


Figure 5: Constructing cues with more than one feature in order to retrieve from episodic memory.

The agent converges to the optimal policy under both conditions, more slowly when two distracter features are present, as expected. These results indicate that an agent can learn to build conjunctive cues from raw features, and use them in a task to retrieve from episodic memory.

Discussion and Conclusions

Although in all three experiments the agent is faced with learning to use its memory while acting in the environment (and thus affecting what information will be retrieved from memory in the future), the interaction of memory and action in the environment is significantly more intertwined in the second experiment. There, the agent's past actions directly impact the usefulness of information retrieved from episodic memory. In all experiments, the agent learns very early on to consume safety when it is not thirsty, and to immediately move to the shelter as soon as it is not thirsty. In the first and third experiments, this means that when the agent retrieves an episode from memory using features of a well as a cue, it will typically be the well that it last consumed water from. However, in the second experiment, the agent is retrieving memories of the first action that it took to quench its thirst, and *not* the memory of when it finally managed to quench it. It not only takes longer to learn how to best act in this setting, but the eventual result is that sometimes instead of converging to the optimal policy it instead converges to a local maximum in the policy space. One issue for future research that we identified in the second experiment is that our approach lacks task-independent strategies for controlling exploration.

In all experiments, the cost of an internal action is less than the cost of external action in the environment. The rationale behind this decision is that it takes significantly more time to act in the world than it does to perform an internal action. Although internal rewards are structured in this way, we have gathered results (not presented here in the interest of space) that demonstrate that this feature of our reward structure does not affect the eventual learned behaviors, but does serve to speed up the learning process by encouraging the selection of internal actions initially.

These three experiments demonstrate that RL can be applied successfully to learn to use internal actions over an episodic memory mechanism while simultaneously learning to act in its environment. Additionally, RL alone cannot be successfully applied to those same tasks, demonstrating that there is a functional advantage to combining RL with an episodic memory in some settings. We also demonstrated that RL can be used to learn when to retrieve, learn which cue to use for retrieval, learn when to use temporal control, and learn to build a cue from a set of possible features.

More broadly, this research opens up the possibility of extending the range of tasks and behaviors modeled by cognitive architectures. To date, scant attention has been paid to many of the more complex properties and richness of episodic memory, such as its temporal structure or the fact that it does not capture just isolated structures and buffers but instead captures working memory has a whole. Similarly, although RL has made significant contributions to cognitive modeling, it has been predominantly used for learning to control only external actions. This research demonstrates that cognitive architectures by incorporate both episodic memory and RL, they can learn behavior that is possible only when they are combined.

Although our research demonstrates that it is possible to learn to use episodic memory, it also raises some important issues. Learning is relatively fast when the possible cues lead to the retrieval of an episode that contains all of the information that an agent requires in order to determine how to act in the world. When retrieving episodes that most closely match the current state and then using temporal control of memory to remember what happened next, however, learning is slower and does not always converge to the best possible behavior. Learning to use episodic memory to project forward is difficult – requiring many trials to converge and without a guarantee that optimal behavior will be achieved. Do these same issues arise in humans or do they have other mechanisms that avoid these issues? One obvious approach to avoid the issues encountered in our experiment is to use one method, such as instruction or imitation, to initially direct behavior so that correct behavior is experienced and captured by episodic memory, and then learning to use those experiences would probably be much faster.

Acknowledgments

The authors acknowledge the funding support of the Office of Naval Research under grant number N00014-08-1-0099.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York, NY: Oxford U. Press.

Derbinsky, N. & Laird, J. E. (2009). Efficiently Implementing Episodic Memory. *Proc. of the 8th Intl. Conf. on Case-Based Reasoning.* Seattle, WA.

Laird, J. E. (2008). Extending the Soar Cognitive Architecture. *Proc. of the First Artificial General Intelligence Conference* (224-235). Memphis, TN.

Littman, M. L. (1994). Memoryless Policies: Theoretical Limitations and Practical Results. *Proc. of the 3rd Intl. Conf. on Simulation of Adaptive Behavior*. (238-245).

Nason, S. & Laird, J. E. (2005). Soar-RL, Integrating Reinforcement Learning with Soar. *Cog. Sys. R., 6,* 51-59.

Nuxoll, A. (2007). *Enhancing Intelligent Agents with Episodic Memory.* Doctoral dissertation, Computer Science & Engineering, U. of Michigan, Ann Arbor.

Nuxoll, A. & Laird, J. E. (2007). Extending Cognitive Architecture with Episodic Memory. *Proc. 21st National Conference on Artificial Intelligence.* Vancouver, BC.

Pearson, D., Gorski, N.A., Lewis, R.L. & Laird, J.E. (2007). Storm: A Framework for Biologically-Inspired Cognitive Architecture Research. *ICCM-07.* Ann Arbor, MI.

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning*. Cambridge, MA: MIT Press.

Tulving, E (1983). *Elements of Episodic Memory*. Oxford: Clarendon Press.

Wang, Y. & Laird, J. E. (2006). *Integrating Semantic Memory into a Cognitive Architecture*. (Tech. Rep. CCA-TR-2006-02). Ann Arbor, MI: Center for Cognitive Architectures, University of Michigan.

Zilli, E. A. & Hasselmo, M. E. (2008). Modeling the Role of Working Memory and Episodic Memory in Behavioral Tasks. *Hippocampus. 18,* 193-209.

Fluctuations in Alertness and Sustained Attention: Predicting Driver Performance

Glenn Gunzelmann (glenn.gunzelmann@us.af.mil)
Cognitive Models and Agents Branch, Air Force Research Laboratory
6030 South Kent St., Mesa, AZ 85212 USA

L. Richard Moore, Jr. (larry.moore@mesa.afmc.af.mil)
Lockheed Martin at Air Force Research Laboratory
6030 South Kent St., Mesa, AZ 85212 USA

Dario D. Salvucci (salvucci@cs.drexel.edu)
Department of Computer Science, Drexel University
3141 Chestnut St., Philadelphia, PA 19104 USA

Kevin A. Gluck (kevin.gluck@us.af.mil)
Cognitive Models and Agents Branch, Air Force Research Laboratory
6030 South Kent St., Mesa, AZ 85212 USA

Abstract

Fatigue has been implicated in an alarming number of motor vehicle accidents, costing billions of dollars and thousands of lives. Unfortunately, the ability to predict performance impairments in complex task domains like driving is limited by a gap in our understanding of the explanatory mechanisms. In this paper, we describe an attempt to generate *a priori* predictions of degradations in driver performance due to sleep deprivation. We accomplish this by integrating an existing account of the effect of sleep loss and circadian rhythms on sustained attention performance with a validated model of driver behavior. Although quantitative empirical data for validation are lacking, the predicted results across four days of sleep deprivation match qualitative trends published in the literature, and illustrate the potential for making useful predictions of performance in naturalistic task contexts that are relevant to real applied problems.

Keywords: Driver Behavior; Fatigue; Computational Model; Sustained Attention; Sleep Deprivation.

Introduction

Accidents on roadways in the United States account for a distressingly high number of fatalities and substantial cost on an annual basis (Horne & Reyner, 1999; Klauer, Dingus, Neale, Sudweeks, & Ramsey, 2006; NTSB, 1995; Pack et al., 1995). According to a National Highway Transportation Safety Administration report, nearly 25% of these accidents can be wholly or partially attributed to the effects of drowsiness or fatigue on driver attention, judgment, and/or performance (NTSB, 1995).

The alarmingly high cost of fatigue in the context of driving has been one motivation for studies to better understand changes in cognitive performance stemming from extended time awake (sleep deprivation), insufficient sleep (sleep restriction), and being awake at times of the day when the body is predisposed to sleep (circadian desynchrony; Dijk, Duffy, & Czeisler, 1992; Van Dongen & Dinges, 2005a; 2005b). This research has succeeded in identifying characteristic consequences of fatigue on cognitive performance. However, there remain significant limitations in the capacity to make valid predictions about performance in novel task contexts based on a history of time awake and circadian rhythms (Dinges, 2004; Van Dongen, 2004).

Our computational modeling research has been targeted at addressing some of these current limitations in predictive validity. Much of this research addresses significant theoretical challenges associated with understanding the link between cognitive processes and fluctuations in overall cognitive arousal, or alertness (e.g., Gunzelmann, Gross, Gluck, & Dinges, 2009; Gunzelmann, Gluck, Kershner, Van Dongen, & Dinges, 2007). However, we are also addressing the issue of how these theoretical insights can be used to make a priori quantitative performance predictions in novel, naturalistic task contexts, based upon the mechanisms and parameters that have been identified (e.g., Gunzelmann, Byrne, Gluck, & Moore, 2009; Gunzelmann & Gluck, in press)

In the research presented here, we evaluate the capacity to make predictions about degradations in driver performance associated with an extended period of total sleep deprivation. We discuss the implications of our research in the context of potential applications of a predictive capacity in the domain of driving. In the next sections, we describe our model of driving behavior, our theoretical mechanisms for fatigue, and how they are integrated to allow for the generation of quantitative predictions of behavior. We then compare the model's predictions with qualitative trends in the empirical literature, demonstrating that the a priori predicted trend in the integrated model are aligned with those published results.

Driver Model

The first component of our exploration of driving and fatigue is the ACT-R driver model (Salvucci, 2006), a computational model of driver performance developed in the ACT-R cognitive architecture (Anderson, 2007; Anderson et al., 2004), which serves as a psychological theory and simultaneously a computational framework for specifying and simulating human behavior models. The driver model is based on a control law of steering behavior (Salvucci & Gray, 2004) that visually encodes two salient points on the roadway: a near point in the lane center immediately in front of the vehicle; and a far point such as the vanishing point on a straight road, the tangent point on a curved road, or the lead vehicle when present. The control law describes how steering can be realized by keeping the far point stable while keeping the near point both stable and centered in the current lane.

The driver model that uses this control law relies on a fundamental component of the ACT-R architecture – the production system that represents central cognition. Central cognition in ACT-R operates through a series of conflict resolution cycles to produce cognitive processing and behavior. During each cycle the subset of productions whose conditions match the current system state is identified. The "system state" is represented by the contents of a set of buffers that provide limited-bandwidth communication between central cognition and peripheral information processing modules such as perception and motor action. Within this set of matching productions, the one with the highest "utility value" is selected and its actions are executed, provided that it exceeds the ACT-R "utility threshold" parameter. The default duration for these cycles is 50 ms.

The driver model uses successive iterations of four ACT-R production rules to represent the control law of steering behavior. Specifically, these four rules comprise a *control update cycle* during which the model (1) encodes the near point, (2) encodes the far point, (3) updates steering and acceleration according to the control law, and (4) checks the vehicle's current stability as measured by the lateral velocity and position of the near and far points. If the vehicle is not yet stable, the model immediately initiates another control update; otherwise, the model waits approximately 500 ms to initiate the next control update.

The driver model has been shown to account well for driver behavior with respect to curve negotiation and lane changing (Salvucci, 2006). The most critical aspect of the model for our purposes here is the execution time for a control update cycle: A single cycle requires approximately 200-250 ms, including 50 ms for each production rule firing (as dictated by ACT-R theory) plus some additional time for visual encoding. The update cycle time can increase, however, when attention is divided between driving and some secondary task, thus resulting in degradations in driver performance. For example, recent work has shown how dialing a phone (Salvucci, 2001; Salvucci & Taatgen, 2008) and rehearsing a memorized list of numbers (Salvucci & Beltowska, 2008) affects the driver model's performance; in both cases, concurrent execution of the secondary task interferes with processing of the driving task, thereby increasing the update cycle time and degrading performance (measured by, e.g., lateral deviation from lane center or brake response time to an external event). As we will describe, proposed mechanisms for fatigue in ACT-R can also prolong or delay the update cycle, leading to similar degradations in driver performance.

Mechanisms for Fatigue

The driver model provides a validated basis for making predictions about driver behavior. In independent research, efforts have been made to identify mechanisms within ACT-R to account for the impact of sleep loss and circadian rhythms on cognitive processing. In some of this research, we have focused on central cognitive mechanisms associated with the production execution cycle (Gunzelmann, Gross, et al., 2009). To account for changes associated with decreased alertness, we have integrated mechanisms in ACT-R that create opportunities for brief breakdowns in cognitive processing called *microlapses*.. In addition, we proposed a secondary process to represent the influence of explicit effort, which decreases the likelihood of a microlapse but also increases the probability of using lower-cost, less effective strategies in pursuit of achieving the goal.

The mechanisms in the fatigue model are based on the theoretical perspective that fluctuations in overall alertness or arousal can be associated with changes in utility values for selecting and executing production rules in ACT-R's central production system. Utility values are decreased, which increases the likelihood that no action will be taken on a given cycle. This situation leads to a microlapse, which is formally defined as a gap in cognitive processing lasting for the duration of one cognitive cycle (approximately 50 ms).

To account for the potential benefits of increased effort, a second parameter is manipulated – the utility threshold – which sets the minimum utility value required for a production to fire. Decreasing the utility threshold instantiates greater *effort* by making it more likely that some production will successfully fire. However, this manipulation also increases the probability that a suboptimal action (a production with a low utility) will be executed instead (see Gunzelmann, Gross, et al., 2009).

To evaluate the validity of our account, we compared the model's performance to human data on a sustained attention task across 88 hrs of total sleep deprivation. The model captured the important features of the human data, including explanations for small increases in the median of appropriately fast responses and increasing probabilities of false starts, slowed responses (lapses), and complete failures to respond (sleep attacks). The task, model, and results are described in detail in Gunzelmann, Gross, et al. (2009).

Integration

The mechanisms for fatigue instantiate a theory of changes in central cognitive processing resulting from fluctuations in alertness attributable to sleep loss and circadian rhythms.

Meanwhile, the model of driver behavior provides a validated account of mechanisms and processes involved in skilled driving. Importantly the ACT-R driver model relies on procedural knowledge for successful performance, including staying within its lane. As a result, an opportunity exists to bring together an existing model of driver behavior with an existing account of fatigue to explore the implications of fatigue on driving behavior. This opportunity represents an important step in the evolution of computational architectural accounts of cognitive phenomena, and illustrates the potential utility of unified theories that integrate theoretical insights from various domains of psychological research.

The integration of the driver model and fatigue mechanisms was a straightforward process. The implementation of the driver model was altered to run on a high-performance computer but was not changed with respect to its core behavior. The driver model is similar to the sustained attention model in that neither makes extensive use of declarative memory, simplifying the account by eliminating the need to consider potential influences of fatigue on declarative knowledge access (e.g., Gunzelmann et al., 2007). The fatigue mechanisms were taken directly from Gunzelmann, Gross, et al. (2009) and applied to the driver model. Thus, our procedural fatigue mechanisms alone provide the moderating effects in the driving model.

The actual effects of the fatigue mechanisms center on the production selection and execution phases of the production cycle in ACT-R. Proportional scaling of utility values during the selection phase of the driver model creates situations where the matching production with the highest utility fails to exceed the utility threshold. Thus, no production is executed on that cycle, producing a microlapse as described above. This is the key component in our theoretical account of performance declines associated with fatigue because it provides an account, based upon a single mechanism, of phenomena in the sleep research community that have been associated with *cognitive lapses* and *cognitive slowing* (e.g., Dinges & Kribbs, 1991). Parameter manipulations associated with fluctuations in alertness influence the frequency of microlapses, and microlapses lead to the performance changes exhibited by "tired" models.

In cases when a microlapse occurs with no other ongoing processes in any of ACT-R's information processing modules, the microlapse is accompanied by additional attenuation of utility values. The noise component of the utility values allows subsequent conflict resolutions to potentially match a production and continue model execution. However, this does not always occur, and as each successive decline in alertness further reduces the possibility of utilities rising about the threshold, a model can quickly spiral into a state analogous to sleep. In the model described in Gunzelmann, Gross, et al. (2009), this mechanism is critical in capturing the most substantial breakdowns in cognitive processing (i.e., sleep attacks).

In the sustained attention task, long periods of time go by – as long as 10 seconds – where the model is simply waiting for a stimulus event. In contrast, the processing in the driver model incorporates a constant monitoring behavior, which leads to cognitive processing in modules outside central cognition throughout the task. Peripheral processing does not affect the occurrence of microlapses, but does prevent any progressive declines in utility values over the course of a 10-minute driving session. The implication is that our model currently does not capture changes in performance that may be expected over the course of a 10-minute driving episode (i.e., time on task effects). However, our focus is on making truly *a priori* predictions, and so we leave them unchanged in the model runs described below.

In the next section, we evaluate the impact of our fatigue mechanisms on the driver model. Recall that the driver model realizes the continuous control law through four key productions. It is in this control update cycle that the fatigue mechanisms are most influential, since microlapses increase the overall update cycle time. As will be shown, even brief delays in cognitive activity can amount to significant and potentially devastating behavioral impacts.

Model Evaluation

To evaluate the model, its behavior was assessed in the context of a driving scenario described in Salvucci and Taatgen (2008). In the task, the driver steered down a single-lane highway, keeping the vehicle as centered as possible in the roadway. The vehicle moved at a constant speed that was not controlled by the driver, thus focusing the task particularly on lateral control. One key measure of performance in the task is *lateral deviation*: the root-mean-squared error between the lane center and the vehicle's lateral position within the lane. The baseline driver model navigating this environment exhibits an average lateral deviation of approximately 15 cm across a 10-minute driving scenario (see Salvucci & Taatgen, 2008).

To produce predictions of driver behavior and performance, we used parameter values for the fatigue mechanisms that were estimated in our research on sustained attention (e.g., see Gunzelmann, Gross, et al., 2009). Specifically, the model for that research was able to account for human sustained attention performance at 2 hour intervals across 88 hours of total sleep deprivation. As an initial assessment of the driver model, we used the parameter values from sessions occurring shortly after participants awakened on the baseline day of the study, and from sessions occurring after 24, 48, and 72 hrs of total sleep deprivation (0800 on each of 4 consecutive days). The model was run 200 times using each of those parameter sets, leading to reliable measures of central tendency in the performance measures as well as evidence regarding the variability in fatigue effects across 10 minute driving sessions.

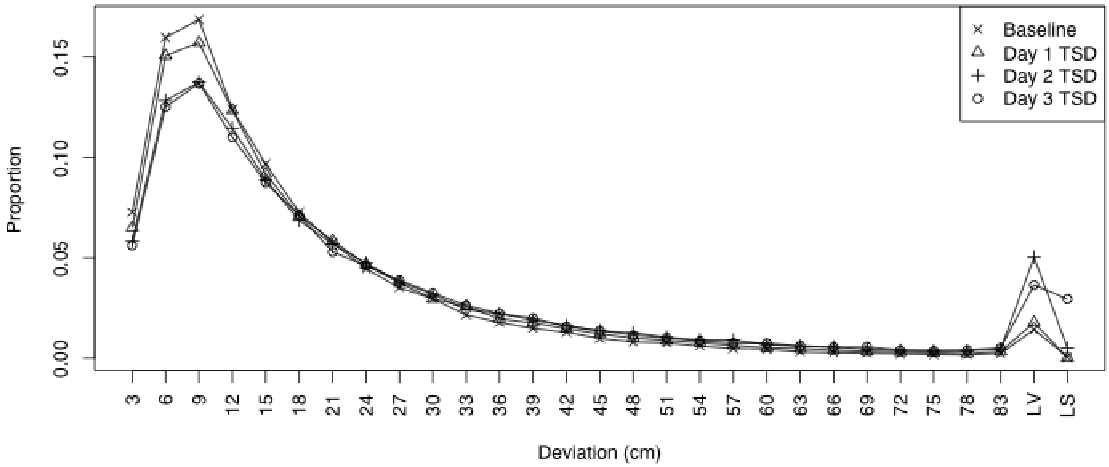


Figure 1: Proportion of 1-second samples of lateral deviation falling into each of the specified bins. The last two categories represent instances where (1) the vehicle is partway out of the proper lane (a lane violation, "LV"), and (2) the vehicle's deviation is more than a full lane width off (a lane switch, "LS"). Separate lines represent 0, 24, 48, and 72 hours of total sleep deprivation (TSD).

To assess the performance, the lateral deviation of the model was recorded for each second during each model run. Figure 1 shows a histogram of these deviation values as a function of degree of sleep deprivation (0, 24, 48, and 72 hrs). Perhaps surprisingly, the distributions are not radically different. Note however, that on the left side of the distribution the proportion of lower deviation values (3-12 cm) decreases with increasing sleep deprivation. The overall trend is toward an increasingly skewed distribution, where performance is basically normal most of the time, but diverges more often and to a greater extent as sleep deprivation increases. This pattern of results matches the data from the sustained attention task that we have used in developing the mechanisms applied to the driver model in this paper (see Gunzelmann, Gross, et al., 2009).

While the distributions in the larger deviations (21-80 cm) are not very different, clear differences emerge in the categories representing the largest deviations. Lane violations ("LV" in the figure) represent points when some portion of the vehicle had crossed the lane line (i.e., the vehicle overlapped the adjacent lane). The proportions of lane violations more than double for Days 2 and 3 of sleep deprivation as compared to the baseline day or a single night without sleep. The final category, lane shifts ("LS" in the figure), represent points during which the vehicle has moved an entire lane's width laterally — clearly a substantial degree of driver performance error. Whereas the Baseline and Day 1 conditions exhibit no lane shifts, there appear a small number of lane shifts in Day 2, and in Day 3, 3% of all lateral deviation values sampled are in this category. This means that 3% of the time, the model is driving completely out of its intended lane (possibly off the road or possibly into oncoming traffic).

To better understand the nature of this performance in terms of the driver model and fatigue mechanisms, Figure 2 shows a histogram of update times for the driver model in each condition — that is, the amount of time needed for the model to complete its four-production control update cycle. As was the case for lateral deviation, the distributions shift with increasing sleep deprivation such that update times reflecting cycles that are not interrupted (200-300 ms) become less frequent and longer update times become more prevalent. The increase in update times arises because production rules are more likely to fall below threshold under the influence of fatigue mechanisms, thus missing an opportunity to fire during a conflict resolution cycle.

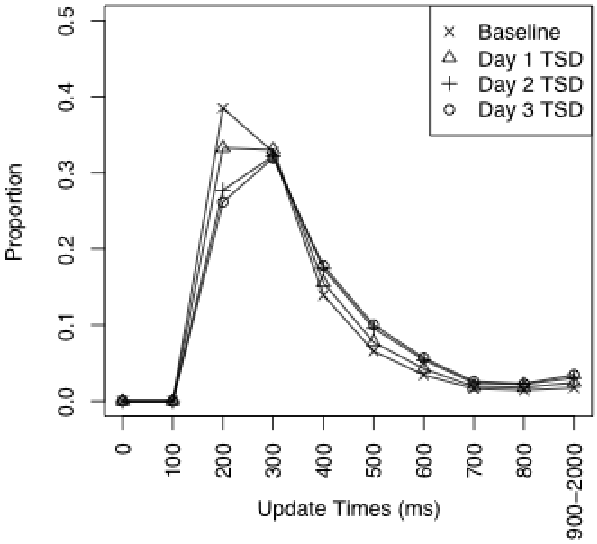


Figure 2: Distribution of model update times as a function of number of days of total sleep deprivation (TSD).

Comparison to Human Performance

To evaluate the model predictions in the context of actual human driver performance, we compared the model's performance to published results from a study of fatigued driving (Peters, Kloeppel, & Alicandri, 1999). Peters et al. (1999) measured lane violations during conditions of restricted sleep and sleep deprivation. Figure 3 compares the pattern of results from Peters et al. (1999) to the data from our model. The data from Peters et al. (1999) are frequency counts of lane violations, while the data from the ACT-R model reflect proportions of 1-second samples of lane deviation that exceeded the threshold for a lane violation. Though these measures are slightly different, they are closely related, and the pattern of results is identical (r=.99).

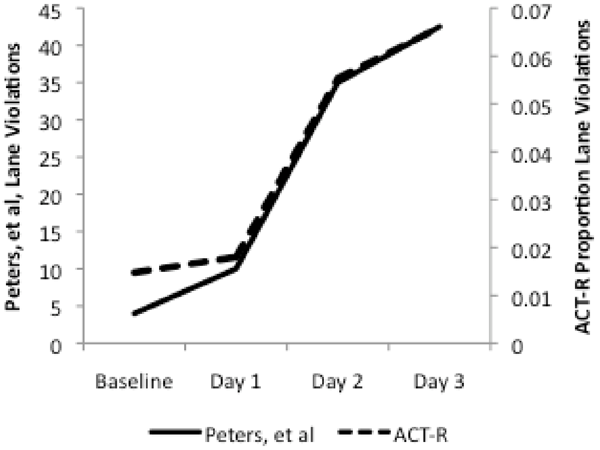


Figure 3: Lane violations from Peters et al. (1999) compared to the proportion of lane deviation samples classified as lane deviations or lane shifts in the model.

The Peters et al. experiment protocol was slightly different than the strict total sleep deprivation protocol assumed in our model predictions. Participants in Peters et al. (1999) were allowed four hours of sleep on the first night, between the Baseline Day and Day 1, whereas the parameters in the model assume total sleep deprivation. This could have some impact on the quantitative results, but the overall pattern would be similar in either case. The pattern is similar for both the human data and the model: only a slight performance decrement in Day 1, but a much larger decrement in Days 2 and 3. While the above caveat concerning the experiment protocol differences should be noted, these results suggest that the integration of the driver and fatigue models indeed captures an important aspect of fatigued driver behavior.

Conclusions and Future Directions

The model described in this paper exhibits declines in performance when mechanisms are implemented to represent the deleterious effects of sleep loss on central cognitive functioning. The foundation is a validated model of skilled driver behavior (Salvucci, 2006). That model is augmented with a set of mechanisms that account for changes in central cognitive processing that result from increased levels of fatigue associated with time awake and circadian rhythms (Gunzelmann, Gross, et al., 2009).

The primary contribution of this research is the demonstration that it is possible to make truly a priori predictions regarding the effects of extended wakefulness on performance in complex, dynamic tasks. The qualitative changes in the model's performance are identical to the performance changes observed in human participants attempting to drive after extended periods of partial or total sleep deprivation. The results go beyond intuitive notions regarding degradations in cognitive processing and performance as time awake increases by providing quantitative estimates about the actual impact of those changes on performance in the driving task.

Of course, qualitative comparisons of overall performance falls short of the rigorous evaluation of the model that we would like to perform. However, the current research effort represents a critical step in the process of using computational cognitive modeling to make predictions about human cognition and behavior in naturalistic task contexts. The modular design of ACT-R facilitates this convergence of research efforts by providing an infrastructure that allows new theoretical components (like the account of fatigue) to be added seamlessly to the architecture. Once added, these new components, or modules, influence the model's behavior to the extent that the proper conditions arise to activate the mechanisms. In this case, the mechanisms for fatigue have a substantial impact on model behavior. Importantly, the impact appears to be in line with human data on a similar task in the research literature.

A major goal of research on fatigue is to develop an understanding of the impact of sleep loss that is useful in making predictions regarding the consequences for performance in applied settings. At the outset, we cited the enormous cost of fatigue – both in dollars and lives – on highways in the United States. A better understanding of the relationship between fluctuations in alertness and changes in observable human behavior has the potential to greatly reduce this cost, potentially saving thousands of lives. Moreover, driving is not the only area where the potential benefits exist. In many applied settings, lack of sleep and circadian desynchrony may lead to disastrous consequences (e.g., Caldwell, Caldwell, Brown, & Smith, 2004; Dinges, 1995). Accurate predictions of the consequences of fatigue could help to avert some of these potential tragedies.

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was sponsored by the Air Force Research Laboratory's Warfighter Readiness Research Division, grant

07HE01COR from the Air Force Office of Scientific Research, and grant #N00014-09-1-0096 from the Office of Naval Research.

References

Anderson, J. R. (2007). *How Can the Mind Exist in a Physical Universe?* Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*, 1036-1060.

Caldwell, J. A., Caldwell, J. L., Brown, D. L., & Smith, J. K. (2004). The effects of 37 hours of continuous wakefulness on the physiological arousal, cognitive performance, self-reported mood, and simulator flight performance of F-117A pilots. *Military Psychology, 16(3)*, 163-181.

Dijk, D-J., Duffy, J. F., & Czeisler, C. A. (1992). Circadian and sleep/wake dependent aspects of subjective alertness and cognitive performance. *Journal of Sleep Research, 1*, 112-117.

Dinges, D. F. (2004). Critical research issues in development of biomathematical models of fatigue and performance. Aviation, Space, and Environmental Medicine, 75(3), A181-A191.

Dinges, D. F. (1995). An overview of sleepiness and accidents. *Journal of Sleep Research*, *4(2)*, 4-11.

Dinges, D. F., & Kribbs, N. B. (1991). Performing while sleepy: effects of experimentally-induced sleepiness. In Monk, T.H. (Ed.) *Sleep, sleepiness and performance* (pp. 97-128). Chichester: John Wiley & Sons.

Gunzelmann, G., Byrne, M. D., Gluck, K. A., & Moore, L. R. (2009). Implications of Computational Cognitive Modeling for Understanding Fatigue. *Human Factors, 51(2)*.

Gunzelmann, G., & Gluck, K. A. (in press). An Integrative Approach to Understanding and Predicting the Consequences of Fatigue on Cognitive Performance. *Cognitive Technology*.

Gunzelmann, G., Gluck, K. A., Kershner, J., Van Dongen, H. P. A., & Dinges, D. F. (2007). Understanding decrements in knowledge access resulting from increased fatigue. In D. S. McNamara and J. G. Trafton (Eds.), *Proceedings of the Twenty-Ninth Annual Meeting of the Cognitive Science Society* (pp. 329-334). Austin, TX: Cognitive Science Society.

Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*.

Horne, J., & Reyner, L. (1999). Vehicle accidents related to sleep: A review. *Occupational and Environmental Medicine, 56*, 289-294.

Klauer, S.G., Dingus, T. A., Neale, V. L., Sudweeks, J.D., & Ramsey, D.J. (2006). The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data. *Technical Report #DOT HS 810 594*. Washington, DC: National Highway Traffic Safety Administration.

Mallis, M., Mejdal, S., Nguyen, T., & Dinges, D. (2004). Summary of the key features of seven biomathematical models of human fatigue and performance. *Aviation, Space, and Environmental Medicine, 75(3)*, A4-A14.

National Transportation Safety Board (1995). Factors that affect fatigue in heavy truck accidents (volume 1). NTSB Safety Study: *Report no. NTSB//SS-95/01 (PB95-9170012)*. Washington, DC: NTSB.

Pack, A. I., Pack, A. M., Rodgman, E., Cucchiara, A., Dinges, D. F., & Schwab, C. W. (1995). Characteristics of crashes attributed to the driver having fallen asleep. *Accident Analysis & Prevention, 27(6)*, 769-775.

Peters, R.D., Kloeppel, E., & Alicandri, E. (1999). Effects of partial and total sleep deprivation on driving performance. *Publication No. FHWA-RD-94-046*, Washington, DC: Federal Highway Administration, US Department of Transportation.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*, 201-220.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, *48*, 362-380.

Salvucci, D. D., & Beltowska, J. (2008). Effects of memory rehearsal on driver performance: Experiment and theoretical account. *Human Factors*, *50*, 834–844.

Salvucci, D. D., & Gray, R. (2004). A two-point visual control model of steering. *Perception*, *33*, 1233-1248.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, *115*, 101-130.

Van Dongen, H. P. A. (2004). Comparison of mathematical model predictions to experimental data of fatigue and performance. Aviation, Space, and Environmental Medicine, 75(3), A15-A36.

Van Dongen, H. P. A., & Dinges, D. F. (2005a). Circadian rhythms in sleepiness, alertness, and performance. In M. H. Kryger, T. Roth, & W. C. Dement (Eds.), *Principles and practice of sleep medicine, 4th ed.* (pp. 435-443). Philadelphia, PA: Elsevier Saunders.

Van Dongen, H. P. A., & Dinges, D.F. (2005b). Sleep, circadian rhythms, and psychomotor vigilance. *Clinical Sports Medicine, 24*, 237-249.

Multi-Associative Memory in fLIF Cell Assemblies

Christian R. Huyck (c.huyck@mdx.ac.uk)

Kailash Nadh (k.nadh@mdx.ac.uk)

School of Engineering and Information Sciences
Middlesex University
London, UK

Abstract

The fundamental mammalian behaviours of perception, recognition, recollection, and all other psychological phenomena are intrinsically related to the basic cognitive tasks of memorisation and association. Based on Hebb's Cell Assembly (CA) theory, it is believed that concepts are encoded as neuronal CAs in mammalian cortical areas. This paper describes a series of simulations that demonstrate various associative memory tasks using CAs based on biologically plausible fatiguing, Leaky, Integrate and Fire neurons. The simulations show the ability of CAs to form, retain and recollect basic concepts and multiple and sequential associations.

Keywords: Cell Assemblies; Multi-associative memory; fLIF neurons

Introduction

Associative memory is a fundamental cognitive process. The concepts in memory and the associations between them are learned. These concepts and associations are critical to cognitive processing.

Like all cognitive processes, associative memory must have a neural basis, but neural models of associative memory are rare and surprisingly incomplete. Cell Assemblies (CAs) can account for many cognitive phenomena, including associative memory. Concepts can be stored as CAs (see Section CAs and auto-associative memory), and associations can be stored in connections between CAs.

Associative memory has a wide range of properties. Concepts can be connected in one to one, one to many, and many to many relationships. Associations can be context sensitive. In this paper, simulated CAs are used to explore these properties performing different tasks including a simple spatial cognitive mapping task. Cognitively, a good associative memory model should be capable of priming, differential associations, timing, gradual learning and change, encoding instances, and many such processes. The model simulations do not account for these phenomena, but this is the beginning of an exploration of a model that will (see Section Discussion and conclusion).

Background

Human associative memory is remarkable. Throughout life, new concepts are learned and new associations formed. Any given concept is associated with many other concepts, and retrieval of an associated concept can be based on a combination of the base concept and the context. Priming studies, for example, show the memory system supports a wide range type and strength of associations between concepts. Memory retrieval and formation of associations are rapid processes.

Simulated neural models of associative memory are not currently capable of many of the tasks described in the prior paragraph. Closely related connectionist models have however been used to perform some of them.

CAs and auto-associative memory

Hebb (1949) hypothesised that the CA is the neural basis of concepts, and the CA is central to most neural models of memory. The theory proposes that objects, ideas, stimuli and even abstract concepts are represented in the brain by simultaneous activation of large groups of neurons with high mutual synaptic strengths (Wennekers & Palm, 2000). If an external stimulus excites a sufficient number of neurons of an existing CA, it can result in the spreading of activation within the CA, in turn igniting it due to recurrent activity and high mutual synaptic strength. The CA can then remain active even after the stimulus is removed. This reverberating behaviour accounts for short term memory.

CAs are learned using the Hebbian learning rule, whereby modifications in the synaptic transmission efficacy are driven by the correlations in the firing activity of pre-synaptic and post-synaptic neurons (Gerstner & Kistler, 2002). When external stimuli are presented to a network, synaptic strength between neurons are adjusted so as to gain more strength if they undergo repeated and persistent activation or firing, gradually assembling them into a group, a CA. This formation of CAs accounts for long term memory. Thus, the CA hypothesis provides a structural and functional account for such cortical processes.

While still unproven, there is significant evidence and wide spread agreement that CAs are the neural basis of concepts. This includes a range of neural recording mechanisms (Abeles, Bergman, Margalit, & Vaddia, 1993; Bevan & Wilson, 1999; Pulvermuller, 1999).

The CA is a form of auto-associative memory. In auto-associative memories, an initial state is allowed to settle into a stored memory, allowing subsequent noisy input to retrieve a stored pattern. The Hopfield Model illustrates this property (Hopfield, 1984). A network of units that are well connected with bidirectional weighted connections is used to store a set of binary patterns (typically using a Hebbian calculation). When an initial set of neurons is switched on, in the discrete version of the system, activation spreads through the system based on the weighted connections. In most cases the system will settle into a stable state with no neurons switching between on and off. If the input pattern is close to a stored pattern, it will settle into that pattern's state, thus functioning as a content-addressable memory. Neurons may also belong to multiple CAs. Hopfield patterns that share on-bits are models of CAs that share neurons.

While CAs are critical for the model of multi-associative memory described in this paper, they are not the solution. The question is how different CAs are associated with each other.

Multi-associative memory

Auto-associative memory is not typically what is meant by associative memory. Instead, associative memory is generally a shortened form (usually implicitly) of multi-associative memory; this has also been called hetero-associative memory. Psychologically, memories are not stored as individual concepts, but large collections of associated concepts that have many to many connections (Anderson & Bower, 1980). Each memory (CA) is associated with many other memories (CAs).

CAs and multi-associative memory

Even though CAs account for memory formation, their precise neural dynamics are far from perfectly understood. As explained in the Section CAs and auto-associative memory, neurons may belong to different CAs, and if they are repeatedly co-activated by different versions of the same stimulus, they tend to become associated (Hebb, 1949). This is based on the notion that events that occur together repeatedly should somehow belong together. Wennekers and Palm (2000) explained that every time these events occur in conjunction, they drive certain subgroups of neurons, their correlated firing should be learned, and, by that, respective groups should become associatively connected.

Repeated co-activation of neurons can lead to the formation of CAs. Similarly, repeated co-activation of multiple CAs results in the formation of multiple and sequential associations, and sometimes new CAs. When an external stimulus activates a CA, it might lead to the activation of neurons that ignites a different CA that is not directly stimulated. This forms the rudimentary, neural level explanation of associative memory. Humans constantly retrieve and form associations with whatever sensory input they receive for the purpose of perception, understanding and reasoning.

Multi-associative memory models

Many multi-associative memory models have been proposed. A select few models are reviewed below.

Non-Holographic Associative Memory is an early multi-associative memory model (Willshaw, Buneman, & Longuet-Higgins, 1969). It is a well-connected network that can learn to map input bit patterns to output bit patterns using a Hebbian learning mechanism. In CA terms, input CAs are connected to output CAs via learned one way associations. This is a one step model. **The Linear Associator** (Kohonen, 1977) is a similar model that, like many other models, encodes memories in well connected systems. The brain is not well connected, but it is often argued that it is broken into compartments that are well connected (Amit, 1989).

The **Multi Modular Associative Memory** (Levy & Horn, 1999) used well connected modules and analysed the storage capacity of a system with items stored in multiple modules. It showed that such a multi modular network is resilient to corrupted input, based on their observation that natural associated memories remain resilient to a great extent in humans who suffer from focal damage. They concluded that multi modular networks are necessary for meaningful implementation of associative neural networks. This is supported by evidence that shows that the memory for a given word is stored in multiple areas of the brain (Pulvermuller, 1999).

The **Valiant model** (Valiant, 2005) is a graph theoretical model of memorisation and association based on four quantitative parameters associated with the cortex: the number of neurons per concept; number of synapses per neuron; synaptic strengths; and number of neurons in total. It is assumed that neurons are randomly connected. The learning algorithm provided is biologically implausible, but the model shows that random graphs allow a method of assigning new memory items and associative relationships between the items.

The **Jets and Sharks** simulation (McClelland, 1981) uses the interactive activation model (Rumelhart & McClelland, 1982) to simulate associative memory. In the model, each concept is represented by a node, and connections are made between nodes to show how closely related these are. The system is not well connected. Activation spreads between the nodes via the weighted connections. The information to be encoded concerns two hypothetical groups (*Jets* and *Sharks*), group members, and some of their demographic characteristics. The system can act as a content-addressable memory system. So, the features of an individual group member can be activated as input, and the individual's representation will quickly become activated by the spread of activation. Additionally, prototypical effects can be derived (Rosch & Mervis, 1975). So, if the *Shark* concept is stimulated, activation will spread and eventually, the prototypical shark will become more active than other individuals. The individual that shares most features with other *Sharks* is the prototypical member.

This has been a brief review of multi-associative memory models. It has been known for 40 years that simulated neural systems can encode multi-associative memories, but it has become apparent that well connected systems are not a good model of the brain. This has been addressed by partitioning the system into modules, and by using sparsely connected random graphs. These models however do not account for a range of associative memory characteristics that the human memory system exhibits, for instance, context effects.

The simulator

This section briefly describes a computational model that simulates CAs using fLIF neurons. Like all models, it is a simplification of the mammalian neural architecture, but has proven successful in modelling many cognitive phenomena.

The fLIF neural network

The fLIF neuron model (Huyck, 2007) encompasses many properties of the biological neuron. The CAs used in the experiments described in this paper emerge from fLIF neural networks. The model is an extension of the LIF (Leaky Integrate and Fire) model (Maas & Bishop, 2001; Gerstner, 2002). fLIF neurons collect activation from pre-synaptic neurons and fire on surpassing a threshold, that is, they integrate and fire. On firing, a neuron loses its activation level, otherwise the activation leaks gradually, resembling the behaviour of a biological neuron.

The activation A of a neuron i at time t is:

$$A_{i_t} = \frac{A_{i_{t-1}}}{\delta} + \sum_{j \in V_i} w_{ij} s_j \tag{1}$$

The current total activation is the activation from the last time step divided by decay factor δ, plus incoming activation. This new activation is the sum of the active inputs s_j of all neurons $j \in V_i, V_i$ being the set of all neurons connected to i, weighted by the connection from neuron j to i. The neuron fires when the accumulated activation A exceeds a threshold θ, and firing neurons do not retain activation. Firing is a binary event, and activation of w_{ij} is sent to all neurons j to which the firing neuron i has a connection. Fatiguing causes the threshold to be dynamic, $\theta_{t+1} = \theta_t + F_t$. F_t is positive (F_+) if the neuron fires at t and negative (F_-) if it does not.

The network architecture

Two of the three the simulations discussed in this paper partitions the network into subnetworks; the context simulation uses only one subnet. The subnets are made of fLIF neurons and the number of neurons vary between subnets. Intrasubnet synapses are based on biologically inspired distance biased connections. This topology makes it likely for a neuron to have excitatory connections to neighbouring neurons, and less likely to far away ones. The subnet is a rectangular array of neurons with distance organized toroidally. Inhibitory connections within a subnet and all inter-subnet connections are set randomly. The connectivity rule for excitatory neurons is given by equation 2. There exists a connection between neuron i and j of a network only if $C_{ij} = 1$.

$$\begin{aligned} C_{ij} &= 1, \text{if } r < (1/(d * v)) \quad (2) \\ C_{ij} &= 0, \text{if not} \end{aligned}$$

where r is a random number between 0 and 1, d is the neuronal distance and v is the connection probability. This indicates that connections in a network are influenced by distance between neurons and the connection probability factor. Distance $d = 5$ throughout all the simulations, as it has been observed to work well. Inspired by biological neural topology, long distance intra-network connections are also present, connected by long distance axons with many synapses (Churchland & Sejnowski, 1992).

In each of the simulations, networks are divided into multiple CAs using unsupervised Hebbian. The CAs are orthogonal and represent different concepts, and this is in response to training stimuli. Neurons in different CAs do have excitatory connections to other CAs, based on the connection rule (Equation 2), but the learned weights are low because neurons in different CAs rarely co-fire. Once learned, when a CA is externally activated, it typically inhibits all inactive CAs in the same network via learned inhibitory connections. Similarly, simultaneous co-activation of CAs increases the connection strength between them, creating associations.

Learning in the network

CAs in a network are learned by a correlatory Hebbian learning rule (Huyck, 2004), whereby synaptic connection weights are modified based on the following equation:

$$\Delta^+ w_{ij} = (1 - w_{ij}) * \lambda \tag{3}$$

$$\Delta^- w_{ij} = w_{ij} * -\lambda \tag{4}$$

w_{ij} is the synaptic weight from neuron i to j and λ is the learning rate. During each cycle, weights change based on the state of pre-synaptic and post-synaptic neurons. If both neurons fire, the weights increase as per the Hebbian rule (Equation 3). If only the pre-synaptic neuron fires, weights decrease as per the anti-Hebbian rule (Equation 4). These two rules act together, changing w_{ij}, gradually increasing the likelihood of j firing if i fires. Without reverberation, the weight would reflect the likelihood that neuron j fires when neuron i fires.

The network parameters used in the simulations are presented in the table 1. The decay parameter has a link to biological data, but the others have been selected via a search of the space. In particular, the fatigue parameters are different across the three experiments described below.

Table 1: Network parameters

Parameter	Symbol	Value
Learning rate	λ	.10
Activation threshold	θ	4.5
Fatigue	$F_+ = F_-$	.80
Decay factor	δ	1.2
Neuronal distance	d	5

Simulations

This section describes three sets of simulations. These simulations demonstrate that the model is capable of supporting complex associations.

Jets and Sharks

This is a CA based implementation of a modified version of the classic Jets and Sharks model that uses five members in each of the hypothetical (*Jets* or *Sharks*) groups. The original experiment had 27 members, but 10 randomly selected member were used here to demonstrate feasibility. Each of the members and their attributes are encoded as CAs in different subnets. A unique *Person* CA represents each member and their attributes, namely *Name*, *Age*, *Education*, *Marital status*, and *Occupation*. There is a one-to-one relationship between each *Person* and their *Name* CA. A subset is illustrated in Figure 1.

Art - *Jet*, *40's*, *Junior High*, *Single*, *Pusher*
Phil - *Shark*, *30's*, *College*, *Married*, *Pusher*

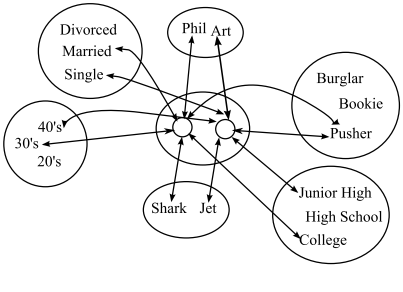


Figure 1: Two people in Jets and Sharks. Circles refer to subnets, names to orthogonal CAs, and arrows to connections.

Each CA is mutually exclusive and is made up of 200 neurons. Inter-subnet connections are random, initially low-weight excitatory connections. The CAs and their associations are learned by external stimulation and co-activation of each *Person* CA and their attributes simultaneously for 200 cycles, in succession.

Multiple memory retrieval tests were conducted. For instance, when the *Name* CA of **Art** is externally stimulated, it propagates activation to *Art*'s *Person* CA. The particular *Person* CA, having learned excitatory connections to different attributes, causes activation to further propagate throughout the network, gradually activating all corresponding attributes of **Art**. On 15 runs, the correct results were retrieved for each of the 10 people. This shows one to one associations (e.g. **Art** to his name), one to many (e.g. **Art** to all his properties) and many to many (e.g. **Pusher** is activated by many people along with other properties).

Similarly, when the attribute *Shark* is externally activated, it propagates activation to all *Person* CAs having that attribute, and the immediate effect is that all *Shark* members ignite Though these CAs do not share neurons, multiple CAs in a subnet may be simultaneously active. Gradually, activations stabilise through competition between CAs. One *Person* CA is found to have more activation than others, emerging as the prototypical *Shark*. The network was tested 15 times to obtain the prototypical *Jet* and **Art** emerged to be so, throughout. The same was done for *Sharks* and **Nick** emerged to be prototypical 9 times and **Ned**, 6 times. This is because both members share the most features with other members of the group, and hence emerge to be prototypical members.

Context sensitive association

Most associative memory models, focusing solely on associations, usually neglect to acknowledge the inherent *types* of associations that exist. A concept may be associated to many others, but the types of associations may vary from concept to concept. The association of *cat* to *mammal* is not the same as *fur* to *mammal*. As a step towards simulating different types of associations and eventually implicit labelled associations, a model capable of differentiating associations based on contexts was developed.

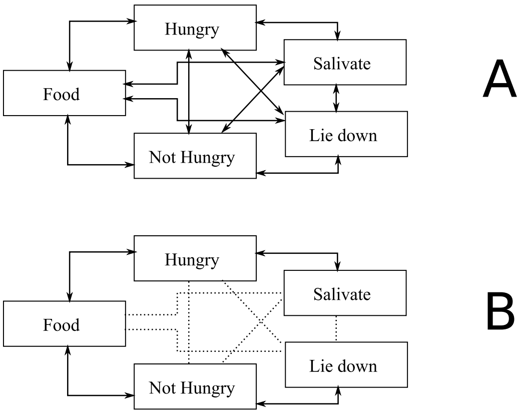


Figure 2: Initial and Learned state of CAs

Figure 2 shows the network setup, the physical connection before (A) and after (B) the CAs and their associations are learned. A single network holds all the 5 orthogonal CAs, namely *Hungry, Not Hungry* (states), *Salivate, Lie down* (actions) and *Food* (object). Since all the CAs are in the same network, they have excitatory and inhibitory connections with each other. The parameters in the simulation are those from Table 1, except the fatigue parameters have been modified. $F_+ = F_- = 0.4$. Initially, patterns corresponding to each of the CAs are presented for 300 cycles so that they are learned independently. When a CA is active, it inhibits all other CAs in the network via learned inhibitory connections. When one CA is active and another is inactive, inter-CA connection weights are decreased. The associations between CAs are learned by co-activation for 300 cycles each, that is, by activating three CAs (object, state, action) simultaneously, in the following manner:

$$\textit{Food} + \textit{Hungry} \Rightarrow \textit{Salivate}$$
$$\textit{Food} + \textit{Not Hungry} \Rightarrow \textit{Lie down}$$

This mimics the behaviour of a hypothetical *dog* that salivates when food is presented when hungry, and lies down ignoring food when not hungry. After the associations are learned, context sensitive behaviour is tested in the following manner: when *Food* and *Hungry* are externally stimulated, *Salivate* activates, suppressing *Lie down*. The tests were repeated on 100 different network configurations, and action CAs (*Salivate, Lie down*) activated correctly 83 times with an average of 84.6 neurons firing.

Igniting any one CA leads to activity in one associated CA, and in less than 20% of trials the third associated CA. No unassociated CAs have been activated in simulations.

Cognitive spatial mapping using sequential memory

Cognitive spatial mapping is a psychological process by which an individual acquires, stores, recalls and decodes information about the relative locations and attributes of a spatial environment for the purpose of spatial navigation (Downs & Stea, 1973). A simplified version of this complex process was implemented, where a virtual agent navigates a 3D virtual world by recognising, memorising, associating and recollecting rudimentary landmarks. The parameters in the simulation are those from Table 1, except the fatigue parameters have been modified. $F_+ = F_- = 0.1$. The change of these parameters between the three simulations has merely been an engineering decision based on a simple exploration of the parameter space. It is likely that different topologies using the same parameters would have also produced similar results.

Figure 3 shows the top view of the virtual world, its 4 rooms, 4 coloured doors, and the exploration path the agent takes. The agent's path is fixed and it lacks the ability to turn back and only moves forward.

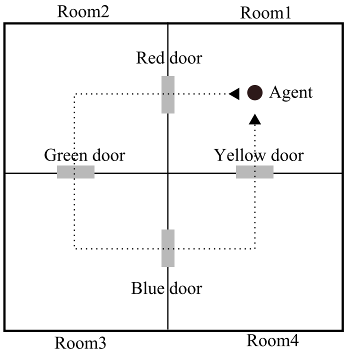


Figure 3: Top view of the virtual world

In the learning mode, the agent explores the world, learning rooms, doors and *Room-Door-Room* sequences in the process. A simple vision system detects doors and door colours, and triggers learning actions, helping the agent navigate.

Figure 4 illustrates the gross subnetwork topology of the spatial mapping module, excluding other subnets of the agent. The solid arrows show physical inter-subnet connections (random, low-weight excitatory connections). **RoomNet1** and **RoomNet2** store instances of the rooms the agent visits. **DoorNet** stores the doors encountered and **SequenceNet**, encodes the sequences of visits. **ColourNet** has CAs that represent colours recognised by the agent, and **GoalNet** encodes the target door, which the agent searches for while in the test mode. The greyed areas show a sample sequence, where the agent has learned the association *Room1⟷Red-Door⟷Room2* by co-activation. The dashed lines represent learned connections with increased synaptic weights.

The CAs representing rooms, doors and sequences in corresponding subnets are made up of 200 neurons each, and are learned as the agent explores. In the learning mode, when the agent encounters a door, a 5-step learning process is triggered: 1) The agent encodes its present location as a CA in both the **RoomNets**; 2) It learns the door, forming a CA in the **DoorNet**; 3) The agent associates the colour of the door in the **ColourNet** with the newly formed door CA; 4) The agent moves to the next room and learns the room (as in Step 1); 5) It then encodes the passage it just made as a CA in the **SequenceNet**, as in *PreviousRoom-ConnectingDoor-PresentRoom*. This process is repeated until the agent is back at its starting position. Each of the CAs are learned by stimulation lasting 300 cycles, triggered by the visual cues the agent reviews. Associations are learned by co-activation, as described in the previous simulations. For instance, passages are learned by simultaneously activating the corresponding sequence CA in the **SequanceNet**, pre-entrance room CA in **RoomNet1**, the connecting door CA in **DoorNet** and the present room CA in **RoomNet2**, for 300 cycles.

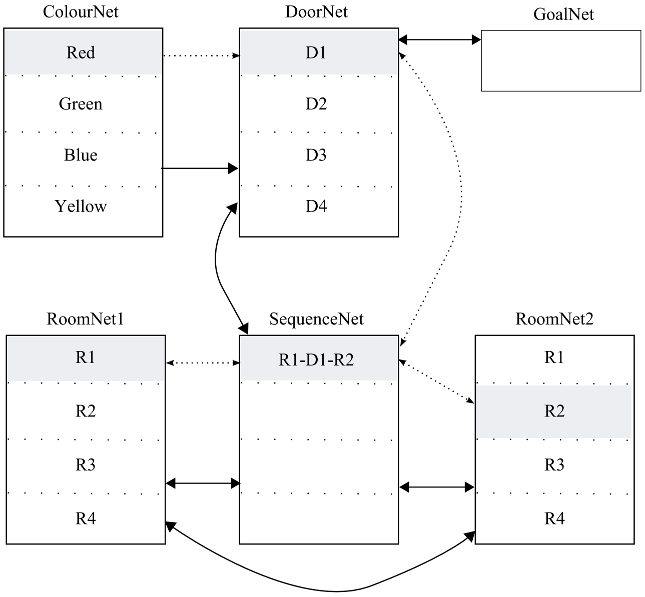


Figure 4: Cognitive spatial mapping network setup

In the test mode, the agent is instructed to go to a room randomly chosen from the 4 rooms. This is done by externally stimulating the target room CA in *RoomNet2*. This causes the corresponding sequence CA to activate, which in turn activates the associated room CA in **RoomNet1** and the connecting door CA in **DoorNet**. When the door CA becomes active, the goal CA is activated externally, leaving them to remain simultaneously active for 300 cycles, causing them to become associated. As a result of this association, the goal CA becomes active whenever the corresponding door CA activates. The active door CA that the agent has set as its goal is the door that leads to the target room. With the goal in memory, the agent moves forward, looking for the target landmark (door). When the target door appears in the agent's visual field, the corresponding door CA in **DoorNet** activates, immediately causing the goal CA to activate due to the previously learned association, indicating achievement of the goal. With this, the agent reaches the target room and the test ends.

The agent could have failed by stopping prematurely, or continuing beyond the target room. However, it correctly reached a the target room all 30 times the test was repeated.

Spatial cognitive mapping is an important associative task. The task is also important for agents, and this cognitive mapping module is currently being incorporated into our current Cell Assembly roBot (CABot3), an agent in a video game based solely on fLIF neurons.

Discussion and conclusion

These simulations show that CAs emerging from model fLIF neurons are capable of learning and retrieving core memories, in the form of CAs, and associations between them. The Jets and Sharks simulations show that the model can handle one to one, one to many, and many to many relations. The second set of simulations show that the model is capable of handling context sensitive associations, and the third set shows that it is capable of a basic form of cognitive mapping, using multi-associative sequential memories. This is the first neural model that simulates all the these processes.

While these are useful capabilities, the model does not exhibit the wide range of behaviours that human associative memory does. Human memories have varying strengths, and so do the associations. Instances of types (tokens) can be learned. Types, associations and tokens are all forgettable. All of these behaviours occur in measurable times. It is planned that future work will include all of these behaviours.

In the simulations described in this paper CAs were orthogonal, that is, neurons were in only one CA, and associations were maintained solely by synapses between neurons in the associated CAs. Another type of association is possible, where CAs are associated by sharing common neurons and subcategorisation associations have been stored using shared neurons in CAs (Huyck, 2007) . For example, the concept *Cat* shares neurons with the concept *Mammal* because of the association that a *Cat isA Mammal*. It is likely that such overlapping CAs are important for a good neural implementation of multi-associative memory.

Other properties may also be necessary to achieve the full range of associative memory behaviours. For instance, global inhibitory mechanisms might be needed to manage spreading of activation and prevent all neurons firing simultaneously. None the less, the current simulations show simulated neural systems can perform a range of associative memory tasks. The authors leave the reader with these questions: what tasks does an associative memory perform, and what are good tests to show that a system performs these tasks?

References

Abeles, M., Bergman, H., Margalit, E., & Vaddia, E. (1993). Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *Journal of Neurophysiology*, *70(4)*, 1629–1638.

Amit, D. (1989). *Modelling brain function: The world of attractor neural networks.* Cambridge University Press.

Anderson, J. R., & Bower, G. H. (1980). *Human associative memory: A brief edition.* USA: Lawrence Erlbaum Assoc.

Bevan, M., & Wilson, C. (1999). Mechanisms underlying spontaneous oscillation and rhythmic firing in rat subthalamic neurons. *Neuroscience*, *19*, 7617–7628.

Churchland, P., & Sejnowski, T. (1992). *The computational brain.* USA: MIT Press.

Downs, R. M., & Stea, D. (1973). Cognitive maps and spatial behaviour. process and products. In *Image and environment: Cognitive mapping and spatial behaviour* (pp. 8–26). Chicago: Aldine.

Gerstner, W. (2002). Integrate-and-fire neurons and networks. In *The handbook of brain theory and neural networks* (pp. 577–581). The MIT Press.

Gerstner, W., & Kistler, W. (2002). Mathematical formulations of hebbian learning. *Biological Cybernetics*, *87*(5-6), 404–415.

Hebb, D. O. (1949). *The organization of behavior.* New York: Wiley.

Hopfield, J. J. (1984). Neurons with graded response have collective computation properties like those of two-state neurons. In *Proceedings of the national academy of sciences* (Vol. 81, pp. 3088–3092).

Huyck, C. (2004). Overlapping cell assemblies from correlators. *Neurocomputing*, *56*, 435-9.

Huyck, C. (2007). Hierarchical cell assemblies. In *Connection science* (Vol. 19:1, pp. 1–24).

Kohonen, T. (1977). *Associative memory - a system theoretical approach.* Berlin: Springer-Verlag.

Levy, N., & Horn, D. (1999). Associative memory in a multimodular network. *Neural Computation*, *11*, 1717–1737.

Maas, W., & Bishop, C. M. (2001). *Pulsed neural networks.* MIT Press.

McClelland, J. L. (1981). Retrieving general and specific information from stored knowledge of specifics. In *Proceedings of the third annual conference of the cognitive science society* (pp. 170–2).

Pulvermuller, F. (1999). Words in the brain's language. *Behavioral and Brain Sciences*, *22*, 253–336.

Rosch, E., & Mervis, C. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, *7*, 573–605.

Rumelhart, D., & McClelland, J. (1982). An interactive activation model of context effects in letter perception: Part 2. the contextual enhancement and some tests and extensions of the model. *Psychological Review*, *89:1*, 60–94.

Valiant, L. G. (2005). Memorization and association on a realistic neural model. *Neural Computation*, *17*, 527–555.

Wennekers, T., & Palm, G. (2000). Cell assemblies, associative memory and temporal structure in brain signals. In *Conceptual advances in brain research* (Vol. 2).

Willshaw, D., Buneman, O., & Longuet-Higgins, H. (1969). Non-holographic associative memory. *Nature*, *222*, 960–962.

A computational model of three non-word repetition tests

Gary Jones (Gary.Jones@NTU.ac.uk)
Division of Psychology, Nottingham Trent University, Chaucer Building, Goldsmith Street
Nottingham, NG1 5LT (UK)

Abstract

The non-word repetition test has been regularly used to examine children's vocabulary acquisition, and yet there is no clear explanation of all of the effects seen in non-word repetition. This paper presents a study of 25 5-6 year-old children's repetition performance on three non-word repetition tests that vary in the degree of their lexicality. EPAM-VOC, a model of children's vocabulary acquisition, is then presented that captures the children's performance in all three repetition tests. The model represents a clear explanation of how working memory and long-term linguistic knowledge interact in a way that is able to simulate performance in non-word repetition.

Keywords: Computational modelling; Non-word repetition; Child development.

Introduction

One ability that sets the human species apart from other species is that of language. However, the learning of language is a complicated process that involves at least the following processes. First, the learner must identify where words begin and end from speech that is often continuous. Second, the learner must store the newly identified words in their long-term lexicon. Finally, the learner must acquire the rules of syntax and grammar that govern the way in which their lexicon words can be combined. It is the second of these three processes that this paper is focused: the process of vocabulary learning.

Research that examines vocabulary learning is proliferated with tests of non-word repetition – a test that involves nonsense words being spoken aloud to the language learner, who must repeat them accurately. The test involves non-words since one can be certain that the child has never encountered the sequence of sounds before, hence providing a true test of vocabulary learning. Furthermore, studies of vocabulary involving non-word repetition have primarily focused on children, since the vast majority of language learning occurs early in one's development.

Non-word repetition research

Non-word repetition tests were originally developed to examine the influence of phonological working memory on the vocabulary learning process. For example, Gathercole and Baddeley (1989) showed that repetition accuracy improved between the ages of 4 and 5 years, and performance declined as non-word length increased for both ages. Both of these findings were interpreted in terms of phonological working memory: an improvement with age could be explained by an increase in memory capacity; and a decrease in performance as non-word length increased could be explained by the decay of items in working memory.

However, subsequent research has shown that the child's existing lexical knowledge plays a major role in their non-word repetition ability. Gathercole (1995) re-analysed the non-words in the original test by separating them into "wordlike" and "non-wordlike" non-words based on adult subjective ratings of wordlikeness. She found that children performed significantly better for non-words that were wordlike. Although wordlikeness is a subjective measure, even when more objective measures are used, there are still clear differences between non-words that share substantial lexical features with words compared to those that do not. For example, if one actively distinguishes non-words based on their constituent phoneme combinations – having one set that contain highly frequent combinations of sounds versus a set containing relatively infrequent combinations – there are clear performance differences, with children regularly finding the high-frequency non-words easier to repeat (e.g. Edwards, Beckman & Munson, 2004; Vitevich, Luce, Charles-Luce & Kemmerer, 1997).

It would therefore seem that non-word repetition, and in turn vocabulary acquisition, can be affected by both phonological working memory and long-term lexical knowledge. There are at least two prominent explanations of vocabulary acquisition that explain repetition performance in terms of both processes.

Explanations of non-word repetition performance

Since non-word repetition performance is affected by an interaction between working memory and long-term memory, any explanation of performance must provide some detail of how these two processes interact. Gathercole (2006) explained this interaction using the idea of phonological frames. Phonological working memory is used to store linguistic stimuli (e.g. non-words in the repetition test) and when these items decay, long-term linguistic knowledge is relied upon to help bolster the decaying representations in working memory. Since non-words that are wordlike, or that contain highly-frequent sounds, will share more information with lexical items in long-term memory, it is these items that gain more help from existing vocabulary knowledge. That is, the support provided by the phonological frames of existing vocabulary items increases as the amount of overlap in shared features (to non-words) increases.

An alternative explanation of vocabulary learning shares many features with Gathercole's idea of phonological frames, yet is more explicit in its detail. Jones, Gobet and Pine's (2007, 2008) EPAM-VOC is a computational model of vocabulary learning that concretely specifies how phonological working memory and long-term phonological knowledge interact. Long-term knowledge is represented by "chunks" of phoneme sequences – as the model is subjected to more and more linguistic input, these chunks of phonemes become larger and larger. Phonological working memory is represented by a fixed amount of chunks that can be stored. Hence, early on in the model's learning, EPAM-VOC is able to store only a limited amount of linguistic information in working memory since the chunks at this point in time will not be large sequences of phonological information. Later on in learning, the phoneme sequences within chunks will be relatively large, and so an increased amount of information can be stored in working memory even when the number of chunks remain the same. This explanation of vocabulary learning puts forward the idea that improved performance with age arises due to an increased amount of linguistic knowledge. However, the model also explains wordlikeness and frequency effects quite easily. Phoneme sequences that appear regularly in the child's language will be represented within the model as relatively large chunks, whereas low frequency sequences will not. Therefore non-words that contain high frequency sequences can be stored in working memory using few chunks, giving rise to an increase in the likelihood of their correct repetition over non-words containing low frequency sequences. A similar explanation can be used for wordlike versus non-wordlike non-words. The former, since they bear great resemblance to words, are likely to be represented within the model using fewer chunks than non-wordlike non-words.

The current paper

EPAM-VOC has thus far been used to successfully simulate the non-word results of Gathercole and Baddeley (1989) plus a non-word set devised for children between 2 and 5 year of age. However, neither of these tests were specifically designed to vary in their lexicality. Since research has shown that non-word repetition is strongly influenced by the lexical nature of the non-words involved, this paper examines EPAM-VOC's repetition performance across three sets of non-words that vary in the degree of their lexicality. The remainder of this paper is as follows. First, the model itself is detailed so that the reader has more extensive knowledge of its workings. Second, a new study of 5-6 year-old children's repetition is described that presents three different non-word repetition tests that vary in the extent of their lexicality. Third, the results of the children are compared to the results of the model. Finally, a discussion of the results are given.

EPAM-VOC: A model of vocabulary learning

EPAM-VOC is a model of vocabulary learning that is based on the EPAM modelling architecture (Feigenbaum & Simon, 1984). This architecture has been used to successfully simulate human behaviour in a range of psychological domains (see Gobet et al., 2001). Furthermore, the modelling environment has been successfully applied to syntax acquisition as well as vocabulary acquisition (e.g. Freudenthal, Pine & Gobet, 2006; Freudenthal, Pine, Aguado-Orea & Gobet, 2007). The model presented here is an updated version of that described by Jones, Gobet and Pine (2007, 2008), since that model did not have an explicit articulation process.

Representing long-term knowledge

Knowledge within EPAM-VOC is represented as a hierarchy of chunks that contain phoneme sequences. Chunks that are lower down in the hierarchy contain larger sequences, and hence EPAM-VOC can be equated to a tree-like structure. The model begins with knowledge of all of the constituent phonemes in English, since there is good reason to believe that even at an early age, children have knowledge of the phonemes of their language (Bailey & Plunkett, 2002).

An example hierarchy of chunks is given in Figure 1. Here it can be seen that the model knows the phoneme sequence for the word "Toys" (T OY Z). Note that we represent phonemes using the phoneme set in the CMU Lexicon Database (available at www.speech.cs.cmu.edu/cgi-bin/cmudict) rather than the International Phonetic Alphabet. This is chiefly because the database allows the semi-automatic conversion of spoken utterances into their phonemic equivalent (this will be detailed later when the input regime for the model is covered).

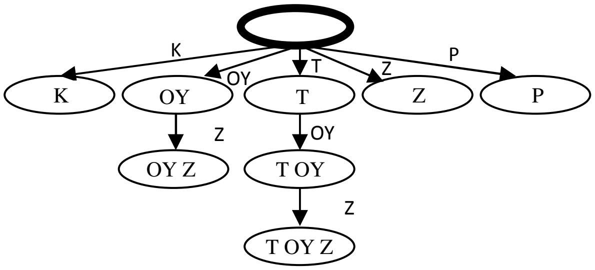


Figure 1: Graphical representation of EPAM-VOC having been presented with "Toys" (T OY Z) twice as input. Chunks are represented by ellipses and links are represented by arrows. Note that although only five phonemes are represented as single phoneme chunks (K, OY, T, Z and P) the model knows all phonemes in English as individual chunks.

Representing phonological working memory

Working memory in this version of EPAM-VOC is no longer limited to a set amount of chunks. Instead, there is a set amount of activation that is spread over the chunks that are in working memory (c.f. Cowan, 1997). However, we base this activation on time, since there is solid research to indicate that items in working memory have a temporal duration of 2,000 ms unless rehearsed (e.g. Baddeley, Thomson & Buchanan, 1975) and there is solid research that places timing estimates on accessing a chunk and accessing its constituent phonemes (Zhang & Simon, 1985).

For any given input, EPAM-VOC's long-term knowledge is accessed in order to convert the input into a series of chunks (i.e. representing the input sequence in as few chunks as possible). Each chunk is then assigned an access time – 400 ms to access the chunk and a further 30 ms to access each constituent phoneme bar the first (e.g. given the long-term knowledge of Figure 1, "Toys" would be allocated 460 ms, whereas the single phoneme "K" would be allocated 400 ms). Once the input has been represented in as few chunks as possible, and each chunk has been assigned an access time, then a pointer to each chunk is placed in working memory. The total access time is calculated by summing the access times for all chunks. When this total exceeds 2,000 ms, then there is a probability of less than 1.0 that a chunk can be correctly accessed from its pointer (when learning or articulating, the model requires the chunk to be accessed from its pointer in working memory).

Let us take the input "My toys are here" as an example (phonemic representation: "M AY T OY Z AE R H IY R") and the knowledge in Figure 1. Only "T OY Z" exists as a multi-phoneme chunk, and this is assigned an access time of 460 ms. All other phonemes ("M", "AY", "AE", etc.) are assigned an access time of 400 ms – there are a total of 8 chunks required to represent the input, in a total access time of (7*400 ms)+(1*460 ms)=2,560 ms. The probability of subsequently accessing a chunk from its pointer is the temporal duration of working memory divided by the total time required to access all of the chunks: 2,000/2,560=.78125.

To summarise, any given input is converted into as few chunks as possible using EPAM-VOC's long-term knowledge of phoneme sequences. This matching process takes a certain amount of time, and the result of the process is that a pointer to each chunk is placed in working memory. Since working memory only contains pointers to chunks, any process that requires the actual information in the chunk (e.g. when learning or articulating items in working memory) must access the chunk itself. The accurate access of information in a chunk is probabilistic, dependent upon whether the total access time for all chunks exceeds the 2,000 ms time limit of working memory.

Learning phoneme sequences

During learning, any given input is coded into as few chunks as possible, and pointers to the chunks are placed in working memory (as described above). The learning process then examines each pair of pointers to see if a phoneme sequence can be learnt that combines the information within each chunk pairing. This can only be done if each chunk is correctly accessed, but if this is the case, a new chunk is learnt whose contents are the combination of both chunks. Let us use the input "Toys" ("T OY Z") as an example. When EPAM-VOC first begins its learning, it only knows single phonemes as chunks, and therefore "T OY Z" would be represented in working memory using three pointers to three chunks (one pointer to each of "T", "OY" and "Z"). Since the time to encode the three chunks is 400 ms for each (totalling 1,200 ms and therefore within the 2,000 ms limit) then the subsequent accessing of the information within the chunks will be completely accurate. EPAM-VOC takes each pair of pointers in turn and tries to learn something from them. The first pair are "T" and "OY". The "T" chunk is accessed, and then a link to a new chunk is placed below the "T" chunk. The link will specify the additional information that is being learnt ("OY") and the new chunk contains the joint set of information ("T OY"). The next pair of chunks ("OY" and "Z") are then examined, and in a similar vein, a new chunk "OY Z" is learnt. Should "T OY Z" be presented to EPAM-VOC a second time, it can now be represented as two pointers to the chunks "T OY" and "Z". The learning from this pair of pointers would result in a new chunk "T OY Z" being added below the "T OY" chunk, and the resulting network would be that shown in Figure 1.

Let us now see how learning progresses when the access time exceeds the 2,000 ms limit. Take the previous example sequence "My toys are here" ("M AY T OY Z AE R H IY R") and the long-term knowledge of Figure 1. It was already determined that there was a .78125 probability of accessing a chunk that related to a pointer for this input. Since the pointers in working memory are analysed in a pairwise fashion, then if one pointer cannot access its associated chunk, no learning can be accomplished for that pointer. For example, if the pointer to the chunk "AY" failed, then EPAM-VOC could not learn the sequence "M AY" or the sequence "AY T OY Z".

Articulating phoneme sequences

In order for a phoneme sequence to be articulated, it must first be represented in working memory as a series of pointers to chunks (as described above). In the same way as for learning, each chunk needs to be correctly accessed from its pointer, otherwise an incorrect articulation takes place. However, even if each chunk is correctly accessed, the chunk may still be incorrectly articulated based on its frequency. EPAM-VOC maintains a frequency for each chunk based on the number of times that the chunk has been accessed. We assume that the articulation of phonemes in a chunk is based on the frequency of that chunk – those chunks that are low in frequency will attract more errors than chunks that are high frequency. Correct articulation of an input sequence (e.g. a non-word) is only achieved when all of the relevant chunks are correctly encoded into phonological working memory, and all of the phonemes are correctly articulated from each chunk based on the frequency of the chunk.

Training the model

The model uses naturalistic speech input based on the maternal input from 12 sets of mother-child interactions to 2-3 year-old children (Theakston, Lieven, Pine & Rowland, 2001). All input is converted into the phonetic alphabet of the CMU Lexicon Database, as discussed previously. 12 simulations are carried out, one for each set of mother's input. However, since comparisons are going to be made to 5-6 year-old children, additional input was sought from paternal interactions with 5 year-old children plus input from reading material for children of this age group (e.g. Snow White).

During training, the model was presented with the same amount of input as per the original maternal input. However, as learning progressed, more and more of the maternal input was replaced with input that reflected that which a 5-6 year-old child would receive.

Since the input to the model can vary based on which utterances from the mother were chosen for replacement, and which input from the 5-6 year-old input set was chosen as the replacement, then the model was run ten times for each "mother". This ensures that the results from the model are replicable and are not simply based on an advantageous input set. Similarly, the non-word repetition tests are carried out ten times for each simulation, since there are probabilistic elements to both encoding and articulation. There were therefore, for any non-word in a non-word repetition test: 12 mothers * 10 simulations runs * 10 attempts at each non-word = 1,200 repetitions of each non-word.

5-6 year-old children's repetition performance

Participants

25 5-6 year-old children (5;4-6;8, M=6;1; 10 male, 15 female) who all scored within normal ranges on the British Picture Vocabulary Scale (Dunn, Dunn, Whetton & Burley, 1997). All children were English monolinguals and had no hearing difficulties, as reported by their school teacher.

Materials

The CNRep (Gathercole, Willis, Baddeley & Emslie, 1994) that includes non-words that are considered high in lexicality since they include syllables that are either real words (e.g. **thick**ery) or morphemes (e.g. glister**ing**). The non-words in this test are either single consonant (e.g. sladding) or clustered consonant (e.g. glistow). There were 5 non-words of each type at each of three lengths (2-4 syllables). 5-syllable non-words were excluded because children at this age had difficulty in repeating non-words of this length.

The non-word test of Dollaghan, Biber and Campbell (1995) that contain 3-syllable non-words. 6 non-words contained a real-word syllable (e.g. **bath**esis) and 6 changed one phoneme in the non-word so that it was entirely non-lexical (e.g. fathesis). This test is considered medium in lexicality.

A new non-word repetition test that was entirely non-lexical and that contained two sets of 3-syllable non-words: 8 that were low frequency (e.g. latmunoz) and 8 that were very low frequency (e.g. wegnerterk). The non-words in each set were matched for syllable structure, number of phonemes, age of acquisition of the phonemes, and number of consonant clusters. The average log frequency was lower for very low frequency non-words than low frequency ones (.51 vs. .44, *t*(7)=3.92, *p*<.01) using a procedure for measuring bi-phone frequency similar to that of Luce and colleagues (e.g. Jusczyk, Luce & Charles-Luce, 1994; Vitevich, Luce, Charles-Luce & Kemmerer, 1997). This test was considered to be low in lexciality.

Design

The CNRep had two independent variables: non-word type (single or clustered) and non-word length (2, 3, or 4 syllables). The Dollaghan non-words had one independent variable (lexicality: lexical or non-lexical). The new non-word test also had one independent variable (frequency: low or very low). The dependent variable in all cases was repetition accuracy.

Procedure

All children were tested individually on a one-to-one basis in a quiet area of their school. Each non-word repetition test was carried out on a separate day. For the CNRep, the original recordings were maintained, but for the other two repetition tests the non-words were recorded by a speaker native to Nottingham. The instructions for each set of non-words are given below, and were the same for each non-word test. Children's responses were recorded onto a Sony ICD-MX20 digital voice dictaphone for later analysis.

"Hello, in a few seconds you will hear a funny made up word. Please say the word aloud yourself as soon as you hear it."

Results

For each repetition test, two sets of results are shown for the model: the average of all of the 1,200 simulations, plus the average of 12 simulation runs (one from each mother). The 12 runs are included since statistical analyses are based on these – the nature of the 1,200 simulations means that they show little variance, since they are all based on a similar set of input data. The selection of the single simulation on which to base statistical analyses was pseudo-random – that is, the individual 1,200 simulation runs were narrowed down to those that approximated the average of all 1,200 runs when taken as a whole, and one run (the first run for each mother together with the seventh of the ten duplicate repetitions) was randomly chosen from that set.

Figure 2 shows the children's results for the CNRep together with the results from EPAM-VOC. A 2 (non-word-type: single or clustered) x 3 (non-word-length: 2, 3, or 4 syllables) repeated measures ANOVA was performed on the children's data. There was a significant effect of non-word-type (*F*(1,24)=43.5, *p*<.001), showing that performance was better for single consonant non-words, and a significant effect of non-word-length (*F*(2,24)=26.7, *p*<.001), showing that performance was better for short non-words. There was no interaction between the two (*F*(2,48)=1.8, *p*>.05). For the model, there was also the same effect of non-word-type (*F*(1,11)=5.5, *p*<.05) and non-word length (*F*(2,22)=9.8, *p*<.001), with no interaction between the two (*F*(2,22)=.3, *p*>.05).

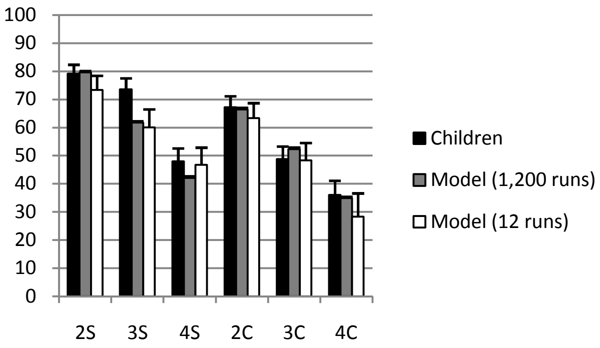


Figure 2: Non-word repetition performance (%) for the CNRep and for the two sets of model runs. The numeric on the x-axis denotes the number of syllables (2, 3, or 4) and the alphabetic character denotes the non-word type (S=single consonant, C=clustered consonant).

Figure 3 shows the children's and model's performance for the Dollaghan non-words and the new set of non-words. For the Dollaghan non-words, the children showed no difference in their ability to repeat lexical and non-lexical non-words (*t*(24)=.6, *p*>.05). The same was found in the model (*t*(11)=.6, *p*>.05). For the new set of non-words, there was no difference in children's repetition accuracy between low and very low frequency non-words (*t*(24)=.1, *p*>.05). Again, the same result was found in the model (*t*(11)=.5, *p*>.05).

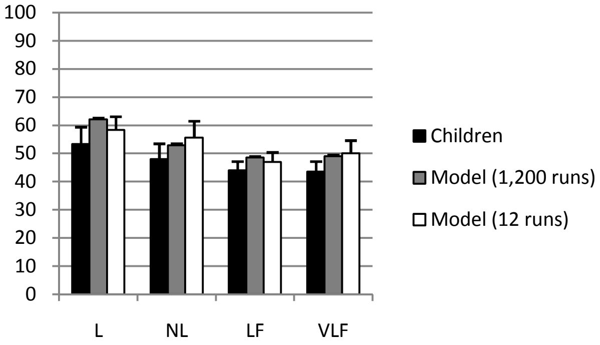


Figure 3: Non-word repetition performance (%) for the Dollaghan and new non-words, for the children and the two sets of model runs. L=Dollaghan, Lexical non-words; NL=Dollaghan, Non-Lexical non-words; LF=Low Frequency new non-words and VLF=Very Low Frequency new non-words.

Discussion

Figures 2 and 3 show that the model provides a very close fit to the repetition data of 5-6 year-old children. The central finding is that the statistical analysis of the model's data mirrors that of the children: clear effects are found for the non-words of the CNRep whereas no effects are found for the other repetition tests. The results from each set of non-words will now be discussed in turn.

The CNRep results are exactly those found in 4 and 5 year old children (e.g. Gathercole & Baddeley, 1989): performance improves for single consonant non-words and for non-words of fewer syllables. In fact, this set of findings is rather robust since they persist in older age groups also (e.g. Briscoe, Bishop & Norbury, 2001). Both the previous version of EPAM-VOC and the new version presented here are able to simulate these findings, suggesting that a reasonable account of working memory and its interaction with long-term linguistic knowledge is sufficient to capture the behaviour shown in the CNRep.

For the Dollaghan non-words, the original study showed an effect of lexicality for 10 year-old children (Dollaghan, Biber & Campbell, 1995). Not only do the children in this study not show this lexicality effect, the model itself also does not capture it. The model puts forward an explanation for the lack of effect, in that the lexical items (e.g. bath) are not robust enough in terms of their frequency of use to cause improved performance for non-words containing a lexical item. It would be interesting to take the learning in the model a stage further to the type of input older children may receive to then see if lexical effects emerge.

For the new set of non-words, there was no effect of frequency in either the children or the model. This shows that frequency effects are not picked up by children of this age, although they may well be for older children.

In summary, EPAM-VOC replicates the findings of 5-6 year-old children on three different non-word repetition tests varying in the degree of their lexicality. It now needs to be seen whether the errors made in children's repetitions are also mirrored by the model – if this is the case, then EPAM-VOC may prove to be a very strong explanation of the way in which children are learning vocabulary.

Acknowledgments

The author would like to thank Sarah Watson for her help in the data collection phase of this study, Marco Tamburelli for his help in the construction of the new set of non-words outlined, Gabrielle Le Geyt for the spoken versions of the non-words, and the schools and children who participated in the non-word repetition study. This research was funded by The Leverhulme Trust under grant reference F/01 374/G.

References

Baddeley, A. D., Thomson, N., & Buchanan, M. (1975). Word length and the structure of short-term memory. *Journal of Verbal Learning and Verbal Behaviour, 14*, 575-589.

Bailey, T. M., & Plunkett, K. (2002). Phonological specificity in early words. *Cognitive Development, 17*, 1265-1282.

Briscoe, J., Bishop, D. V. M., & Norbury, C. F. (2001). Phonological processing, language, and literacy: A comparison of children with mild-to-moderate sensorineural hearing loss and those with specific language impairment. *Journal of Child Psychology and Psychiatry, 42*, 329-340.

Cowan, N. (1997). The development of working memory. In N. Cowan & C. Hulme (Eds.), *The development of memory in childhood* (pp. 163-199). Hove, UK: Psychology Press.

Dollaghan, C. A., Biber, M. E., & Campbell, T. F. (1995). Lexical influences on non-word repetition. *Applied Psycholinguistics, 16*, 211-222.

Dunn, L. M., Dunn, L. M., Whetton, C., & Burley, J. (1997). *The British Picture Vocabulary Scale (2nd ed.).* Windsor, UK:NFER-Nelson.

Edwards, J., Beckman, M. E., & Munson, B. (2004). The interaction between vocabulary size and phonotactic probability effects on children's production accuracy and fluency in non-word repetition. *Journal of Speech, Language, and Hearing Research, 47*, 421-436.

Feigenbaum, E. A. & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science, 8,* 305-336.

Freudenthal, D., Pine, J.M., Aguado-Orea, J. & Gobet, F. (2007). Modelling the developmental patterning of finiteness marking in English, Dutch, German and Spanish using MOSAIC. *Cognitive Science, 31*, 311-341.

Freudenthal, D., Pine, J.M. & Gobet, F. (2006). Modelling the development of children's use of Optional Infinitives in English and Dutch using MOSAIC. *Cognitive Science*, *30*, 277-310.

Gathercole, S. E. (2006). Non-word repetition and word learning: The nature of the relationship. *Applied Psycholinguistics, 27*, 513-543.

Gathercole, S. E. (1995). Is non-word repetition a test of phonological memory or long-term knowledge? It all depends on the non-words. *Memory & Cognition, 23*, 83-94.

Gathercole, S. E., Willis, C. S., Baddeley, A. D., & Emslie, H. (1994). The children’s test of non-word repetition: A test of phonological working memory. *Memory, 2*, 103-127.

Gathercole, S. E., & Baddeley, A. D. (1989). Evaluation of the role of phonological STM in the development of vocabulary in children: A longitudinal study. *Journal of Memory and Language, 28*, 200-213.

Gobet, F., Lane, P. C. R., Croker, S., Cheng, P. C-H., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *TRENDS in Cognitive Sciences, 5*, 236-243.

Jones, G., Gobet, F., & Pine, J. M. (2008). Computer simulations of developmental change: The contributions of working memory capacity and long-term knowledge. *Cognitive Science*.

Jones, G., Gobet, F., & Pine, J. M. (2007). Linking working memory and long-term memory: A computational model of the learning of novel sound patterns. *Developmental Science, 10*, 853-873.

Jusczyk, P. W., Luce, P. A., & Charles-Luce, J. (1994). Infants’ sensitivity to phonotactic patterns in the native language. *Journal of Memory and Language, 33*, 630-645.

Theakston, A. L., Lieven, E. V. M., Pine, J. M., & Rowland, C. F. (2001). The role of performance limitations in the acquisition of verb-argument structure: An alternative account. *Journal of Child Language, 28*, 127-152.

Vitevitch, M. S., Luce, P. A., Charles-Luce, J., & Kemmerer, D. (1997). Phonotactics and syllable stress: Implications for the processing of spoken nonsense words. *Language and Speech, 40*, 47-62.

Zhang, G. & Simon, H. A. (1985). STM capacity for Chinese words and idioms: Chunking and acoustical loop hypotheses. *Memory & Cognition, 13*, 193-201.

The Persistent Visual Store as the Locus of Fixation Memory in Visual Search Tasks

David Kieras (kieras@umich.edu)
Electrical Engineering & Computer Science Department, University of Michigan
2260 Hayward Street, Ann Arbor, Michigan 48109 USA

Abstract

Experiments on visual search have demonstrated the existence of a relatively large and reliable memory for which objects have been fixated; an indication of this memory is that revisits (fixations on previously fixated objects) typically comprise only about 5% of fixations. Any cognitive architecture that supports visual search must account for where such memory resides in the system and how it can be used to guide eye movements in visual search. This paper presents a simple solution for the EPIC architecture that is consistent with the overall requirements for modeling visually-intensive tasks and other visual memory phenomena.

Keywords: visual search; cognitive modeling; eye movements.

Introduction

Many everyday and work activities involve visual search, the process of visually scanning or inspecting the environment to locate an object of interest that will then be the target of further activity. Many human-computer interaction tasks require such visual search to be made in a visual environment that is must simpler than natural scenes. For example, a particular icon coded by color, shape, and other attributes must be located on a screen and then clicked on using a mouse. This domain combines relative simplicity of the visual characteristics of the searched-for objects with practical relevance: the task is a natural one in the sense that such activities are very common in current technology. Visual search is so heavily relied on in many computer-based systems that it probably is a major bottleneck in system performance. Thus understanding in detail how visual search works in such domains can lead to better system designs. In addition, if visual search can be understood in the context of a comprehensive computational cognitive architecture, then it will add to our knowledge of human perception, cognition, and action in the especially rigorous and coherent way characteristic of computational cognitive architectural modeling.

Visual Search and Active Vision

In a laboratory visual search task, a display of objects is presented, and the participant must locate a particular object specified by perceptual properties and make a response based on whether such an object is present or exactly which properties it has (e.g. the specific shape). In most experiments, the display is static and contains some number of objects, only one of which is the target that must be responded to; the others are distractors. The properties of the display or the displayed objects are manipulated, and reaction time (RT) and/or eye movements are measured. The empirical literature on this task was dominated for a long time by studies that measured only RT, and often for tachistoscopically presented displays that ruled out eye movements, but more recently the cost of eye movement data collection has decreased to the point that it has become much more common, and deservedly so. While any behavioral measurement only indirectly reflects the mental processes that produce it, RT is clearly much less diagnostic of what goes on during visual search than eye movements. Furthermore, tasks in which the eye is free to move about a static display in a naturalistic manner, typical of eye movement studies of visual search, will be more representative of the normal operation of the visual system and the role of attention in visual activity. This point was argued eloquently by Findlay and Gilchrist (2003) in presenting an *active vision* framework for understanding visual activity; it is markedly different from traditional approaches to visual attention which have ignored both the role of eye movements and extra-foveal information.

A key process in visual search is choosing the next object for inspection. A variety of studies (see Findlay & Gilchrist, 2003, for a review) have shown that this choice is not at all random; rather the color, shape, size, orientation, or other visual properties of objects influences which object is chosen for the next fixation; the phenomenon is called *visual guidance* or *eye guidance*. In the active vision framework, these properties are available in extra-foveal or peripheral vision to some extent, meaning that visual attention, which in the context of normal visual activity is almost synonymous with where the eye is fixated, is a process of selecting for detailed examination one of a large number of objects currently perceived to be in the visual scene, and doing this selection on the basis of the visual properties available in extra-foveal vision.

Fixation Memory

An important fact about visual guidance in visual search tasks is that an object that was previously fixated will be only rarely selected for a new fixation. This is an old result in eye movement studies (e.g. Barbur, Forsyth, & Wooding, 1993), but it did not receive much attention until the controversial Horowitz and Wolfe (1998) claim that "Visual search has no memory." They compared search RTs of a static display with a changing display, in which the objects changed positions during search, and found no difference in RT. If the visual search mechanism remembered where it had already inspected, it should be disrupted if the objects changed location; the RT being unaffected argues that the search was not disrupted, which means in turn that there was no memory for the previous fixations. Peterson, Kramer, Ranxiao, Irwin, and McCarley (2001) countered with a study demonstrating that "Visual search has memory". They recorded eye movements during search of a static display, and discovered, as earlier studies had noted, that revisits were rare, meaning that the previous fixations were remembered in some way.

Encoding failures trigger revisits. Peterson et al. went further with a detailed analysis showing that most revisits were made immediately after only one intervening fixation, which rules out memory failure as the cause of a revisit. Rather, Peterson et al. proposed that revisits were due to *encoding failures*: soon after fixating an object and moving on to the next, the person would realize that the previous object had not been fully encoded, and so would revisit it. Using a Monte-Carlo model, they demonstrated that this explanation accounted for the statistical structure of the revisits considerably better than either a no-memory or memory-failure model.

Search strategies dominate. Several subsequent studies (e.g. von Mühlenen, Müller & Müller, 2003; Geyer, von Mühlenen, & Müller, 2006) using RT, eye tracking, and changing displays make it clear that the Horowitz and Wolfe results were an artifact of how the changing displays would force a change in task strategy. If the display is changing, the only way to perform the task successfully is use a strategy that compensates, such as to "wait and see" whether the target appears in a subset of the display. In other words, the changing display paradigm forces a strategy that produces a no-memory effect. Regardless of the methodological issues and the merits of the results, an important implication is that making use of memory for previous fixations is not "hard-wired" in the visual system, e.g. at the oculomotor level, but rather is an optional feature of a visual search task strategy.

Objects, not locations. Additional studies (e.g. Beck, Peterson, & Vomela, 2006) have attempted to determine whether what is remembered on each fixation is the location, the identity, or the properties of the objects. However, it should be clear that in a changing-display paradigm, if objects are identified in terms of their properties (e.g. shape), then they are "teleporting" from one location to the next, which is not a natural input to the visual system. Hulleman (2009) performed the most elegant and naturalistic test of whether fixation location was remembered simply by having the objects move around on the display during search similar to the Pylyshyn & Storm (1988) multiple object tracking paradigm. He observed almost no difference in search rates compared to a static display. This strongly suggests that fixation locations themselves were not remembered, since the objects were continuously changing location. The conclusion would seem to be that previously fixated *objects* are being remembered, where object identity persists over changes in location. In a naturally static display, such as the Peterson et al. (2001) paradigm, the issue does not arise: objects retain their location and properties.

Large capacity. The consensus of the empirical literature at this point is that memory for previous fixations exists. Moreover, it has a fairly large effective capacity. The Peterson et al. study involved twelve objects, half of which would have to be visited on the average. Results described in Kieras and Marshall (2006) involved 48 objects for two targets, with low revisit rates. Takeda (2004) estimated the capacity as high as 20 objects. This effective capacity is much more than the typical estimates for working memory, and so-called visual working memory in particular (e.g. Luck & Vogel, 1997) which has been estimated as holding only about four objects in a change-detection paradigm.

The locus puzzle. From the point of view of cognitive architecture, this result presents a serious quandary. Where is this capacious and reliable memory situated, and how does it work? Is it a special-purpose memory, or is it simply a by-product of some other memory function? These questions were addressed as part of program of detailed quantitative modeling of visual search tasks using the EPIC architecture, which was developed to represent perceptual-motor constraints on performance as fully as cognitive constraints, and so is well-suited to the goal. This work with EPIC visual search models focussed on representing how multiple stimulus attributes could guide visual search through conjunctive feature guidance, and how to represent their differential availability at the retinal level. These models were successful at accounting for detailed results in very simple tasks such as Findlay's (1997) first-saccade conjunctive search, searching very large displays of 100 multiattribute objects as in Williams (1967), and searching dense displays of 48 complex objects (Kieras & Marshall, 2006). However, in these models, the memory for fixations was represented in an unsatisfactory *ad hoc* manner. This paper presents a detailed model for the Peterson results to show how the fixation memory could be a side function of a memory system that is already present.

The EPIC Cognitive Architecture

The EPIC architecture for human cognition and performance provides a general framework for simulating a human interacting with an environment to accomplish a task. Due to lack of space, the reader is referred to Kieras & Meyer (1997), Meyer & Kieras (1997), or Kieras (2004) for a more complete description of EPIC. Figure 1 provides an overview of the architecture, showing perceptual and motor processor peripherals surrounding a cognitive processor; all of the processors run in parallel with each other. To model human performance of a task, the cognitive processor is programmed with production rules that implement a strategy for performing the task. When the simulation is run, the architecture generates the specific sequence of perceptual, cognitive, and motor events required to perform the task, within the constraints determined by the architecture and the task environment.

Figure 2 expands the visual processor shown in Figure 1. The *eye processor* explicitly represents differential retinal availability in terms of acuity functions that specify which visual properties of objects are currently visible as a function of the current position of the eye and the size of the object. The currently available visual properties for each object are represented in the *sensory store*; the *perceptual processor* then encodes the properties of each object, possibly in relation to other objects, and passes the encoded representation on to the *perceptual store* where they are available to the cognitive processor to match the conditions of production rules. The perceptual store thus contains the current representation of the visual world that cognition can reason and make decisions about, especially decisions about

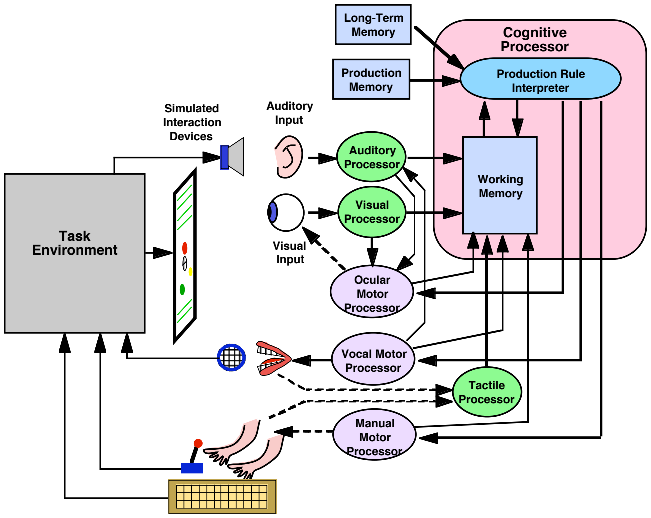


Figure 1. The overall structure of the EPIC architecture.

where to move the eyes next by commanding the *ocular motor processor*. The perceptual store retains the representations for *all objects currently visible*, with more information and detail about those that have been fixated.

When the eyes move away from an object, the properties of the object persist for a short time (e.g. 200 ms) in the sensory store, and when lost, the perceptual processor notes that the corresponding property in the perceptual store no longer has sensory support. After a relatively long time, the property will then be lost from the perceptual store. But if the object disappears completely, it and all of its properties will be removed from the perceptual store fairly quickly.

The concept is that as the eyes move around the visual scene, a complete and continuous representation of the objects currently present in the visual situation will be built up and maintained in the perceptual store, allowing the cognitive processor to make decisions based on far more than the properties of the currently fixated object. The notion that this information persists for a considerable time as long as the scene is present in supported by studies summarized by Henderson & Castelhano (2005) in which a visual scene is continuously present, but using a gaze-contingent forced-choice paradigm, subjects are tested for their memory of a previously fixated object in a naturalistic scene; retention times at least several seconds long were observed.

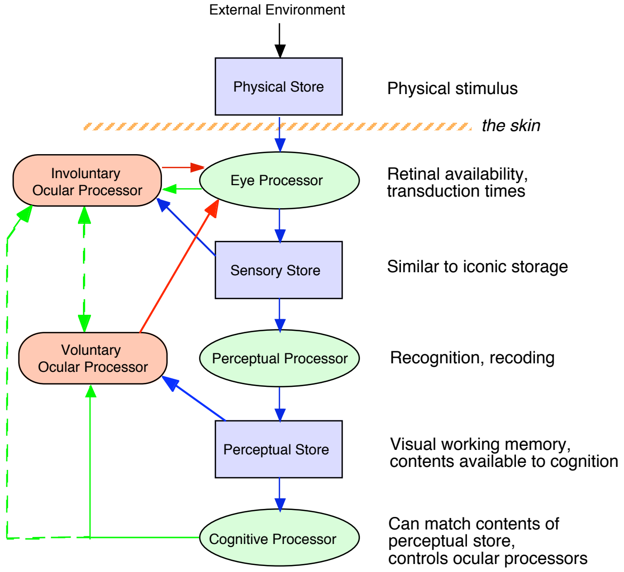

External Environment
Physical Store
Physical stimulus
the skin
Involuntary Ocular Processor
Eye Processor
Retinal availability, transduction times
Sensory Store
Similar to iconic storage
Voluntary Ocular Processor
Perceptual Processor
Recognition, recoding
Perceptual Store
Visual working memory, contents available to cognition
Cognitive Processor
Can match contents of perceptual store, controls ocular processors

Figure 2. EPIC's visual system.

Modeling Fixation Memory

The earliest attempts to fit models with the EPIC architecture for visual search in several tasks determined that some kind of fixation memory is required in order to account simultaneously for basic measures such as the number of fixations, search time, and distribution of fixations on objects with different properties (e.g. Kieras & Marshall, 2006). In order to include fixation memory, these earliest models simply "tagged" each object in memory to designate that it had already been fixated and then made an occasional random fixation to produce a revisit. This is an unsatisfactory ad-hoc solution.

The model presented here examines a more interesting possibility, namely that the perceptual store, which represents the current visual scene, could serve as a memory for fixations. That is, if the object has been fixated, then its representation would include the relevant property of the object; if the object was the target, the search would stop as soon as this was determined. But if it was not, then the next object to be examined can be chosen from the set of objects currently lacking information about the property in question. Thus by choosing objects whose properties are unknown, previously fixated objects will not be revisited.

However, since the encoding of the fixated objects is not perfectly reliable, there will be occasions when a previously fixated object will be lacking the target property, and so will get visited again. This concept is the basis for the simple statistical model presented by Peterson et al. (2001); the explicit cognitive architectural model presented here provides a generalization to other visual search tasks, and in addition, clarifies some aspects of their results.

Model for the Peterson Task

Figure 3 shows the EPIC model display of the physical visual situation consisting of the stimuli for a single trial in the Peterson task after several fixations. The stimuli on each trial were twelve objects presented in random locations on a static display; eleven were distractors, consisting of rotated L-shapes, and one was the target, a T-shape rotated either to the left or to the right. The participant's task was to locate the T shape and press a key depending on whether it was the left- or right-rotated shape. Figure 3 shows how the objects were quite small, being 0.19° in visual angle size, and were widely spaced, a minimum of 4.9° apart. Participants with normal vision would thus have to fixate each object

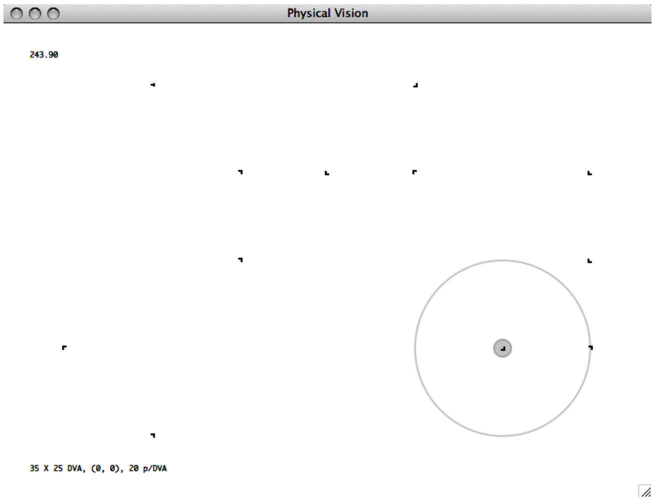


Figure 3. An example of the physical situation in a Peterson et al. (2001) task trial after several fixations as depicted in EPIC's display. The concentric circles show the current location of the eyes; the small inner circle has a 1° radius corresponding to the conventional fovea size; the outer circle is a calibration ring with 10° radius. The sizes of the overall display and the search objects are shown to scale, so the objects are indeed very small.

individually to recognize it. Because of space limitations, the very small shapes are obscured in the figure.

The EPIC model to fit the data comprised a choice of (1) visual acuity parameters, (2) an encoding process in the visual perceptual processor, (3) a parameter for the encoding failure rate, (4) a parameter for the *decay time* of visual properties in the perceptual store that are no longer sensorily supported, and (5) a set of production rules that implemented the visual search strategy. Each of these model inputs will be described briefly.

(1). The visual acuity parameters for this situation are very simple, specifying that the shape of an object was available only in the fovea, while the location of an object is available throughout the visual field, meaning that any object can be selected as a fixation target. The object color plays no role in the task, but its availability was left at the default value. Figure 4 shows the effects of the acuity functions for the same display as in Figure 3.

(2, 3).The perceptual processor encodes the objects by in terms of the recognized shapes for distractors and targets, which are then stored in the visual perceptual store where they become available for production rules to match on. The Peterson et al. encoding failure concept is represented as follows: with some constant probability, the encoding could fail and result in a partial encoding that retains some information about whether a distractor or target was present, but not enough to identify the actual shape. For example, a partial encoding for distractor could be that two line segments were joined at the ends, while a partial encoding for a target could be that one line segment joined another in the middle. For purposes of display in the model, these partial encodings are represented by partially rotated L and T shapes. The probability of partial encoding of targets and distractors is assumed to be the same.

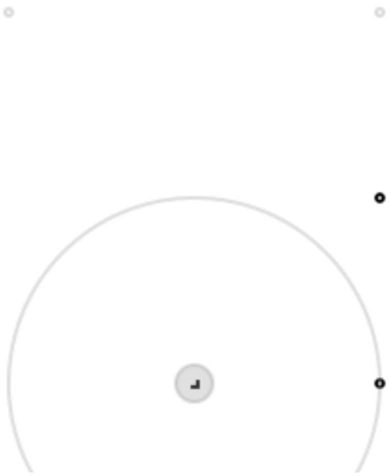


Figure 4. An example of the contents of the sensory store corresponding to the lower right corner of Figure 3. Objects whose location, but no other properties, are known are represented as light gray open circles (top two). Objects which are close enough to the current fixation point to have their black color available, but not their shape, are represented as black open circles (right hand two). Both the shape and the color are available for the currently fixated object.

(4). After encoding, if the eye is then moved to a different object, the actual shape quickly becomes unavailable, and the encoded shape is marked as no longer having sensory support. The encoded property then disappears from the perceptual store after the time specified by the decay time. In accordance with the Henderson and Castelhano (2005)

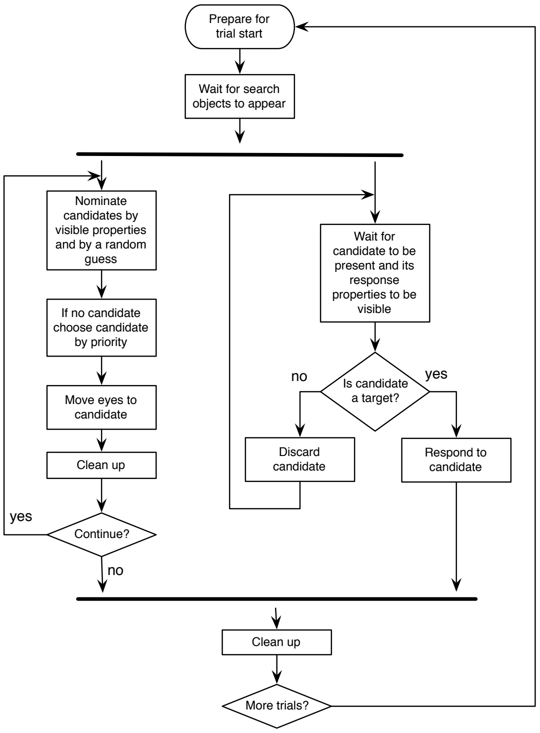

Prepare for trial start
Wait for search objects to appear
Nominate candidates by visible properties and by a random guess
If no candidate choose candidate by priority
Move eyes to candidate
Clean up
yes
Continue?
no
Wait for candidate to be present and its response properties to be visible
no
Is candidate a target?
yes
Discard candidate
Respond to candidate
Clean up
More trials?

Figure 5. Flowchart for the search task strategy. Two threads overlap the process of choosing the next candidate and checking whether the current candidate is a target.

results, this parameter is assumed to be a few seconds in magnitude, though for purposes of model fitting for this data, it was made as small as possible.

(5). The visual search strategy in the model is an application of a basic strategy, shown in Figure 5, that has been used in several EPIC visual search models. There are two threads of execution. Nomination rules in the first thread propose objects to fixate based on available visual properties, and also nominate a random choice. Choice rules then pick a single candidate from the nominated objects according to a priority scheme, and launch an eye movement to the chosen candidate. The rules in the second thread wait for all relevant properties of the fixated candidate to be fully visible and either respond if it is a target, or discard the candidate if not. The overlapped processing provided by the two threads enables the time between successive eye movement initiations to be short, about 250 ms, which is commonly observed in high-speed visual search tasks.

For the Peterson model, the strategy chooses objects for the next fixation according to the following simple scheme: Only objects not being currently inspected are considered. If an object is partially encoded as a target, it is given first priority for the next fixation, followed by an object not encoded as a distractor (either no encoding at all or partially encoded as a distractor), followed by an object chosen at random. Thus the strategy favors possible targets, then unvisited or partially encoded objects, and avoids objects fully known to be distractors. Figure 6 summarizes the model by showing the contents of the perceptual visual store corresponding to Figure 3, right before a target revisit.

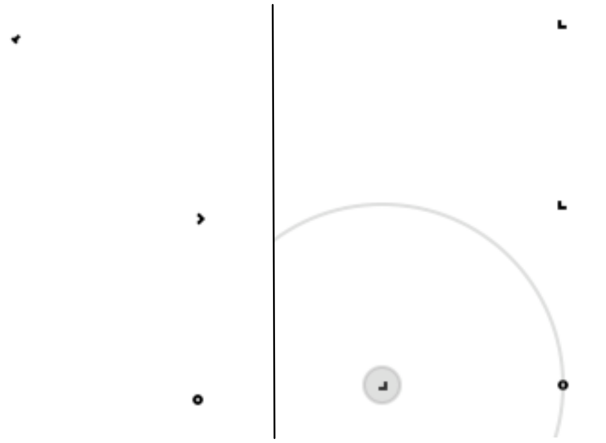


Figure 6. An example of the contents of the perceptual store after several fixations corresponding to the upper left corner (left panel) and lower right corner (right panel) of Figure 3. Two objects whose color is known to be black, but whose shape is unknown are represented as black open circles. Previously fixated objects have encoded shapes available. In the right panel, three distractors have been fixated, including the current one. In the left panel, there is a partially encoded target at the top left, and partially encoded distractor in the center right, represented as partially rotated shapes. The strategy is about to move the eyes back to the previously visited target.

Results

Figure 7 shows the observed and predicted results for this model, with the observed data from Peterson et al. (2001) shown as solid points and lines with 95% confidence intervals. The graph shows the proportion of fixations that are revisits as a function of *lag*, the number of fixations between the original and the revisit. Thus most of the revisits occur after fixating one intervening object. The total number of revisits is shown in the upper curve, and the number of revisits on targets in the lower curve.

The predicted values from the model are shown as open points and dotted lines. The model parameter values were chosen by iteration to produce a good fit with 10,000 simulation trials per run. The fit of the model predictions is very good; almost all of the predicted values are within the confidence intervals; the R^2 and standard error of prediction is 0.986 and 0.001 for Revisits, and 0.999 and 0.000 for Target Revisits. The parameter values producing this fit are 0.14 for the probability of encoding failure, and 4000 ms for the decay time of properties in the perceptual store. Any shorter decay time produces an increase in the number of predicted revisits at very long lags.

A comparison to the Peterson et al. 2001 model is useful. Although they reported the number of target revisits, they modeled only the total number of revisits, and so did not attempt to account for the fact that most of the immediate revisits are due to revisits to the target. Exploration with a variety of strategies and parameter values makes it clear that to fit both curves, the model must make the distinction between partially encoded targets and partially encoded distractors. Partially encoded targets must be favored for revisits, and partially encoded distractors treated similarly to unvisited objects — otherwise, there is no way to fit both curves simultaneously. That is, if possible targets are not favored for a revisit, then parameters that fit the overall rate of revisits far underpredict the proportion of target revisits.

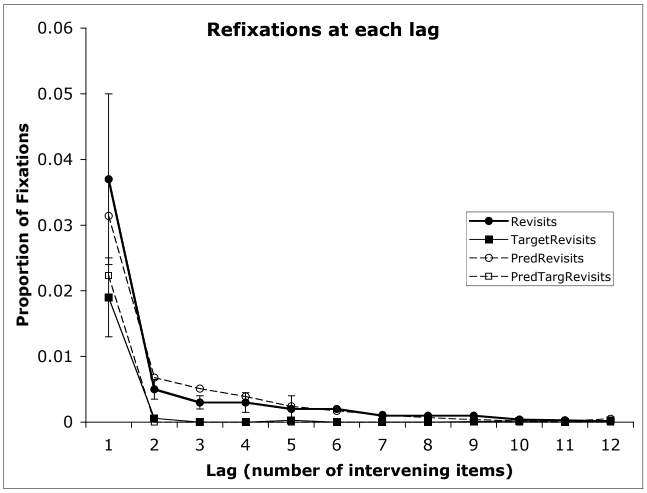


Figure 7. Solid points and lines and confidence intervals are the observed proportion of fixations at each lag for total Revisits and Target Revisits. Open points and dotted lines are model predictions.

In fact, according to the model, the revisit data for all objects are the sum of two underlying functions: Partially encoded distractors are revisited only because they are treated like unvisited distractors, yielding a shallow descent in revisits as a function of lag (imagine the total revisits curve for lags 2 to 12 extrapolated back to lag 1). But partially encoded targets are revisited immediately, producing the sharp descent from lag 1 to lag 2. The sum of these two trends produces the sharp-then-shallow curve for total revisits which was modeled by Peterson et al. The current EPIC model always revisits partially encoding targets immediately, and never favors partially encoded distractors over unvisited distractors. It might be possible to improve the fit slightly by using different encoding failure parameters for targets and distractors, and a more subtle choice strategy, but the current model fits the data acceptably well with few free parameters and a simple strategy.

Conclusion

The Peterson et al. (2001) experiment is fundamental in that it well isolates a set of basic processes underlying visual search that a successful cognitive architecture must be able to explain naturally. The present EPIC model demonstrates a how memory for fixations can emerge from the operation of a strategy for choosing the next object based on a persistent visual store of information about previously fixated objects. In this task, the only relevant properties of the objects is their location, whose wide availability makes it possible to choose an previously unvisited object for fixation, and the shape, visually available for only the one object foveated at a time. This model works by relying on the persistence of the perceptual encoding in the visual store and a simple strategy that maximizes task performance by making the most efficient use of partial encoding results.

The persistent visual store needs to be present in the architecture to allow cognition to reason about the entire visual situation. Its persistence is required for this architectural function, and is consistent with other empirical results such as those surveyed by Henderson and Castelhano (2005).

Thus the architectural puzzle posed by the existence of fixation memory can be solved by relying on this otherwise-required store; no special architectural mechanism is need to account for fixation memory. Models currently being refined for other visual search tasks (such as that described in Kieras & Marshall, 2006) show that this concept of fixation memory scales to more complex displays, objects, and search tasks.

Acknowledgment

This work was supported by the Office of Naval Research, under Grant No. N00014-06-1-0034.

References

Barbur, J.L., Forsyth, P.M., & Wooding. D.S. (1993). Eye movements and search performance. In D. Brogan, A. Gale, & K. Carr (Eds.) *Visual Search 2*. London: Taylor & Francis. 253-264.

Beck, M.R., Peterson, M.S., Vomela, M. (2006). Memory for where, but not what, is used during visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 3**2**, 235-250.

Findlay, J. (1997). Saccade target selection during visual search. *Vision Research*, **37**, 617–631.

Findlay, J.M., & Gilchrist, I.D. (2003). *Active Vision.* Oxford: Oxford University Press.

Geyer, T.,von Mühlenen, A, & Müller, H.J. (2006). What do eye movements reveal about the role of memory in visual search? *Quarterly Journal of Experimental Psychology*, 60, 924-935

Henderson, J.M. & Castelhano, M.S. (2005). Eye movements and visual memory for scenes. In G. Underwood (Ed.), *Cognitive processes in eye guidance*. New York: Oxford University Press. 213-235.

Horowitz, T.S. and Wolfe, J.M. (1998). Visual search has no memory. *Nature*, 394, 575–577.

Hulleman, J. (2009). No need for inhibitory tagging of locations in visual search. Psychonomic Bulletin & Review, 2009, 16 (1), 116-120.

Kieras, D.E. (2004). EPIC Architecture Principles of Operation. Web publication available at ftp://www.eecs.umich.edu/people/kieras/EPIC/EPICPrinOp.pdf

Kieras, D.E. (2007). The control of cognition. In W. Gray (Ed.), *Integrated models of cognitive systems*. (pp. 327 - 355). Oxford University Press.

Kieras, D.E, & Marshall, S.P. (2006). Visual Availability and Fixation Memory in Modeling Visual Search using the EPIC Architecture. *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, 423-428.

Kieras, D. & Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12, 391-438.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, 390, 279–281.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, **104**, 3-65.

von Mühlenen, A. Müller, H.J., & Müller, D. (2003). Sit-and-wait strategies in dynamic visual search. *Psychological Science*, **14**, 309-314.

Peterson, M.S., Kramer, A.F., Ranxiao, F.W., Irwin, D.E., & McCarley, J.S. (2001). Visual search has memory. *Psychological Science*, **12**, 287-292).

Pylyshyn, Z. W., & Storm, R. W. (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, **3**, 179-197.

Takeda, Y. (2004). Search for multiple targets: Evidence for memory-based control of attention. *Psychonomic Bulletin & Review*, **11**, 71–76.

Williams, L.G. (1967). The effects of target specification on objects fixated during visual search. In A.F. Sanders (Ed.) *Attention and Performance*, North-Holland. 355-360.

Why EPIC was Wrong about Motor Feature Programming

David Kieras (kieras@umich.edu)
Electrical Engineering & Computer Science Department, University of Michigan
2260 Hayward Street, Ann Arbor, Michigan 48109 USA

Abstract

The EPIC computational cognitive architecture was among the first to propose representing motor movement constraints explicitly in the form of motor processors that implemented a specified time course for the preparation, initiation, and production of movements. A key feature of this proposal was that movements were specified in terms of features, and movement preparation time was linear with the number of features that had to be prepared before a movement was initiated. While successful in modeling many high-speed tasks involving choice reaction times with keypress responses, serious difficulties appeared in modeling high-speed visual search tasks involving saccades and mouse movements. A reappraisal of the basis for EPIC's assumptions requires a critical change: *visually aimed manual and ocular movements require no feature preparation time*.

Keywords: cognitive architecture; motor processing; motor features; S-R compatibility; spatial compatibility

Introduction

The EPIC architecture for human cognition and performance provides a general framework for simulating a human interacting with an environment to accomplish a task. Due to lack of space, the reader is referred to Kieras & Meyer (1997), Meyer & Kieras (1997), or Kieras (2004) for a more complete description of EPIC. Figure 1 provides an overview of the architecture, showing perceptual and motor processor peripherals surrounding a cognitive processor; all of the processors run in parallel with each other. To model human performance of a task, the cognitive processor is programmed with production rules that implement a strategy for performing the task. When the simulation is run, the architecture generates the specific sequence of perceptual, cognitive, and motor events required to perform the task, within the constraints determined by the architecture and the task environment. Components of EPIC, especially the motor processors, have been incorporated into other cognitive architectures that use their own cognitive processor.

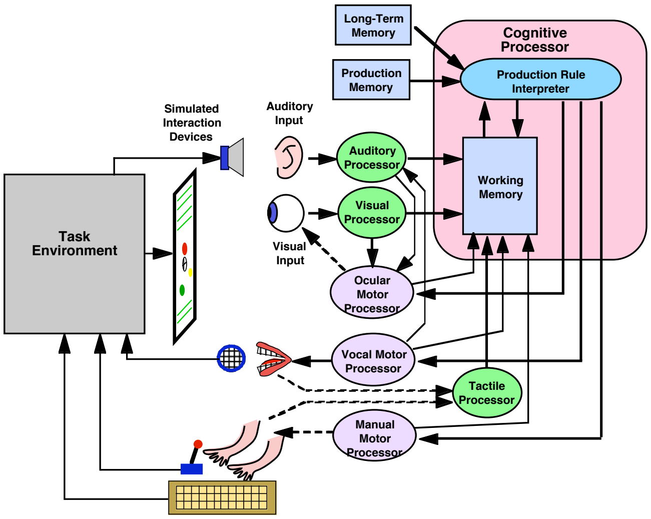


Figure 1. The overall structure of the EPIC architecture.

Motor Feature Preparation

Meyer and Kieras (1997) argued that a cognitive architecture must explicitly represent the constraints on motor activity in order to comprehensively account for task performance. They specified these constraints in the EPIC cognitive architecture in terms of motor processors that were equal in status to perceptual processors and the cognitive processor. These motor processors, one for each of the manual, ocular, and vocal motor modalities, accept symbolic movement commands from a production-rule cognitive processor, and then generate simulated movements that are inputs to a simulated task environment. Their characterization focussed on the temporal constraints, not on how muscle activity would be controlled, and can be summarized as follows:

1. Movements are described in terms of motor features, such as the direction and distance of a pointing movement, or the hand and finger used for a button-pushing movement. The type of movement, the *style*, was considered the dominant feature within each movement modality.
2. When a movement is commanded, the motor processor prepares each feature serially, requiring a constant time per feature, estimated as 50 ms. When all features have been prepared, the movement is initiated. After an initiation time delay (also estimated as 50 ms), the mechanical movement begins.
3. Once prepared, the features for a movement are retained in the motor processor. If a movement is repeated, its features do not have to prepared, and the movement can be initiated immediately.
4. The motor processor can be commanded to prepare one or more movement features in advance; these are stored in the motor processor. When the movement is commanded, the previously prepared features do not have to be prepared, allowing the movement to be initiated sooner by the amount saved in preparation time.
5. The feature preparation mechanism is used for the motor processors in all modalities; the only difference is in the specific feature structure of different movements possible in each modality.

Meyer and Kieras based the motor processor assumptions on the available literature on motor control (see Rosenbaum, 1991 for an overview). Because the motor control area is seriously under-researched (Rosenbaum, 2005), the only useful theoretical concept available was Rosenbaum's theory of motor feature programming, and so it was adopted. However, any cognitive architect has to go beyond the specific literature to some extent by simplifying and generalizing the empirical effects and available theory to produce a conceptually and practically manageable architecture. Uniformity of mechanism is a compelling first approximation as well. Meyer and Kieras therefore assumed that feature preparation held for all motor modalities and that the per-feature time was constant and uniform for all features and modalities. They also proposed the specific features for various movements and postulated dependencies between them. For example, manual pointing movement features could not be re-used in eye movement feature preparation; changing hands or fingers could reuse the remaining manual features, but changing the style of manual movement requires all features to be prepared.

Kieras and Meyer successfully constructed many EPIC models for high-speed choice reaction tasks, especially dual tasks, with these motor processors (summarized in Meyer & Kieras, 1999). Depending on the details of the task, motor feature preparation time often set a substantial constraint on other processes in the models, especially if features could be prepared in advance. However, since many experiments are done with the exact required response movement randomized over trials, the net effect of feature preparation usually is simply to produce an average preparation time that serves as a component in the overall latency of response. Also most of the modeled experiments involved button presses, typically using laboratory methodology in which the stimulus location is constant and the fingers are pre-positioned on the alternative response keys, meaning there is little or no eye or hand movement.

Since EPIC has been one of the few cognitive architectures that attempted to represent motor processes and constraints, even in highly abstracted form, its analysis of motor processing has been explicitly adopted in other architectures, in particular, the widely used ACT-R/PM and current ACT-R architectures (e.g. Anderson & Lebiere, 1998). Thus the status of EPIC's characterization of motor processing has broad relevance and concern to the cognitive architecture community as a whole. This paper presents why a major revision in this characterization is required: *visually aimed manual and ocular movements require no feature preparation time*.

Symptoms of the Problem

When models for high-speed visual search tasks were constructed, it proved to be extremely difficult to fit basic latency data given the constraints on ocular feature preparation. For example, models were constructed for Findlay's (1997) results for latency and accuracy in the first saccade in a conjunctive visual search task. Findlay observed that the latency of the first saccade was only about 250 ms, which was quite difficult to obtain in the EPIC architecture with its standard timing parameter values. Due to the syntax and semantics of the production rules, two production rule cycles are required to identify the target of the eye movement, for a total of 100 ms. A motor initiation requires 50 ms. An eye movement in the task required preparing an average of one feature, for an additional 50 ms. The total is 200 ms, which leaves only 50 ms total for stimulus transduction and recognition, which seems implausibly short — 100 ms seems a more reasonable perceptual processing time.

In a more complex visual search task (such as in Kieras & Marshall, 2006), there are enough eye and hand movements that feature preparation time can sum to several hundred ms in the total RT. A more complex task strategy will also require more production rule firings to choose the next fixation target, making it even more difficult to fit the commonly observed 250 ms delay between successive saccades, even if multithreaded production rule strategies are deployed.

While the difficulty of programming a model is not normally grounds for rejecting a model, it is significant if the difficulty is due to a cognitive architectural feature. A cognitive architecture is supposed to capture the underlying mechanisms and processes of human activity; it is natural to expect that simple activities should have reasonably simple representations in the architecture. So undue difficulty in constructing a model for a straightforward task is a strong suggestion that the architecture is incorrect.

In the case of the visual search task modeling, it was observed that setting the feature preparation time to zero for aimed manual and ocular movements gave the strategy programming adequate "breathing room" in fitting the data. This led to a re-examination of empirical literature behind this basic feature of EPIC's motor processors to see if the original reasons for the motor feature programming were still justifiable.

Reappraising the Literature

Manual movement feature preparation

The seminal experimental demonstration of motor feature preparation is Rosenbaum (1980) in which participants made button-press movements in response to precues and cues. The experimental task is diagrammed in Figure 2. On each trial, the participant received a precue which specified some of the putative features of the movement, then a cue, which specified the exact movement, whereupon the participant made the response movement. The latency of the initiation of the movement was recorded. More specifically, as shown in Figure 2, the response buttons were a set of eight buttons arranged in two parallel rows, one on the right, and one on the left, running forwards towards the display, and rearwards towards the participant. The two center buttons in each row were the "home" buttons; at the start of the trial, the participant held each home button down with their left- and right-hand index fingers, and in response to the cue, moved one of the fingers to the response button; the time of release of the home button is the RT. The buttons were color-coded; the participant was practiced in associating the color codes with the physical location of the buttons. The response cue was a colored disk appearing on the display that designated which button to press.

The precue was presented on the display before the cue, and consisted of three letters, one for each putative feature of the movement which Rosenbaum described as *Direction*,

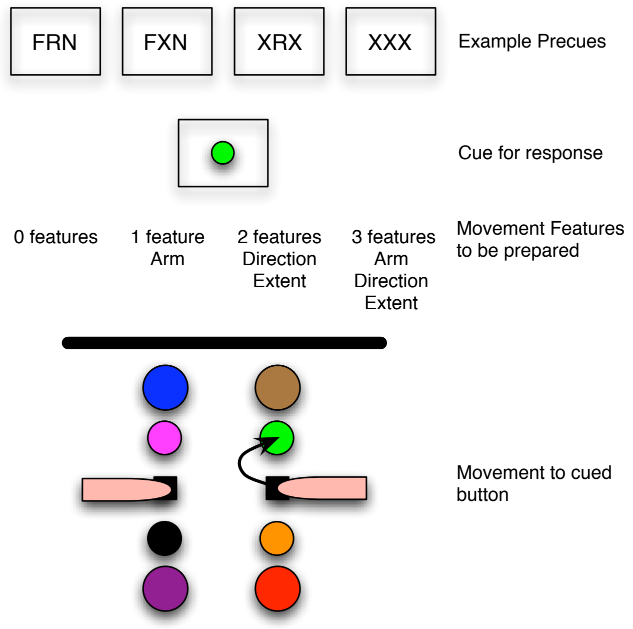

FRN
FXN
XRX
XXX
Example Precues
Cue for response
0 features
1 feature Arm
2 features Direction Extent
3 features Arm Direction Extent
Movement Features to be prepared
Movement to cued button

Figure 2. The Rosenbaum task. First a precue appears – four example are shown. Then appears a color-coded cue designating the button to be pressed. Depending on the precue, some number of movement features must be prepared, then the participant moves the left or right index finger from the home button to the designated button. The response buttons are hidden from the participant's view.

Arm, and *Extent*. For example, FRN stood for *forward, right, near*, which completely specifies the features of the movement to be made; in contrast, XRX specifies only a single feature, *right*. The participant was practiced in interpreting the precues. The logic of the paradigm is that when the cue appeared, the participant would have to prepare the remaining features before the movement could be initiated. Figure 2 shows additional examples of precues that vary the number of features that would have to be prepared before the movement could be initiated. The more features needing to be prepared, the greater the latency should be, and the results confirmed the prediction: zero, one, two and three features produced latencies of about 300, 450, 550, and 700 ms respectively. However, there were subtle and confusing specific-feature effects: different features appeared to require different times to prepare (ranging from 150 to 200 ms), and some features required different times depending on their values; for example, near movements were initiated faster than far movements, and more so if more features had to be prepared. Thus while demonstrating feature preparation time effects, the effects as presented were a complex mixture of general and specific effects.

Theorists of choice RT paradigms usually postulate a *response selection* stage of processing in which the stimulus is mapped to the response to be made. Motor feature preparation would follow response selection and should be a distinct process. However, it is clear that performing this task requires some complex mappings - first from the letter codes to movement features, and then from the cue color to the button. An immediate question that arises is the extent to which the effect of the precue is actually a response selection effect - maybe the precue is assisting response selection, not movement preparation.

To eliminate the possibility that purely cognitive response selection effects were responsible for the latency effects, Rosenbaum conducted a second, *decision task,* experiment in the participant viewed the precue and then the cue, and rather than making the response movement, made a vocal response for whether or not the cue was valid (consistent with) the precue. The resulting RTs showed a strong effect of the number of precued features, but no effect of the specific features or feature values. Even though he primary effect of number of features was present in both experiments, Rosenbaum claimed that the lack of feature-specific effects meant that response selection effects were not responsible for the differences in movement RTs. However, this argument is hard to understand – the feature preparation concept would not seem to require feature-specific effects, which in any case are hard to explain. Additionally, the logic of deciding which movement to make would seem to overlap a lot with deciding whether the movement could be made. The present author correlated the mean decision RT with the mean movement RT for each precue condition, and discovered that 91% of the variance in movement RT is accounted for by the decision RT. This strongly suggests, contrary to Rosenbaum's claim, that most of the movement RT is accounted for by some form of response selection process, even if there are specific feature effects.

Response Selection Effects: S-R compatibility

A long-studied aspect of response selection is *S-R compatibility*, which can be described as the ease with which the mapping from stimulus to response can be made. See Proctor & Vu (2006) for a recent review, and Rosenbaum & Newell (1987) or John, Rosenbloom, & Newell (1985) for computational model accounts of some forms of S-R compatibility. One feature of Rosenbaum's task is that both the precues and the cues would require a complex mapping to the actual response movements. Goodman and Kelso (1980) examined this issue in a critical but usually overlooked response to Rosenbaum. They first replicated Rosenbaum's results using color words or number labels for the target buttons. In a second study they used a precue and cue display, diagrammed in Figure 3, consisting of an array of lights in the same spatial arrangement as the response buttons. Precues were indicated by illuminating the lights corresponding to the buttons consistent with the precued movement features. For example, the FRN precue would illuminate the single light for the button corresponding to the three features, while XRX precue would illuminate all the lights for the right-hand side of the button set. The cue would then consist of the single light for the to-be-pressed button. This presentation has an especially powerful form of S-R compatibility termed *spatial compatibility* – the spatial properties of the stimulus map directly to the spatial properties of the response.

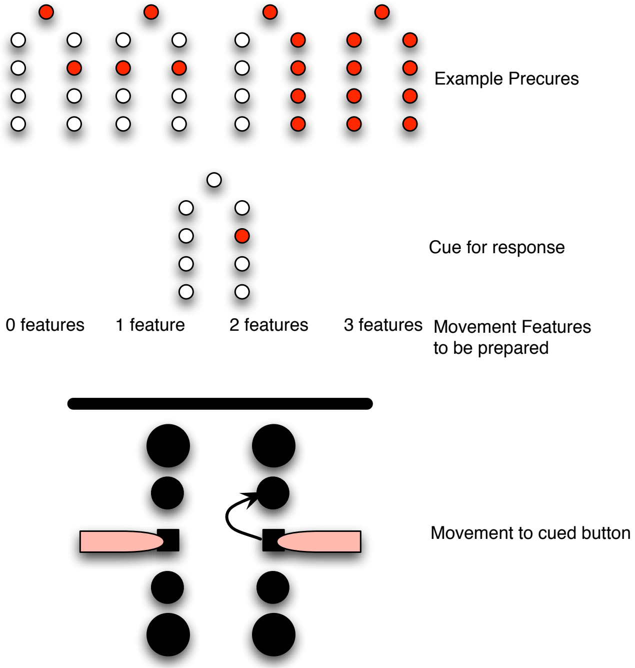


Figure 3. The Goodman & Kelso version of the Rosenbaum task. First a precue appears on an array of lights that matches the layout of the response buttons. The top light indicates a precue (vs. cue) display. Then in the same array appears a cue designating the button to be pressed. Depending on the precue, some number of movement features must be prepared, then the participant moves the left or right index finger from the home button to the designated button.

Compared to Rosenbaum's and the replication presentations. this spatially-compatible presentation of precue and cue information drastically reduced the effect of number of precued features. In contrast to the 300-700 ms range of Rosenbaum's latencies, the range was only about 250-350 ms.

S-R compatibility is normally assumed to be a response-selection process; there is nothing in the movement feature concept that suggests S-R compatibility would be involved. That is, once the response has been selected, the spatial similarity of the stimulus to the response should be irrelevant to computing the movement features. This is a further strong suggestion that Rosenbaum's effects were actually response-selection effects rather than movement preparation effects.

Response Selection Effects: Hick's Law

A second major aspect of response selection difficulty is the number of possible responses in the selection set, long codified as Hick's Law (Hick, 1952), which states that the RT in a choice reaction task is proportional to *log*$_2$ of the number of alternative (possible) responses. One consequence is that if the number of possible responses is held constant, then the RT should be constant. This would take ordinary response selection effects out of the picture, leaving only motor feature programming to produce RT differences. Thus, Goodman and Kelso (1980) conducted an additional experiment that held the number of possible responses constant at two. Using the same compatible display, they precued two possible responses by precuing both possible values of a single feature, such as illuminating the lights for both right and left forward near buttons. Also included were ambiguous precues that illuminated two lights, but which had no feature values in common, such as left-rearward-far and right-forward-near. These results were highly persuasive: the movement RTs were virtually identical at about 300 ms for all feature precue types, including the ambiguous precues. Apparently the specific movement features involved were irrelevant; what matters is only the difficulty of response selection, governed in this case by the number of possible responses.

It is also an old result that Hick's Law effects disappear in the presence of high S-R compatibility (Teichner & Krebs, 1974). Goodman & Kelso's highly compatible presentation of cues and precues drastically reduced the putative feature programming effects, and when the number of alternatives was held constant, they disappeared altogether.

This suggests that other aimed movement tasks in which S-R compatibility is manipulated might shed light on whether movement feature preparation is involved. That is, if S-R compatibility results in no Hick's Law effects, then there would be no response selection effects to be confused with feature preparation, and then perhaps other evidence of motor feature preparation would be visible, such as a movement latency long enough to have "room" for something like 50 ms or more per feature, and evidence of feature reuse, as described above, in which a repeated movement could be initiated more quickly.

Dassonville, Lewis, Foster, and Ashe (1999) had participants make joystick movements to place a cursor on visible targets arranged in a circle around the starting position, with various cues that differed in compatibility. In highly compatible mappings, there was no effect of the number of possible targets (no Hick's Law effect) and a latency of only about 300 ms. If the cue/response was repeated, the second response was substantially faster in the incompatible mappings, but not in the compatible mappings.

Wright, Marino, Belovsky, and Chubb (2007) had participants move a physical stylus from a starting point to one of several target pads arranged in an arc. The movement target and response cue was indicated by illuminating the pad, a perfectly compatible S-R mapping. There was no Hick's Law effect of the number of targets, the latencies were about 250 ms, and there was little or no effect of repetitions.

These results all point to the same conclusion: The motor feature preparation hypothesis states that features should require substantial time to prepare before a movement could be initiated and then could be reused in subsequent movements. Instead the effects are due to response selection effects described by Hick's Law, and when these effects are removed by highly compatible specifications of movement targets, reuse effects disappear, and the movement is launched so rapidly that there is no time to spare from other aspects of EPIC's architecture for feature programming to occur.

The same story for eye movements

A parallel story appears in the case of eye movements. Another key demonstration of motor feature programming was Abram & Jonides (1988) who applied the Rosenbaum feature preparation paradigm to saccade preparation. The movement task, illustrated in Figure 4, consisted of a saccade to one of four targets, two on each side of the fixation point at different distances. The precue and cue were shown in four circles, two on each side of the fixation point, inside the actual targets. The saccade responses required were *anti-saccades* — the eye movement had to be made in the *opposite* direction from the precue or cued direction, an S-R incompatibility. The saccade latencies increased by about 50 ms per feature preparation required, consistent with the feature preparation model. But in a second experiment, they required compatible response saccades, and held the number of precued alternatives constant at two. While there were some feature-specific differences, the saccade latency was basically constant across number of precued features, corresponding to the Goodman & Kelso (1980) results with the number of possible responses held constant. Again the feature-preparation effect seems to be confounded with a response selection effect.

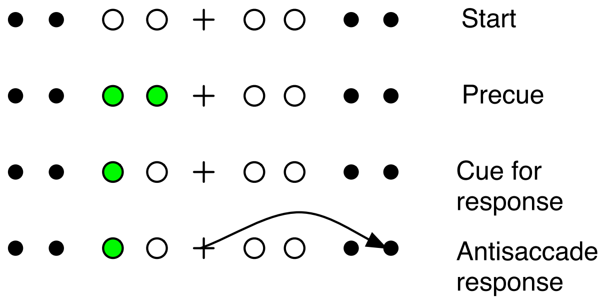


Figure 4. The Abram and Jonides task. The trial starts with the participant fixating the central cross. A precue is then presented showing e.g. the direction and both possible extents of the movement. Then a cue appears designating the actual movement target, one of the four small outer circles. The participant responds by fixating the target at the same distance but opposite direction as the cue.

What if the cue and response are more compatible? As illustrated in Figure 5, Crawford and Mueller (1990) used targets that were six lights, three on each side of the fixation point. A precue consisted of a background illumination around the possible target; the cue was illuminating the target light itself; in response, the participant made an eye movement to the illuminated target. The precue locations were either the same as the target (valid), different from the target (invalid), or at the fixation point (neutral), and presented either 100 ms or 500 ms before the cue. The results were very short latencies (about 250 ms), a small benefit of valid or neutral precues if the precue-cue delay was short, and no effect at all if it was long. Such an effect would not be expected from the motor feature preparation concept - if anything, the benefit of the precue should be larger with more time. Rather the delay results suggest some low-level visual effect on saccade initiation.

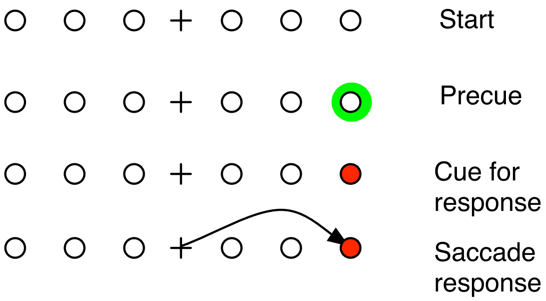


Figure 5. The Crawford and Mueller task. The trial starts with the participant fixating the central cross. A precue is then presented, e.g. a valid cue designating the future movement target. Then a cue appears designating both the actual movement target and acting as the stimulus for the movement. The participant responds by fixating the target.

Additional studies further clarify the compatibility effects for eye movements. Lee, Keller, and Heinen (2005) had participants make eye movements to memorized color-coded locations in a circular array given a color cue, not unlike Rosenbaum's approach. Hick's Law effects were observed. Kveraga, Berryhill, and Hughes (2002) and Kveraga, Boucher, and Hughes (2005) used targets arranged in a circle or semicircle, and the movement cue was co-located with the target, producing no Hick's Law effect. However, if anti-saccades or key presses were required to the same stimuli, Hick's Law effects were obtained.

The results for eye movements point to the same conclusion as for aimed manual movements: Effects suggesting motor feature preparation for eye movements are better explained as response selection effects accounted for by Hick's Law, and when these effects are removed by highly compatible specifications of movement targets, there is no evidence of feature preparation and the movement is launched so rapidly that there is no time to spare for feature programming to occur.

Conclusion

It was wrong

Empirically, once the target has been visually identified, an aimed manual movement or eye movement can be quickly launched to it without any S-R translation or motor feature programming delays; there is no evidence of feature programming effects. In terms of the EPIC architecture, once the production rules have identified the target of a movement as an object currently visible, and passed the identity of that object to the motor processor in a movement command, the movement will be initiated without any feature programming time. There seems to be no reason to maintain feature preparation delays for aimed movements in the architecture at the cost of making the models substantially more difficult to fit to important classes of data, especially in the high-speed performance tasks that motivated the design of EPIC. This original feature of EPIC was simply an incorrect overgeneralization. Fortunately, the solution is simple: set the per-feature preparation time for saccades and aimed manual movements to zero.

Implications for previous models

What effect does this change have on previous models built with EPIC? As mentioned earlier, most of the models in the original Meyer & Kieras (1997, 1999) work used keypress responses, which are not affected by this correction because they would not seem to be aimed manual movements (but see Welford, 1971).

Furthermore, because experimental results are typically aggregated over specific response movements, the net effect is that previous models using aimed manual movements or eye movements have a variable component of response time that instead of being due to movement preparation, has to be reattributed to stimulus encoding or response selection. At this point the theoretical implications appear to be minor.

Should feature preparation be discarded for keypress movements as well? Unfortunately, this question cannot be easily answered because the motor control literature remains so sparse (Rosenbaum, 2005) that we are still in the earliest stages of our theoretical understanding of how movements are performed. An interim heuristic would be to assess whether keystroke feature preparation can be replaced by changing the response selection strategy.

Acknowledgment

This work was supported by the Office of Naval Research, under Grant No. N00014-06-1-0034.

References

Abrams, R.A., & Jonides, J. (1988). Programming saccadic eye movments. *Journal of Experimental Psychology: Human Perception and Performance*, **14**, 428-443.

Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Crawford, T.J., & Muller, H.J. (1992). Spatial and temporal effects of spatial attention on human saccadic eye movements. *Vision Research*, **32**, 293-304.

Dassonville, P., Lewis, S.M., Foster, H.E., & Ashe, J. (1999). Choice and stimulus–response compatibility affect duration of response selection. *Cognitive Brain Research*, **7**, 235–240.

Findlay, J. (1997). Saccade target selection during visual search. *Vision Research*, **37**, 617–631.

Goodman, D., & Kelso, J.A.S. (1980). Are movements prepared in parts? Not under compatible (naturalized) conditions. *Journal of Experimental Psychology: General*, **109**, 475-495.

Hick, W.E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, **4**, 11–26.

John, B. E., Rosenbloom, P. S., & Newell, A. (1985). A theory of stimulus-response compatibility applied to human-computer interaction. In *Proceedings of CHI 1985*, New York: ACM. pp. 212-219.

Kieras, D.E. (2004). EPIC Architecture Principles of Operation. Web publication available at ftp://www.eecs.umich.edu/people/kieras/EPIC/EPICPrinOp.pdf

Kieras, D.E. (2007). The control of cognition. In W. Gray (Ed.), *Integrated models of cognitive systems*. (pp. 327 - 355). Oxford University Press.

Kieras, D.E, & Marshall, S.P. (2006). Visual Availability and Fixation Memory in Modeling Visual Search using the EPIC Architecture. *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, 423-428.

Kieras, D. & Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12, 391-438.

Kveraga, K., Berryhill, M., & Hughes, H.C. (2006). Directional uncertainty in visually guided pointing. *Perceptual and Motor Skills*, **102**, 125–132.

Kveraga, K., Boucher, L., & Hughes, H.C. (2002). Saccades operate in violation of Hick's law. *Experimental Brain Research*, **146**, 307–314.

Lee, K.M., Keller, E.L., Heinen, S.J. (2005). Properties of saccades generated as a choice response. *Experimental Brain Research*, **162**, 278-286.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, **104**, 3-65.

Meyer, D. E., & Kieras, D. E. (1999). Precis to a practical unified theory of cognition and action: Some lessons from computational modeling of human multiple-task performance. In D. Gopher & A. Koriat (Eds.), *Attention and Performance XVII*. (pp. 15-88) Cambridge, MA: M.I.T. Press.

Proctor, R.W., & Vu, K.L. (2006). *Stimulus-response compatibility principles: Data, theory, and application*. New York: CRC Press.

Rosenbaum, D. A. (1980). Human movement initiation: Specification of arm, direction, and extent. *Journal of Experimental Psychology: General*, **109**, 444-474.

Rosenbaum, D. A. (1991). *Human motor control*. New York, Academic Press.

Rosenbaum, D.A. (2005). The Cinderella of psychology; The neglect of motor control in the science of mental life and behavior. *American Psychologist*, **50**(4), 308-317.

Rosenbloom, P.S., & Newell, A. (1987). An integrated model of stimulus-response compatibility and practice. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol, 21, pp 1-52). New York:Academic Press.

Teichner, W.H., & Krebs, M.J. (1974). Laws of visual choice reaction time. *Psychological Review*, **81**, 75–98.

Welford, A.T. (1971). What is the basis of choice reaction-time? *Ergonomics*, **14**, 679-693.

Wright, C.E., Marino, V.F., Belovsky, S.A., & Chubb, C. (2007). Visually guided, aimed movements can be unaffected by stimulus–response uncertainty. *Experimental Brain Research*, **179**, 475–496.

Optimizing Memory Retention with Cognitive Models

Robert Lindsey (robert.lindsey@colorado.edu)
Michael Mozer (mozer@colorado.edu)
Department of Computer Science and Institute of Cognitive Science
University of Colorado at Boulder

Nicholas J. Cepeda (ncepeda@yorku.ca)
Department of Psychology, York University

Harold Pashler (hpashler@ucsd.edu)
Department of Psychology, University of California at San Diego

Abstract

When individuals learn facts (e.g., foreign language vocabulary) over multiple sessions, the durability of learning is strongly influenced by the temporal distribution of study (Cepeda, Pashler, Vul, Wixted, & Rohrer, 2006). Computational models have been developed to explain this phenomenon known as the *distributed practice effect*. These models predict the accuracy of recall following a particular study schedule and retention interval. To the degree that the models embody mechanisms of human memory, they can also be used to determine the spacing of study that maximizes retention. We examine two memory models (Pavlik & Anderson, 2005; Mozer, Pashler, Lindsey, & Vul, submitted) that provide differing explanations of the distributed practice effect. Although both models fit experimental data, we show that they make robust and opposing predictions concerning the optimal spacing of study sessions. The Pavlik and Anderson model robustly predicts that contracting spacing is best over a range of model parameters and retention intervals; that is, with three study sessions, the model suggests that the lag between sessions one and two should be larger than the lag between sessions two and three. In contrast, the Mozer et al. model predicts equal or expanding spacing is best for most material and retention intervals. The limited experimental data pertinent to this disagreement appear to be consistent with the latter prediction. The strong contrast between the models calls for further empirical work to evaluate their opposing predictions.

Keywords: distributed practice effect; optimization; study schedules

Introduction

In educational settings, individuals are often required to memorize facts such as foreign language vocabulary words. A question of great practical interest is how to retain knowledge once acquired. Psychologists have identified factors influencing the durability of learning, most notably the temporal distribution of practice: when individuals study material across multiple sessions, long-term retention generally improves when the sessions are spaced in time. This effect, known as the *distributed practice* or *spacing* effect, is typically studied via an experimental paradigm in which participants are asked to study items over two or more sessions, and the time between sessions—the *interstudy interval* or *ISI*—is varied. Retention is often evaluated via a cued recall test at a fixed lag following the final study sessionthe *retention interval* or *RI* (Figure 1).

Typical experimental results are shown in the data points and dotted lines of Figures 2a (Glenberg, 1976) and 2b (Cepeda, Vul, Rohrer, Wixted, & Pashler, 2008). In both experiments, participants studied material at two points in time, with a variable ISI, and then were tested following a fixed RI. The graphs show recall accuracy at test as a function of ISI for several different RIs. The curves, which we will refer to as *spacing functions*, typically show a rapid rise in memory retention as ISI increases, reach a peak, and then gradually drop off. From the spacing function, one can determine the *optimal ISI*, the spacing of study that yields maximal retention. The exact form of the spacing function depends on the specific material to be learned and the RI. The distributed practice effect is obtained over a wide range of time scales: ISIs and RIs in the Glenberg study are on the order of seconds to minutes, and in the Cepeda et al. study are on the order of weeks to months. On the educationally relevant time scale of months, optimally spaced study can double retention over massed study. Thus, determining the optimal spacing of study can have a tremendous practical impact on human learning.

Pavlik and Anderson (2005; 2008) used the ACT-R declarative memory equations to explain distributed practice effects. ACT-R supposes a separate trace is laid down for each study and that the trace decays according to a power function of time. The key feature of the model that yields the distributed practice effect is that the decay rate of a new trace depends on an item's current memory strength at the point in time when the item is studied. This ACT-R model has been fit successfully to numerous experimental datasets. The solid lines of Figure 2a show the ACT-R fit to the Glenberg data.

Mozer, Pashler, Lindsey, and Vul (submitted) have recently proposed a model providing an alternative explanation of the distributed practice effect. In this model, when an item is studied, a memory trace is formed that includes the current *psychological context*, which is assumed to vary randomly over time. Probability of later recall depends in part on the similarity between the context representations at study and test. The key feature of this model that distinguishes it from related past models (e.g., Raaijmakers, 2003) is that the context is assumed to wander on multiple time scales. This

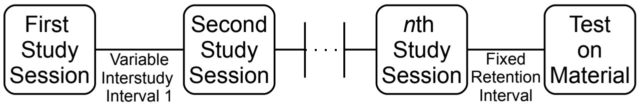


Figure 1: Structure of a study schedule.

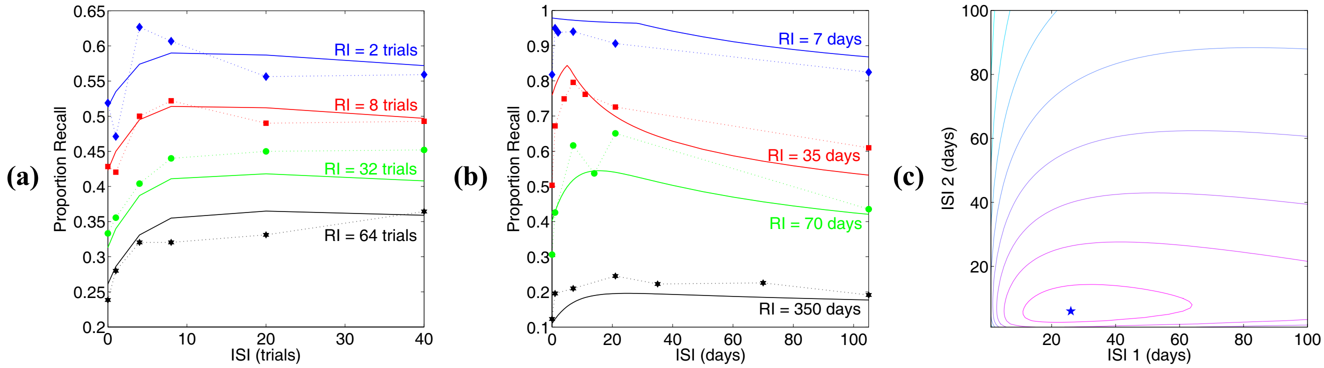


Figure 2: Results from **(a)** Glenberg (1976) and **(b)** Cepeda et al. (2008) illustrative of the distributed practice effect. The dotted lines correspond to experimental data. The solid lines in **(a)** and **(b)** are the ACT-R and MCM fits to the respective data. **(c)** A contour plot of recall probability as a function of two ISIs from ACT-R with parameterization in Pavlik and Anderson (2008).

model, referred to as the *multiscale context model* (MCM), has also been successfully fit to numerous empirical datasets, including the Glenberg study. In Figure 2b, we show the MCM prediction (solid lines) of the Cepeda et al. data.

Both ACT-R and MCM can be parameterized to fit data post hoc. However, both models have been used in a predictive capacity. Pavlik and Anderson (2008) have used ACT-R to determine the order and nature of study of a set of items, and showed that ACT-R schedules improved retention over alternative schedules. Mozer et al. (submitted) parameterize MCM with the basic forgetting function for a set of items (the function relating recall probability to RI following a single study session) and then predict the spacing function for the case of multiple study sessions. Figure 2b is an example of such a (parameter free) prediction of MCM.

Most experimental work involves two study sessions, the minimum number required to examine the distributed-practice effect. Consequently, models have mostly focused on this simple case. However, naturalistic learning situations typically offer more than two opportunities to study material. The models can also predict retention following three or more sessions. In this paper, we explore predictions of ACT-R and MCM in order to guide the design of future experiments that might discriminate between the models.

Study Schedule Optimization

A cognitive model of the distributed practice effect allows us to predict recall accuracy at test for a particular study schedule and RI. For example, Figure 2c shows ACT-R's prediction of recall probability for a study schedule with two variable ISIs and an RI of 20 days, for a particular parameterization of the model based on Pavlik and Anderson (2008). It is the two-dimensional generalization of the kind of spacing functions illustrated in Figures 2a and 2b. Recall probability, shown by the contour lines, is a function of both ISIs. The star in Figure 2c indicates the schedule that maximizes recall accuracy.

Models are particularly important for study-schedule optimization. It is impractical to determine optimal study schedules empirically because the optimal schedule is likely to depend on the particular materials being learned and also because the combinatorics of scheduling $n+1$ study sessions (i.e., determining n ISIs) make it all but impossible to explore experimentally for $n > 1$. With models of the distributed practice effect, we can substitute computer simulation for exhaustive human experimentation.

In real-world learning scenarios, we generally do not know exactly when studied material will be needed; rather, we have a general notion of a span of time over which the material should be retained. Though not the focus of this paper, models of the distributed practice effect can be used to determine study schedules that maximize retention not only for a particular prespecified RI, but also for the situation in which the RI is treated as a random variable with known distribution. The method used in this paper to determine optimal study schedules can easily be extended to accomodate uncertain RIs.

Pavlik and Anderson ACT-R Model

In this section, we delve into more details of the Pavlik and Anderson (2005; 2008) model, which is based on ACT-R declarative memory assumptions. In ACT-R, a separate trace is laid down each time an item is studied, and the trace decays according to a power law, t^{-d}, where t is the age of the memory and d is the power law decay for that trace. Following n study episodes, the activation for an item, m_n, combines the trace strengths of individual study episodes:

$$m_n = \beta_s + \beta_i + \beta_{si} + \ln\left(\sum_{k=1}^{n} b_k t_k^{-d_k}\right),$$

where t_k and d_k refer to the age (in seconds) and decay associated with trace k, and the additive parameters β_s, β_i, and β_{si} correspond to participant, item, and participant-item factors that influence memory strength, respectively. The variable b_k reflects the salience of the kth study session (Pavlik, 2007); larger values of b_k correspond to cases where, for example, the participant self-tested and therefore exerted more effort.

The key claim of the ACT-R model with respect to the distributed-practice effect is that the decay term on study trial k depends on the item's overall activation at the point when study occurs, according to the expression:

$$d_k(m_{k-1}) = ce^{m_{k-1}} + \alpha,$$

where c and α are constants. If spacing between study trials is brief, the activation m_{k-1} is large and consequently the new study trial will have a rapid decay, d_k. Increasing spacing can therefore slow memory decay of trace k, but it also incurs a cost in that traces $1...k-1$ will have substantial decay.

The model's recall probability is related to activation by:

$$p(m) = 1/(1+e^{\frac{\tau-m}{s}}),$$

where τ and s are additional parameters. The pieces of the ACT-R model relevant to this paper include 3 additional parameters, for a total of 10 parameters, including: h, a translation of real-world time to internal model time, u, a descriptor of the maximum benefit of study, and v, a descriptor of the rate of approach to the maximum.

Pavlik and Anderson (2008) use ACT-R activation predictions in a heuristic algorithm for scheduling the trial order *within* a study session, as well as the trial type (i.e., whether an item is merely studied, or whether it is first tested and then studied). They assume a fixed intersession spacing. Thus, their algorithm reduces to determining how to best allocate a finite amount of time within a session.

Although they show a clear effect of the algorithm used for within-session scheduling, we focus on the complementary issue of scheduling the lag between sessions. The ISI manipulation is more in keeping with the traditional conceptualization of the distributed-practice effect. Fortunately, the ACT-R model can be used for both within- and between-session scheduling. To model between-session scheduling, we assume—as is true in controlled experimental studies—that each item to be learned is allotted the same amount of study (or test followed by study) time within a session.

Pavlik and Anderson (2008) describe their within-session scheduling algorithm as optimizing performance, yet we question whether their algorithm is appropriately cast in terms of optimization. They argue that maximizing probability of recall should not be the goal of a scheduling algorithm, but that activation gain at test should be maximized so as to encourage additional benefits (e.g., improved long-term retention). We believe that had Pavlik and Anderson (2008) sought simply to maximize probability of recall at test and had more rigorously defined their optimization problem, they would have seen results of the ACT-R within-session scheduler even better than what they achieved. In light of these facts, we contend that our work is the first effort to truly optimize memory retention via cognitive models.

Multiscale Context Model

One class of theories proposed to explain the distributed-practice effect focuses on the notion of encoding variability. According to these theories, when an item is studied, a memory trace is formed that incorporates the current psychological context. Psychological context includes conditions of study, internal state of the learner, and recent experiences of the learner. Retrieval of a stored item depends partly on the similarity of the contexts at the study and test. If psychological context is assumed to fluctuate randomly, two study sessions close together in time will have similar contexts. Consequently, at the time of a recall test, either both study contexts will match the test context or neither will. A longer ISI can thus prove advantageous because the test context will have higher likelihood of matching one study context or the other.

Raaijmakers (2003) developed an encoding variability theory by incorporating time-varying contextual drift into the Search of Associative Memory (SAM) model and used this model to explain data from the distributed-practice literature. The context consists of a pool of binary-valued neurons which flip state at a common fixed rate. This behavior results in exponentially decreasing similarity between contexts at study and test time as a function of the study-test lag.

In further explorations, we (Mozer et al., submitted) found a serious limitation of SAM: Distributed-practice effects occur on many time scales (Cepeda et al., 2006). SAM can explain effects for study sessions separated by minutes or hours, but not for sessions separated by weeks or months. The reason is essentially that the exponential decay in context similarity bounds the time scale at which the model operates.

To address this issue, we proposed a model with multiple pools of context neurons. The pools vary in their relative size and the rate at which their neurons flip state. With an appropriate selection of the pool parameters, we obtain a model that has a power-law forgetting function and is therefore well suited for handling multiple time scales. The notion of multiscale representations comes from another model of distributed-practice effects developed by Staddon, Chelaru, and Higa (2002) to explain rat habituation. We call our model, which integrates features of SAM and Staddon et al.'s model, the Multiscale Context Model (MCM).

MCM has only five free parameters. Four of these parameters configure the pools of context neurons, and these parameters can be fully constrained for a set of materials to be learned by the the basic forgetting function—the function characterizing recall probability versus lag between a single study opportunity and a subsequent test. Given the forgetting function, the model makes strong predictions concerning recall performance at test time given a study schedule.

MCM predicts the outcome of four experiments by Cepeda et al. (in press, 2008). These experiments all involved two study sessions with variable ISIs and RIs. Given the basic forgetting functions for the material under study, MCM accurately predicted the ISI yielding maximal recall performance at test for each RI. Although MCM is at an early stage of development, the results we have obtained are sufficiently promising and robust that we find it valuable to explore the model's predictions and to compare them to the well-established ACT-R model.

Comparing Model Predictions

Having introduced the ACT-R model and MCM, we now turn to the focus of this paper: obtaining predictions from the two models to determine whether the models are distinguishable. We focus on the most important, practical prediction that the models can make: how to schedule study to optimize memory retention. We already know that the models make similar predictions in empirical studies with two study sessions (one ISI); we therefore turn to predictions from the models with more than two sessions (two or more ISIs). Even if the models make nonidentical predictions, they may make predictions that are quantitatively so similar the models will in practice be difficult to distinguish. We therefore focus our explorations on whether the models make qualitatively different predictions. Constraining our explorations to study schedules with three study sessions (i.e., two ISIs), we test whether the models predict that optimal study schedules have *expanding*, *contracting*, or *equal* spacing, that is, schedules in which ISI 1 is less than, greater than, or equal to ISI 2, respectively. For the sake of categorizing study schedules, we judge two ISIs to be equal if they are within 30% of one another. The key conclusions from our experiments do not depend on the precise setting of this criterion.

In all simulations, we used the Nelder-Mead Simplex Method (as implemented in Matlab's `fminsearch`) for finding the values of ISI 1 and ISI 2 that yield the maximum recall accuracy following a specified RI. Because this method finds local optima, we used random restarts to increase the likelihood of obtaining global optima. We observed some degenerate local optima, but for the most part, it appeared that both models had spacing functions like those in Figures 2a and 2b with a single optimum.

Our first exploration of the models' spacing predictions uses parameterizations of the models fit to the Glenberg (1976) data (Figure 2a for ACT-R, not shown for MCM). Because the models have already been constrained by the experimental data, which involved two study opportunities, they make strong predictions concerning memory strength following three spaced study opportunities. We used the models to predict the (two) optimal ISIs for RIs ranging from ten minutes to one year. We found that both models predict contracting spacing is optimal regardless of RI. The spacing functions obtained from the models look similar to that in Figure 2c. Because the models cannot be qualitatively discriminated based on the parameters fit to the Glenberg experiment, we turn to exploring a wider range of model parameterizations.

Randomized Parameterizations

In this section, we explore the predictions of the models across a wide range of RIs and model parameterizations, in order to determine whether we can abstract regularities in the models' predictions that could serve to discriminate between the models. In particular, we are interested in whether the optimality of contracting spacing predicted by both models for the Glenberg paradigm and material is due to peculiarities of that study, or whether optimality of contracting spacing is a robust parameter-independent prediction of both models.

Methodology. We performed over 200,000 simulations for each model. In our simulations, we systematically varied the RIs from roughly 10 seconds to 300 days. We also chose random parameter settings that yielded sensible behavior from the models. We later expand on the notion of "sensible."

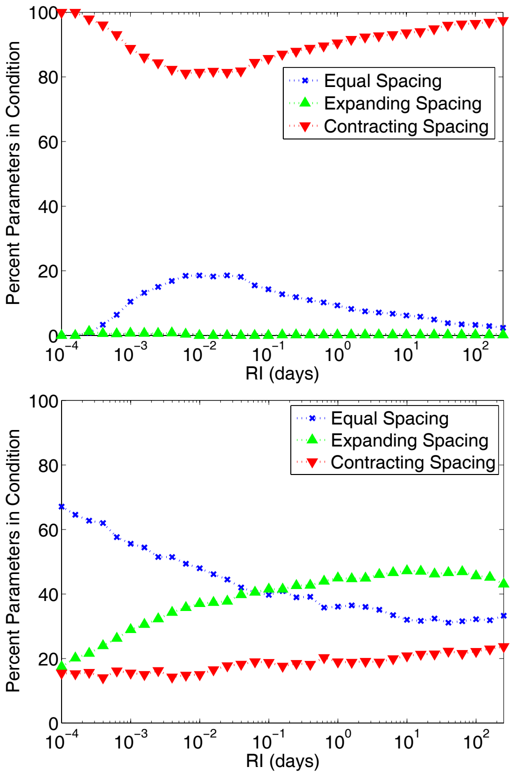


Figure 3: The distribution of qualitative spacing predictions of ACT-R (upper panel) and MCM (lower panel) as a function of RI, for random model variants. Each point corresponds to the percentage of valid model fits that produced a particular qualitative spacing prediction.

For the ACT-R model, we draw the parameters β_i, β_s, β_{si} from Gaussian distributions with standard deviations specified in Pavlik and Anderson (2008). The parameters h, c, and α are drawn from a uniform distribution in [0, 1]. The study weight parameter b is fixed at 1, which assumes test-practice trials (Pavlik & Anderson, 2008). Remaining parameters of the model are fixed at values chosen by Pavlik and Anderson (2008). For MCM, we vary the four parameters that determine the shape of the forgetting function.

To ensure that the randomly generated parameterizations of both models are sensible—i.e., yield behavior that one might expect to observe of individuals studying specific materials—we observe the forgetting function for an item studied once and then tested following an RI, and place two criteria on the forgetting function: (1) With an RI of one day, recall probability must be less than 0.80. (2) With an RI of thirty days, recall probability must be greater than 0.05. We thus eliminate parameterizations that yield unrealistically small amounts of forgetting and too little long-term memory.

Results. Results of our random-parameter simulations are presented in Figures 3 and 4. The upper graphs of each figure are for the ACT-R model and the lower graphs are for MCM. Figure 3 shows, as a function of the RI, the proportion of simulations that yield contracting (red curve), expanding (green curve), and equal (blue curve) optimal spacing. The ACT-R model (Figure 3, upper) strongly predicts that contracting spacing is optimal, regardless of the RI and model parameters. In contrast, MCM (Figure 3, lower) suggests that the qualitative nature of the optimal study schedule is more strongly dependent on RI and model parameters. As the RI increases, the proportion of expanding spacing predictions slowly increases and the proportion of equal spacing predictions decreases; contracting spacing predictions remain relatively constant. Over a variety of materials to be learned (i.e., parameterizations of the model), MCM predicts that expanding spacing becomes increasingly advantageous as the RI increases.

Each scatter plot in Figure 4 contains one point per random simulation, plotted in a log-log space that shows the values of the optimal ISI 1 on the x-axis and the optimal ISI 2 on the y-axis. In other words, each point is like the star (point of optimal retention) of Figure 2c, plotted for a unique parameterization and RI. The two solid diagonal lines represent the decision boundary between the different qualitative spacing predictions. Points between the decision boundaries are within 30% of each other (in linear space) and fall under the label of equal spacing. Points above the upper diagonal line are classified as expanding spacing, and points below the lower diagonal line are classified as contracting spacing. The color of the individual points specifies the corresponding RI.

The spacing functions produced by the ACT-R model are fairly similar, which is manifested not only in the consistency of the qualitative predictions (Figure 3, upper), but also the optimal ISIs (Figure 4, upper). The relationship between optimal ISI 1 and optimal ISI 2 appears much stronger for the ACT-R model than for MCM, and less dependent on the specific model parameterization. Not only do we observe a parameter-independent relationship between the optimal ISIs, but we also observe a parameter-independent relationship between the RI and each of the ISIs. The apparent linearity in the upper panel of Figure 4 translates to a linear relationship in log-log space between RI and each of the optimal ISIs. The least-squares regression yields:

$$\log_{10}(ISI_1) = 1.0164\log_{10}(RI) + 0.5091$$
$$\log_{10}(ISI_2) = 1.0237\log_{10}(RI) + 0.9738$$

with coefficient of determination (ρ^2) values of 0.89 and 0.90, respectively. We emphasize that these relationships are predictions of a model, not empirical results. The only empirical evidence concerning the relationship between RI and the optimal ISI is found in Cepeda et al. (2006), who performed a meta-analysis of all cogent studies of the distributed-practice effect, and observed a roughly log-log linear relationship between RI and optimal ISI for experiments consisting of two study sessions (one ISI). Were this lawful relationship to exist, it could serve as an extremely useful heuristic for educators who face questions such as: If I want my students to study this material so that they remember it for six months until we return to the same topic, how should I space the two classes I have available to cover the material?

In further contrast with ACT-R, MCM's optimal ISI predictions are strongly parameter dependent (Figure 4, lower). Is this result problematic for MCM? We are indeed surprised by the model's variability, but there are no experimental data at present to indicate whether such variability is observed in optimal study schedules for different types of material (as represented by the model parameters).

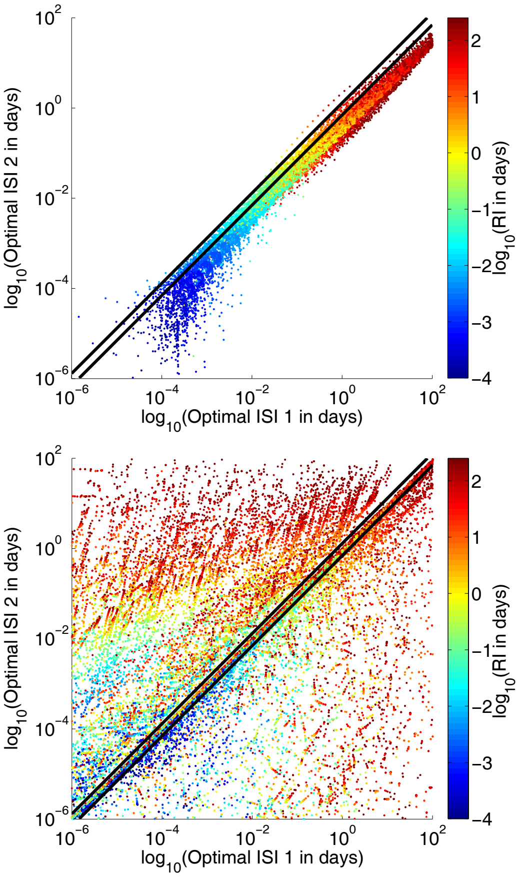


Figure 4: Optimal spacing predictions in log-space of ACT-R (upper figure) and MCM (lower figure) for random parameter settings over a range of RIs. Each point corresponds to a parameter setting's optimal spacing prediction for a specific RI, indicated by the point's color. The black lines indicate the boundaries between expanding, equal, and contracting spacing predictions.

Although ACT-R shows greater regularity in its predictions than MCM, as evidenced by the contrast between the upper and lower panels of Figure 4, note that both models make optimal spacing predictions that can vary by several orders of magnitude for a fixed RI. No experimentalist would be surprised by the prediction of both models that optimal spacing of study for a given RI is material-dependent, but this point has not been acknowledged in the experimental literature, and indeed, the study by Cepeda et al. (2008) would seem to suggest otherwise: two different types of material yielded spacing functions that appear, with the limited set of ISIs tested, to peak at the same ISI.

Another commonality between the models is that both clearly predict the trend that optimal ISIs increase with the RI. This is evidenced in Figure 4 by the fact that the long RIs (red points) are closer to the upper right corner than the short RIs (blue points). Although the experimental literature has little to offer in the way of behavioral results using more than two study sessions, experimental explorations of the distributed-practice effect with just two study sessions do suggest a monotonic relationship between RI and the optimal ISI (Cepeda et al., 2006).

Discussion

In this paper, we have explored two computational models of the distributed practice effect, ACT-R and MCM. We have focused on the educationally relevant issue of how to space three or more study sessions so as to maximize retention at some future time. The models show some points of agreement and some points of fundamental disagreement.

Both models have fit the experimental results of Glenberg (1976). With the parameterization determined by this fit, both models make the same basic prediction of contracting spacing being optimal when three study sessions are involved. Both models also agree in suggesting a monotonic relationship between the RI and the ISIs. Finally, to differing extents, both models suggest that optimal spacing depends not only on the desired RI, but also on the specific materials under study.

When we run simulations over the models' respective parameter spaces, we find that the two models make remarkably different predictions. ACT-R strongly predicts contracting spacing is best regardless of the RI and materials. In contrast, MCM strongly predicts that equal or expanding spacing is best, although it shows a greater dependence on both RI and the materials than does ACT-R. This stark difference between the models gives us a means by which the models can be evaluated. One cannot ask for any better set-up to pit one model against the other in an experimental test.

In reviewing the experimental literature, we have found only four published papers that involve three or more study sessions and directly compare contracting versus equal or contracting versus expanding study schedules (Foos & Smith, 1974; Hser & Wickens, 1989; Landauer & Bjork, 1978; Tsai, 1927). *All four studies show that contracting spacing leads to poorer recall at test than the better of expanding or equal spacing.* These findings are consistent with MCM and inconsistent with ACT-R. However, the findings hardly allow us to rule out ACT-R, because it would not be surprising if a post-hoc parameterization of ACT-R could be found to fit each of the experimental studies.

Nonetheless, the sharp contrast in the predictive tendencies of the two models (Figure 3) offers us an opportunity to devise a definitive experiment that discriminates between the models in the following manner. We conduct an experimental study with a single ISI and parameterize both models via fits to the resulting data. We then examine the constrained models' predictions regarding three or more study sessions. If ACT-R predicts decreasing spacing and MCM predicts equal or increasing spacing, we can then conduct a follow-on study in which we pit the predictions of two fully specified models against one another. We (Kang, Lindsey, & Pashler, in preparation) have just begun this process using Japanese-English vocabulary pairs that Pavlik and Anderson (2008) have modeled extensively with ACT-R. Without extensive simulation studies of the sort reported in this paper, one would not have enough information on how the models differ to offer an approach to discriminate the models via experimental data.

Acknowledgments

This work was supported NSF Science of Learning Center grant SBE-0542013 (Garrison Cottrell, PI), and by NSF BCS-0339103, BCS-720375, and SBE-0518699. We thank Dr. Pavlik for considerately helping us with the ACT-R model.

References

Cepeda, N. J., Coburn, N., Rohrer, D., Wixted, J. T., Mozer, M. C., & Pashler, H. (in press). Optimizing distributed practice: Theoretical analysis and practical implications. *J. Exp. Psych.* (In press)

Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psych. Bull.*, *132*, 354–380.

Cepeda, N. J., Vul, E., Rohrer, D., Wixted, J. T., & Pashler, H. (2008). Spacing effects in learning: A temporal ridgeline of optimal retention. *Psych. Sci.*, *19*, 1095–1102.

Foos, P. W., & Smith, K. H. (1974). Effects of spacing and spacing patterns in free recall. *J. Exp. Psych.*, *103*, 112–6.

Glenberg, A. M. (1976). Monotonic and nonmonotonic lag effects in paired-associated and recognition memory paradigms. *J. Verb. Learn. & Verb. Behav.*, *15*, 1–16.

Hser, Y. I., & Wickens, T. D. (1989). The effects of the spacing of test trials and study trials in paired-association learning. *Ed. Psych.*, *9*, 99–120.

Kang, S. H. K., Lindsey, R. V., & Pashler, H. (in preparation). *Investigations into study schedules.* (In preparation)

Landauer, T. K., & Bjork, R. A. (1978). Optimum rehearsal patterns and name learning. In *Practical aspects of memory* (pp. 625–632). London: Academic Press Inc.

Mozer, M. C., Pashler, H., Lindsey, R. V., & Vul, E. (submitted). *Predicting the optimal spacing of study: A multiscale context model of memory.* (Submitted)

Pavlik, P. I. (2007). Understanding and applying the dynamics of test practice and study practice. *Instruc. Sci.*, *35*, 407–441.

Pavlik, P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cog. Sci.*, *29*(4), 559-586.

Pavlik, P. I., & Anderson, J. R. (2008). Using a model to compute the optimal schedule of practice. *J. Exp. Psych.: Applied*, *14*, 101–117.

Raaijmakers, J. G. W. (2003). Spacing and repetition effects in human memory: application of the sam model. *Cog. Sci.*, *27*, 431–452.

Staddon, J. E. R., Chelaru, I. M., & Higa, J. J. (2002). A tuned-trace theory of interval-timing dynamics. *J. Exp. Anal. of Behav.*, *77*, 105–124.

Tsai, L. S. (1927). The relation of retention to the distribution of relearning. *J. Exp. Psych.*, *10*, 30–39.

The Locus of the Gratton Effect in Picture-Word Interference

Leendert van Maanen (leendert@ai.rug.nl)
Department of Artificial Intelligence, University of Groningen
P.O.Box 407 9000 AK Groningen, The Netherlands

Hedderik van Rijn (D.H.Van.Rijn@rug.nl)
Department of Psychology, University of Groningen
Grote Kruisstraat 2/1 9712 TS Groningen, The Netherlands

Abstract

It has been shown that between-trial effects in Stroop-like interference tasks are caused by differences in the amount of cognitive control. Trials following an incongruent trial show less interference, an effect suggested to result from the increased control caused by the previous trial (the Gratton effect). In this study we show that cognitive control not only results in a different amount of interference, but also in a different locus of the interference. That is, the stage of the task that shows the most interference changes as a function of the preceding trial. Using computational cognitive modeling we explain these effects by a difference in the amount of processing of the irrelevant dimension of the stimulus.

Keywords: Picture-word interference; Gratton effect; Cognitive control; Dual-task study (PRP); ACT-R; RACE/A.

Introduction

Picture-word interference is a Stroop-like interference effect that is observed when participants are asked to provide the name of a picture, while they should also try to ignore a word that is inscribed in the picture (e.g., Glaser & Düngelhoff, 1984). The common finding is that reaction times are increased if word and picture bear a categorical relationship, as opposed to when they do not bear a relationship. In addition, reaction times are decreased when word and picture are identical, that is, describe the same object. In many respects, this is analogous to the Stroop effect, in which color naming reaction times are increased for trials in which the word also is a color name, as opposed to trials in which the word is not a color name. Also, in Stroop experiments a decrease in reaction times is found when word and ink color refer to the same color name.

Many theories ascribe the congruency effect – the increased reaction times as a result of a categorical relationship between the word and the picture – to the semantic relation between picture and word (e.g., Glaser & Düngelhoff, 1984; Roelofs, 1992; Van Maanen & Van Rijn, 2007). A word that is a category-member of the picture (e.g., “dog” and a picture of a cat) makes picture naming harder than an unrelated word (e.g., “book” and a picture of a cat), resulting in increased reaction times. In addition, the congruency effect has also been ascribed to a failure to suppress the more automatic word reading response (e.g., Lovett, 2005; MacLeod & Dunbar, 1988). Thus, because it is hard to not read a word, it will interfere with a response on the color or picture, resulting in increased reaction times.

The amount of suppression of the automized reading response has been hypothesized to be under cognitive control (e.g., Botvinick, Braver, Barch, Carter, & Cohen, 2001). This means that a control mechanism exists that dynamically adapts the amount of suppression of the reading response to the task demands.

For instance, the influence of cognitive control is observed as a between-trial effect in congruency tasks, in which the congruency effect is decreased in trials following an incongruent trial. This effect has been interpreted as an increase in control, resulting from the increased difficulty of the task (Verguts & Notebaert, 2008). Similarly, the congruency effect is increased after congruent trials, suggesting a relaxation in control of the reading response. This particular between-trial effect is referred to as the Gratton effect (Gratton, Coles, & Donchin, 1992).

Experiment

To study the locus of the interference leading to the Gratton effect, we re-analyzed a picture-word interference experiment in a Psychological Refractory Period (PRP) paradigm (Van Maanen, Van Rijn, & Taatgen, submitted). In a PRP design, participants are asked to perform two tasks sequentially. The first task is often relatively simple, whereas the second task is the task of interest (the main task). The interval between the stimulus onsets of the two tasks is manipulated (Stimulus Onset Asynchrony or SOA). A typical finding, known as the PRP effect (Telford, 1931) is a negative correlation between SOA and response latency on the main task. Responses to the first task are typically unaffected by varying the SOA.

The PRP effect has been explained by the assumption that both tasks share a cognitive resource that can only be used by one task at a time (e.g., Pashler, 1994; Welford, 1967). Thus, the second task is delayed because the first task still requires a critical resource, as illustrated by Figure 1. As the interval between the tasks increases, the delay becomes smaller, resulting in a faster main task response.

The PRP design has been used to study the locus of various effects (e.g., for PWI, Dell'Acqua, Job, Peressotti, & Pascali, 2007; for word frequency and age of acquisition effects, Dent, Johnston, & Humphreys, 2008; for the Stroop-effect, Fagot & Pashler, 1992). For PWI, it was found that the locus of interference was located before the singular resource that both tasks share. The reasoning behind this is that a small interval between the first and the second task

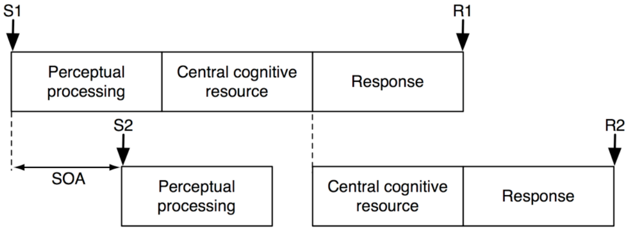


Figure 1: Diagram of the PRP design. The top bar indicates processing of the first task. The bottom bar indicates processing in the second task. S1: stimulus of task 1; S2: stimulus of task 2; R1: response on task 1; R2: response on task 2; SOA: Stimulus Onset Asynchrony

generates a large delay in processing of the second task (referred to as "cognitive slack"), in which the interference that is present in PWI can be resolved. If the interval between the tasks increases, the delay in processing of the second task disappears, and therefore the interference becomes apparent in the reaction times. Following this logic, no congruency effect at small SOAs (but a congruency effect at larger SOAs) would mean a locus before the singular resource, whereas a congruency effect at every SOA would mean a locus after the singular resource. We applied the same reasoning to study which processing stages in a PWI task are affected by cognitive control.

Methods

To study the locus of the Gratton effect in picture-word interference, we re-analyzed the data from a previous experiment (Van Maanen, Van Rijn, & Taatgen, submitted).[1] In this experiment, participants were required to perform a tone classification task and a PWI task concurrently. For the tone classification task, participants had to classify a tone as either low, medium, or high pitch by pressing the b, n, or m keys respectively with the index, middle and ring fingers of the right hand. For the PWI task, participants were required to name an image in which a word was written in the center, and ignore the word. Of each image, three PWI stimuli were created that consisted of the image, with a word written in the center of the image. The words were selected as follows: For the Related condition, category members of the image descriptors were selected. The words for the Unrelated condition were then selected from the CELEX lexical database (Baayen, Piepenbrock, & Van Rijn, 1993), and matched to the related distractors with respect to word length (plus or minus 1 letter) and word frequency (plus or minus 10%). For the Congruent condition, Dutch translations of the most common English picture names were used.

In addition to the Relatedness condition (Related, Unrelated, Congruent), we also manipulated the interval between the tone presentation and the PWI-stimulus presentation (SOA), which could be either 100ms, 350ms, or 800ms. These SOAs were chosen to ensure the PRP effect. Importantly, the correct response order was stressed, to ensure that participants responded to the tone first and to the PWI-stimulus second.

Results

We excluded trials according to the following criteria: Responses that were more than three standard deviations from a participants' mean were excluded (2.1% on the PWI stimulus, and 2.3% on the tone, respectively). Trials in which the responses were in the incorrect order were also excluded (5.3%). Overall, 7.7% of the trials were excluded. In this paper, we will only focus on the effects on the PWI task, and not discuss the effects on the secondary tone classification task.

For each trial, we determined the relatedness between picture and word on the previous trial (Previous). An analysis of variance (ANOVA) showed significant main effects of Relatedness (the congruency effect), and of SOA (the PRP effect), but not of Previous ($F_{\text{Relatedness}}(2,42) = 50$, $p<0.001$; $F_{\text{SOA}}(2,42) = 104$, $p<0.001$; $F_{\text{Previous}}(2,42) = 1.3$, $p=0.28$). However , there was a Relatedness times Previous interaction present ($F_{\text{Relatedness x Previous}}(4,84) = 4.0$, $p=0.005$), representing the Gratton effect. In addition, there was an effect of SOA on the Relatedness condition ($F_{\text{SOA x Relatedness}}(4,84) = 2.5$, $p=0.047$), as well as a significant three-way interaction between SOA, Relatedness, and Previous ($F_{\text{SOA x Relatedness x Previous}}(8,168) = 3.4$, $p=0.001$).

A visual inspection of the data (Figure 2) shows that the three-way interaction appears as a difference in the congruency effect at the small SOAs (100ms and 350 ms) between the trials directly following a Congruent trial ("post-C" in Figure 2) and the trials following a Related trial ("post-R" in Figure 2). Where the post-C trials do not show a congruency effect at small SOAs ($t<1$), the post-R trials do (paired t-test, $t=3.2$, $df=43$, $p=0.002$). The Gratton effect is visible at SOA=800ms as a smaller congruency effect for post-R trials then for post-C trials.

Discussion

The lack of a consistent pattern in the responses on the trials following an Unrelated trial (the post-U trials) can be explained by individual differences in how participants adapt their control. Some participants might treat Unrelated trials similar to Congruent trials (because they are both non-conflicting). Other participants might adapt their control on post-U trials similar to the control in post-R trials, following the similarity between related PWI and unrelated PWI stimuli (both incongruent). A mixture of these two strategies could explain the data found for the post-U trials.

The experiment shows that in PWI, the Gratton effect is present as an interaction between congruency and the previous trial. However, for trials immediately following a Congruent trial, the congruency effect disappears at small SOAs, whereas for trials following a Related trial, the effect remains. Similar observations have been interpreted as a different effect locus (e.g., for Stroop and PWI, Dell'Acqua et al., 2007; for word frequency and age of acquisition effects, Dent, Johnston, & Humphreys, 2008). Therefore,

[1] The submitted manuscript contains an extensive description of the experiment. The manuscript can be downloaded from http://www.ai.rug.nl/~leendert/pubs.

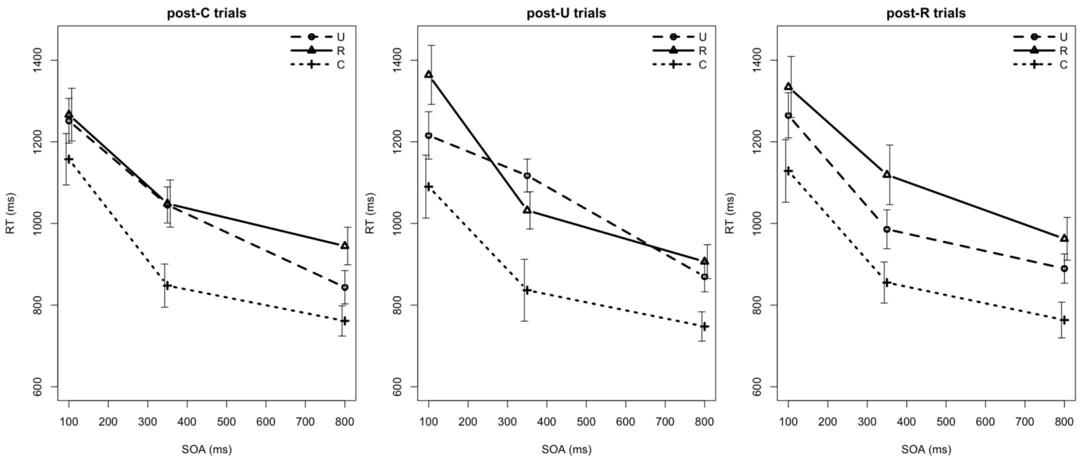


Figure 2: Response times as a function of SOA for the Relatedness conditions. U: Unrelated; R: Related; C: Congruent.

the experiment suggests that the locus of the congruency effect in PWI is influenced by the previous trial. In the following section, we will present a computational cognitive model that accounts for this apparent shift in locus in terms of a difference in processing speed between conditions.

A Cognitive Model of the Gratton Effect

RACE/A

The basis of our computational model of the Gratton effect is a recent model of declarative memory retrieval that we have developed (Van Maanen, 2009; Van Maanen & Van Rijn, 2007; Van Maanen, Van Rijn, & Taatgen, submitted). The model – termed Retrieval by Accumulating Evidence in an Architecture or RACE/A – describes memory retrievals as a sequential sampling process (Ratcliff, 1978). In addition, RACE/A assumes that the dynamics of the retrieval process are constrained by other cognitive processes that co-occur with a particular retrieval process. This aspect is captured by integrating the sequential sampling process in the cognitive architecture ACT-R (Anderson, 2007).

The accumulation process can be characterized by two equations that determine the long-term dynamics and the short-term dynamics of the activation. The short-term dynamics are mediated by the presence or absence of stimuli and spreading activation from other chunks. During a retrieval process, the activation of chunks that match a set of retrieval conditions gradually accumulates until a certain decision criterion (explained below) has been reached. The chunk that has been decided upon is retrieved from declarative memory, and the accumulation of activation stops. Because no new activation is being accumulated, the short-term component of the activation of all chunks decays. The short-term activation dynamics can be represented by a drift, a starting point, and a decision boundary, which will be discussed below.

Drift Drift in RACE/A is the reflection of the current demands of the environment. Thus, drift is a function of stimuli, as well as the currently active declarative facts. All facts and stimuli, which will collectively referred to as *sources of activation*, continuously spread excitatory activation towards associated chunks. This means that a chunk that has more sources of activation (more evidence) or sources with more activation ("stronger" evidence) will accumulate faster than a chunk with less sources of activation or sources with less activation. In the absence of evidence for a particular chunk, the short-term activation will decay. The drift in RACE/A is also determined by a logistically distributed noise sample, adding stochasticity to the system. These considerations are reflected by Equation 1, which may be referred to as the drift equation (Usher & McClelland, 2001). The drift equation captures the dynamics of short-term activation (*C*) of one chunk (chunk *i*) over time.

$$dC_i = [-\alpha C_i + \beta \sum_j S_{ji} A_j + \varepsilon_i]dt \qquad (1)$$

In this equation, the decay of short-term activation is expressed by α, which should be a value in the range [0,1]. The spreading activation component is a sum of the activation of other chunks (A_j), weighted by the associations that exist with chunk i (S_{ji}). Note that this differs from the implementation in ACT-R, in which only the chunks in buffers spread activation. In RACE/A, all chunks may spread activation. The spreading activation component is scaled by a factor β that determines the overall accumulation speed. The noise is expressed by ε_i.

Starting point The starting point of the accumulation reflects the prior probability that a chunk is needed. This is reflected by ACT-R's base-level activation equation (Equation 2, Anderson, 2007), which incorporates the usage history of a chunk. Chunks with a high base-level activation start the accumulation of activation at a higher starting point, and are thus more likely to be retrieved from memory.

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \qquad (2)$$

Given that the usage history of the retrieved chunk has been altered (because it has been retrieved recently), the chunk's long-term component is being increased in such a way that it greatly exceeds the current level of short-term activation. For this reason, the net activation of each chunk in the system can be described as

$$A_i = \max(B_i, C_i) \tag{3}$$

indicating that the activation of a chunk is the maximum of the need probability of a chunk (reflected by B_i) and the accumulating evidence for that chunk (reflected by C_i).

Decision Boundary The decision boundary in RACE/A is relative to the activation of competitors in the system. This choice reflects the insight that if multiple memory representations are relevant, responding becomes more difficult (Hick, 1952; Luce, 1986). This is reflected by Equation 4, which expresses the conditions under which a decision will be made. If the activation of a certain chunk (chunk *i* in Equation 4) exceeds the activation of all competitors (*j*, including *i*) by a certain ratio θ (referred to as the decision ratio), then that chunk is retrieved from memory. The ratio between the activation of one chunk and the summed activation of all competitors reflects the relative likelihood of a chunk, and will be referred to as the Luce ratio for that chunk (Luce, 1963). The duration of the retrieval process constitutes the interval between the onset of the retrieval process (when the request for a retrieval is made) and the moment at which the Luce ratio of one chunk exceeds the decision ratio.

$$\frac{e^{A_i}}{\sum_j e^{A_j}} \geq \theta \tag{4}$$

The Model

The model concurrently performs the tone classification task and the PWI task. The tone classification task was modeled using ACT-R's standard auditory perception module. If a tone is presented, the model processes auditory information, and retrieves a memory trace that encodes the appropriate stimulus-response mapping (that is, which button to press given the perceived tone). Finally, the model made a motor response to press the correct button.

When the PWI-stimulus is presented, the model activates conceptual representations in response to the image, and activates a lemma representation in response to the word (e.g., Roelofs, 1992). Because lemmas spread activation to the conceptual representations that relate to them, the presentation of a distractor word causes interference at the conceptual level. The decision boundary that determines retrieval from memory becomes harder to reach for the conceptual representation of the picture, increasing the retrieval time. The different activation levels of the target chunk versus competing chunks determine the latency difference between the related and unrelated PWI conditions. In the related condition, the concepts of the target and the distractor spread activation to each other. This mutual excitation causes both activation values to increase, making it even harder to reach the decision boundary. In the unrelated condition mutual excitation is not present. Therefore, there is less competition and a faster retrieval in the unrelated than in the related condition.

In order to name the image, the relevant concept has to be retrieved from memory. Once a concept has been retrieved, the model initiates a response, but not before the selection of the appropriate tone-to-button mapping for the tone classification response has been retrieved. This ensures that the model displays cognitive slack time in which interference in the first processing stage may be resolved.

In processing the PWI response, the model first retrieves a lemma representation that encodes the syntactic information associated with the desired response, than it retrieves a motor program to articulate the desired response. Thus, to complete the task the model needs to do three memory retrievals.

Simulation Results

The model is similar to a previous model of a PRP study of PWI (Van Maanen, Van Rijn, & Taatgen, submitted). However, in the current model we manipulated the speed of word processing relative to the speed of picture processing. Following Botvinick et al. (2001) we assumed that a previous conflict trial leads to more cognitive control, leading to more suppression of the reading response. Thus, high control in the model means a low value for the parameter controlling word processing speed. On the other hand, if the previous trial was congruent, we assume a relaxation of control, resulting in less suppression of the reading response and a high value of the parameter that controls word processing speed (low control).

Figure 3 presents the model behavior for high and low control, respectively. Similar to the pattern in the data (Figure 2), the model shows no interference effect for the high control condition (analogous to the post-R trials), whereas it shows an interference effect for the low control condition (analogous to the post-C trials). In our simulations we ignored the post-U condition from the experiment, since we assume that behavior in that condition was a mixture of behavior from post-C trials and post-R trials.

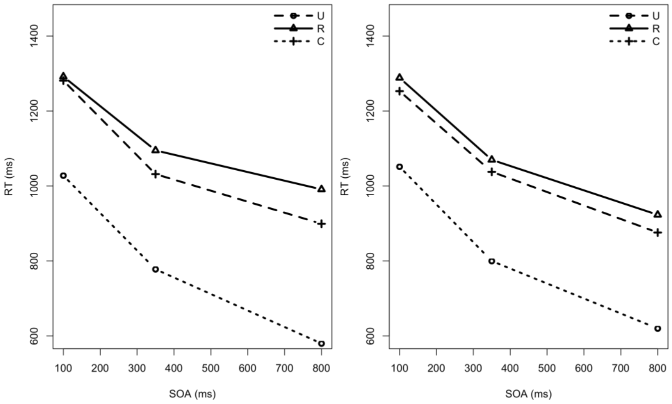


Figure 3: Simulation results for low control (left) and high control condition (right). R: Related; U: Unrelated; C: Congruent.

The explanation for this effect follows directly from the dynamics of the activation of the chunks (conceptual and lemma information) in the model. Retrieval times for the concept and lemma information are determined by the activation ratio (the Luce ratio) between the chunks. Thus, a high Luce ratio in favor of the relevant chunk (the one associated to the correct response) leads to a fast retrieval. A high Luce ratio is reached by a large difference in processing speed fo the two stimulus dimensions, hypothesized to reflect high control (Figure 4, High Control). A high ratio in favor of the irrelevant chunk, or a low ratio in favor of the relevant chunk leads to slower retrievals. The competition between chunks results from mutual excitation of the competing chunks. Therefore, strong competition results in high activation of the competing chunks, and also in a high activation difference (Figure 4, Low Control). As a result, a subsequent retrieval of the same chunk will be faster, because the starting points of accumulation of activation of the competing chunks differ more than at the start of the first retrieval. A similar argument can be constructed for chunks that are strongly associated, as are the concept chunks and lemma chunks in our model. An initial concept retrieval already influences the activation at the start of the subsequent lemma retrieval.

Figure 5 presents the activation dynamics of four chunks in the model over time. the top panel (Low control) presents a prototypical model run in which the word processing speed is high, the bottom panel (High control) presents a model run in which the word processing speed is low. Figure 5 illustrate how a fast retrieval in the first stage of the PWI process may lead to a slow retrieval in the later stages, resulting in a shift of the overall interference pattern.

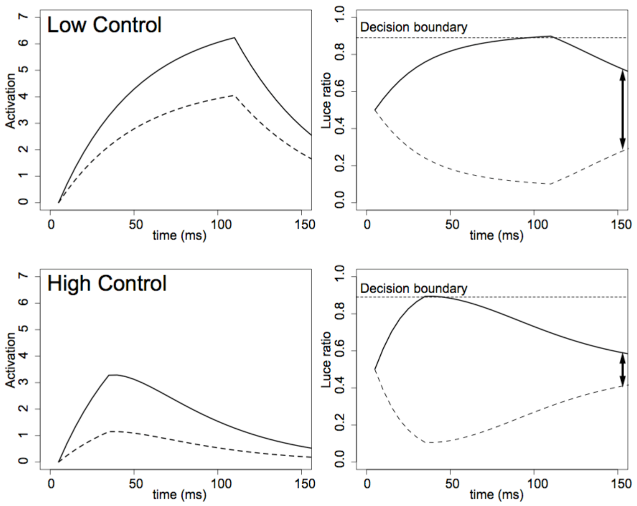


Figure 4: The activation dynamics in RACE/A.

Discussion & Conclusion

Although we implemented the effect of more cognitive control as a lower speed of word processing relative to picture processing, we make no claims on the exact mechanism. Besides actual slower perceptual processing, another possibility could be that more cognitive control results in active inhibition of the undesired response. However, similar to our current implementation this would result in less competition, and our results would not differ.

Analogy with Stroop and PWI

The results from our study show a remarkable analogy with the results from experiments that studied the difference between the Stroop effect and PWI. Dell'Acqua et al. (2007) found an early locus of interference in PWI, similar to the post-C condition in our experiment. By contrast, Fagot and Pashler (1992) found a late locus of interference in the Stroop task, similar to our post-R condition. In a previous study, we explained this difference by a difference in processing speed between colors and images (Van Maanen & Van Rijn, 2008; Van Maanen, Van Rijn, & Borst, submitted). The cognitive models in that study showed that both Fagot and Pashler's data and Dell'Acqua et al.'s data could be explained by one model that maintained a lower processing speed for color information than for picture information.

Our current results suggest that it may not be the speed of perceptual processing per se that is important in shifting the locus of interference, but rather the difference in speed between the two stimulus dimensions (words and pictures for PWI). In the current model, the processing speed of the word and picture differed more for the low control than for

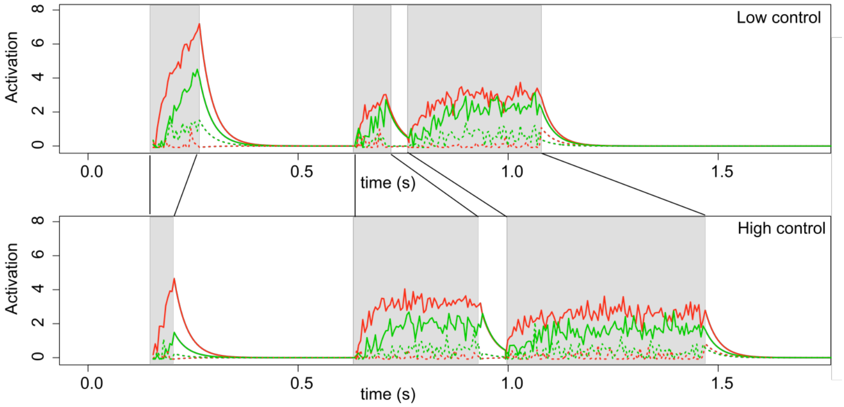


Figure 5: A simulated trial for the low control condition (top) and the high control condition (bottom). The grey areas indicate the duration of every memory retrieval during a trial.

the high control condition. This was explained by more suppression of the reading response in the high-control condition. In the Stroop/PWI model, the processing speed for the PWI condition differed more than for the Stroop condition. This was explained by the faster processing of colors than of pictures, and hence a greater difference in processing speed between words and pictures in PWI than between words and colors in Stroop.

Conclusion

The experiment demonstrated that the Gratton effect is not only present as a difference in interference effect size after Congruent and Related trials, but also entails a shift in the locus of the interference. The absence of observable interference at small SOAs in the post-C trials suggests that the locus of interference in those trials is in an early processing stage, but is absorbed in the cognitive slack time that is created by the PRP design. The presence of observable interference in post-R trials suggest that the locus of interference is late, after the bottleneck that is created by the PRP design.

Our simulations suggest a mechanism for this shift in locus. The simulations show that if the speed with which words are processed is high, the locus of interference is early, whereas a low processing speed for words results in a late locus. The processing speed for words was hypothesized to be under cognitive control, where an increase in control leads to a decrease in word processing speed, and vice versa. These results suggest that the specifics of the stimulus determine the magnitude and spacing of interference over the entire task, a result which may be extended to the Stroop/PWI literature as well.

Acknowledgments

Chris Janssen is acknowledged for suggesting the between-trial analysis that lead to this study.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford UP.

Baayen, R. H., Piepenbrock, R., & Van Rijn, H. (1993). The CELEX lexical database (cd-rom).

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review, 108*(3), 624-652.

Dell'Acqua, R., Job, R., Peressotti, F., & Pascali, A. (2007). The picture-word interference effect is not a Stroop effect. *Psychonomic Bulletin & Review, 14*(4), 717-722.

Dent, K., Johnston, R., & Humphreys, G. (2008). Age of acquisition and word frequency effects in picture naming: A dual-task investigation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 34*(2), 282-301.

Fagot, C., & Pashler, H. (1992). Making two responses to a single object: Implications for the central attentional bottleneck. *Journal of Experimental Psychology: Human Perception and Performance, 18*(4), 1058-1079.

Glaser, W. R., & Düngelhoff, F. J. (1984). The time course of picture-word interference. *Journal of Experimental Psychology: Human Perception and Performance, 10*(5), 640-654.

Gratton, G., Coles, M. G. H., & Donchin, E. (1992). Optimizing the use of information: Strategic control of activation of responses. *Journal of Experimental Psychology: General, 121*(4), 480-506.

Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology, 4*(1), 11-26.

Lovett, M. C. (2005). A strategy-based interpretation of Stroop. *Cognitive Science, 29*(3), 493-524.

Luce, R. D. (1963). Detection and recognition. In R. D. Luce, R. R. Bush & E. Galanter (Eds.), *Handbook of mathematical psychology* (Vol. 1, pp. 103-189). New York: Wiley.

Luce, R. D. (1986). *Response times*. New York: Oxford UP.

MacLeod, C. M., & Dunbar, K. (1988). Training and Stroop-like interference: Evidence for a continuum of automaticity. *Journal of Experimental Psychology-Learning Memory and Cognition, 14*(1), 126-135.

Pashler, H. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116*(2), 220-244.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review, 85*(2), 59-108.

Roelofs, A. (1992). A spreading-activation theory of lemma retrieval in speaking. *Cognition, 42*(1-3), 107-142.

Telford, C. W. (1931). The refractory phase of voluntary and associative responses. *Journal of Experimental Psychology, 14*, 1-36.

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: The leaky, competing accumulator model. *Psychological Review, 108*(3), 550-592.

Van Maanen, L. (2009). *Context effects on memory retrieval: Theory and applications.* Doctoral dissertation, University of Groningen.

Van Maanen, L., & Van Rijn, H. (2007). An accumulator model of semantic interference. *Cognitive Systems Research, 8*(3), 174-181.

Van Maanen, L., & Van Rijn, H. (2008). The picture-word interference effect is a Stroop effect after all. In V. Sloutsky, B. Love & K. McRae (Eds.), *Proceedings of the 30th Annual Meeting of the Cognitive Science Society* (pp. 645-650). Washington DC.

Van Maanen, L., Van Rijn, H., & Borst, J. P. (submitted). Stroop and picture-word interference are two sides of the same coin.

Van Maanen, L., Van Rijn, H., & Taatgen, N. A. (submitted). Accumulators in context: An integrated theory of context effects on memory retrieval.

Verguts, T., & Notebaert, W. (2008). Hebbian learning of cognitive control: Dealing with specific and nonspecific adaptation. *Psychological Review, 115*(2), 518-525.

Welford, A. T. (1967). Single-channel operation in brain. *Acta Psychologica, 27*, 5-22.

The Wisdom of Crowds in Rank Ordering Problems

Brent Miller (brentm@uci.edu)
Pernille Hemmer (phemmer@uci.edu)
Mark Steyvers (mark.steyvers@uci.edu)
Michael D. Lee (mdlee@uci.edu)

Department of Cognitive Sciences
3151 Social Science Plaza
Irvine, CA 92697-5100

Abstract

When averaging the estimates of individuals, the aggregate can often come surprisingly close to the true answer. We are interested in extending this “wisdom of crowds” phenomenon to more complex situations where a simple strategy like taking the mode or mean of responses is inappropriate, or might lead to bad predictions. We report the performance of individuals in a series of ordering tasks, where the goal is to reconstruct from memory the order of time-based events, or the magnitude of physical properties. We introduce a Bayesian version of a Thurstonian model that aggregates orderings across individuals, and compare it to heuristic aggregation techniques inspired by existing models of social choice and voting theory. The Bayesian model performs as well as the heuristics in reconstructing the true ordering, and has the advantage of being well calibrated, in the sense that it gives more confident responses the closer it is to the truth.

Keywords: Bayesian Modeling; Rank Ordering; Consensus; Wisdom of Crowds; Rank aggregation.

Introduction

When Galton first surveyed English fair-goers in 1906, it was a novel curiosity that their estimates of the dressed weight of an ox, when averaged, closely approximated the true weight (Galton, 1907). Subsequently, many demonstrations have shown that aggregating the judgments of a number of individuals often results in an estimate that is close to the true answer. This phenomenon has come to be known as the “wisdom of crowds” (Surowiecki, 2004). The wisdom of crowds idea is currently used in several real-world applications, such as prediction markets (Dani et al., 2006), spam filtering, and the prediction of consumer preferences through collaborative filtering.

Many wisdom of crowds demonstrations have involved situations where a single numerical quantity needs to be estimated. In these cases, a robust estimate of the central tendency of individual estimates can be an effective aggregation method (Yaniv, 1997). Other situations have involved recovering the answers to multiple choice questions. For example, on the game show "Who Wants to be A Millionaire", contestants are given the opportunity to ask all members of the audience to answer a multiple choice question. In this case, an aggregation method based on the modal response can be quite effective. Over several seasons of the show, the modal response of the audience corresponded to the correct answer 91% of the time. More sophisticated approaches have been developed, such as Cultural Consensus Theory (e.g., Romney, Batchelder, Weller, 1987), that additionally take differences across individuals and items into account when aggregating multiple choice answers.

In this paper, we extend the wisdom of crowds idea to the more complex problem of rank ordering. Is it possible to recover the correct order of events or physical properties from a large number of independent individual responses? How confident can we be that these aggregations represent the ground truth?

Aggregating rank order data is not a new problem. In social choice theory, a number of systems have been developed for aggregating rank order preferences for groups (Marden, 1995). Preferential voting systems, where voters explicitly rank order their candidate preferences, are designed to pick one or several candidates out of a field of many. These systems, such as the Borda count, perform well in aggregating the individuals' rank order data, but with an inherent bias towards determining the top members of the list.[1] However, as voting is a means for expressing individual preferences, there is no ground truth. The goal for these systems is to determine an aggregate of preferences that is in some sense “fair” to all members of the group.

Relatively little research has been done on the rank order aggregation problem with the goal of approximating a known ground truth. In follow-ups to Galton's work, Gordon (1924) and Bruce (1935) tested a large number of individuals in psychophysical ordering tasks. They found that the group estimate approximates the ground truth better as the size of the group increases. Interestingly, these authors used the Borda count voting method (without making this connection to voting theory explicit in their work) to aggregate the rank orderings of individuals. Romney et al. (1987) also developed an informal aggregation model for rank order data based on Cultural Consensus Theory, using factor analysis of the covariance structure of rank order judgments. With this, they were able to partially recover the correct order of 34 causes of death in

[1] This is necessary to satisfy the *Condorcet Criterion*, which requires that a top ranked candidate selected by a voting system should be a candidate who has more votes when compared to every other voter on the ballot (Shepsle & Bonchek, 1997)

the US on the basis of the individual orderings of 36 subjects.

We present empirical and theoretical research on the wisdom of crowds phenomenon for rank order aggregation. We conduct an empirical study where people are asked to rank order the occurrence of events (e.g., US presidents by term of office[2]) or the magnitude of some physical property (e.g., rivers by length). Most importantly, no communication between people is allowed for these tasks, and therefore the aggregation method operates on the data produced by independent decision-makers.

Importantly, for all of the problems there is a known ground truth. The ground truth might only be partially known to the tested individuals. If different individuals have knowledge of different parts of the ordering problems, aggregation across individuals can yield a group answer that comes closer to the ground truth than any of the individuals in the group. For example, if some individuals know that the Congo is longer than the Parana River, and other individuals know that the Parana River is longer than the Mekong River, aggregation might lead to the correct overall ordering (i.e., Congo > Parana > Mekong). Therefore, for the wisdom of crowd phenomenon to work, the errors in semantic memory need to have some degree of independence. If all individuals have access to the same knowledge, there will be no advantage to aggregating their answers.

We compare several heuristic computational approaches—based on voting theory and existing models of social choice—that analyze the individual judgments and provide a single answer as output, which can be compared to the ground truth. We refer to these synthesized answers as the "group" answers because they capture the collective wisdom of the group, even though no communication between group members occurred.

We also develop a probabilistic model based on a Thurstonian approach that represents items as distributions on an interval dimension. We make inferences about the parameters of the model using Markov chain Monte Carlo (MCMC). The advantage of MCMC estimation procedure is that it gives a probability distribution over group orderings, and we can therefore assess the likelihood of any particular group ordering. We use this likelihood as a confidence measure to test whether the model is calibrated, in the sense that the group answers with high confidence are close to the ground truth.

Experiment

Method

Participants were 78 undergraduate students at the University of California, Irvine. The experiment was composed of 20 questions (3 were excluded from analysis; one because participants misunderstood the question, one because of the lack of a proper ground truth, and the last for consistency as it only included 5 elements for ordering, whereas all the others included 10). The remaining questions involved general knowledge regarding: population statistics (4 questions), geography (3 questions), dates, such as release dates for movies and books (7 questions), U.S. Presidents, material hardness, the 10 Commandments, and the first 10 Amendments of the U.S. Constitution

All questions had a ground truth obtained from Pocket world in figures and various online sources. An interactive interface was presented on a computer screen. Participants were instructed to order the presented items (e.g., "Order these books by their first release date, earliest to most recent"), and responded by dragging the individual items on the screen using the computer mouse, and "snapping" the item into the desired location in the ordering. Once participants were satisfied with their response they clicked on the submit button. They were prompted to confirm that they wished to proceed before being presented with the next question. Once their response was submitted it was not possible to return to that question. The questions were presented in a fixed order. Half the participants received the forward ordering of questions, the other half received the backwards ordering of questions. The initial ordering of the 10 items within a question was randomized across all questions and all participants.

Results

We first evaluated participants' responses based on whether or not they reconstructed the correct ordering. Table 1 shows the proportion of individuals who got the ordering exactly right (PC) for each of the ordering task questions. On average, about one percent of participants recreated the correct rank ordering perfectly. We also analyzed the performance of participants with a more fine-grained measure, using Kendall's τ distance. This distance metric is used to count the number of pair-wise disagreements between the reconstructed and correct ordering. The larger the distance, the more dissimilar the two orderings are.

Table 1: *Participant performance statistics.*

		Percentiles of τ				
Problem	PC	25	50	75	90	100
books	0.000	15	10	8	5	3
city population europe	0.000	19	15	12	10	7
city population us	0.000	20	14	11	8	6
city population world	0.000	23	18	15	12	5
country landmass	0.000	12	9	7	5	2
country population	0.000	17	15	11	9	4
hardness	0.000	18	15	12	11	7
holidays	0.051	12	8	5	3	0
movies releasedate	0.013	9	6	4	2	0
oscar bestmovies	0.013	14	10	6	4	0
oscar movies	0.000	16	10	5	2	1
presidents	0.064	10	7	3	1	0
rivers	0.000	19	15	13	11	3
states westeast	0.026	10	6	3	1	0
superbowl	0.000	24	17	14	11	6
ten ammendments	0.013	19	13	10	4	0
ten commandments	0.000	23	17	11	7	1
AVERAGE	0.011	16.5	12.1	8.8	6.2	2.6

[2] The ordering of US Presidents has been studied before in the context of memory research by Healy, Havas, and Parker (2000).

Values of τ range from: $0 \leq \tau \leq N(N-1)/2$, where N is the number of items in the order (10 for all of our questions). A value of zero means the ordering is exactly right, and a value of one means that the ordering is correct except for two neighboring items being transposed, and so on up to the maximum possible value of 45.

Table 1 shows the distribution of τ values over the ranked population of participants for each of the 17 sorting task questions, in terms of values at the 25th, 50th, 75th, 90th and 100th percentiles. For six of the questions, one or more participants get the ordering exactly right, as indicated by a τ of 0 for the 100th percentile. The best individuals on each question achieve good performance, and solve the problem exactly, or are within a few pair transposes, for most questions. As this is a prior knowledge task, it is interesting to note the best performance overall was achieved on the *Presidents*, *States from west to east*, *Oscar movies*, and *Movie release dates* tasks. These four questions relate to educational and cultural knowledge that seems most likely to be shared by our undergraduate subjects.

Modeling

We evaluated a number of heuristic aggregation models and compared the performance of these methods against a probabilistic model based on a Thurstonian approach. For each model, the set of orderings from individuals is analyzed in order to create a single group ordering, which is then compared to the ground truth.

Heuristic Models

We tested four heuristic aggregation models. The simplest heuristic, based on the mode, has been used since the earliest rank order experiments (Lorge et al. 1957). For this heuristic, the group answer is based on the most frequently occurring sequence of all observed sequences. In cases where several different sequences correspond to the mode, a randomly chosen modal sequence was picked.

The second method, which we refer to as the "greedy count", counts the number of participants responses for each item in each position. The item and the position with the largest agreement among participant is selected first. The selection of items then proceeds in a greedy algorithm fashion, making sure that each item and position is not already filled.

The third method takes the group answer as the participant ranking that is "closest", as determined by a distance measurement metric, to the rankings of all participants. This is a variation of a Kemeny scheme (see Dwork et. al. 2001) where we restrict ourselves to the user-submitted responses. It is implemented here by finding the participant ordering that has the smallest distance, measured by the sum of Kendall's τ's between strings, to the orderings of all other participants. Note that we restrict ourselves to finding a ranking from the *existing* set of participants' responses. This method can be extended to find *any* arbitrary rank order that is closest to the "middle" of observed rankings, but that approach suffers from well-known computational complexity problems.

The fourth method uses the Borda count method, a widely used technique from voting theory. In preferential voting systems, voters express their candidate choices in terms of an ordering of all ballot candidates. In the Borda count method, weighted counts are assigned such that the first choice "candidate" receives a count of N (where N is the number of candidates), the second choice candidate receives a count of N-1, and so on. These counts are summed across candidates and the candidate with the highest count is considered the "most preferred". Here, we use the Borda count to create an ordering over all items by ordering the Borda counts.

Table 2 reports the performance of all of the aggregation models. For each, we checked whether the inferred group order is correct (C) and measured Kendall's τ. We also

Table 2: *Performance of the four heuristic models and the Thurstonian model*

	Kemeny Scheme			Thurstonian Model			Borda Counts			Greedy Count			Mode		
Problems	C	τ	Rank	C	τ	Rank	C	τ	Rank	C	τ	Rank	C	τ	Rank
books	0	4	96	0	6	88	0	7	82	0	7	82	0	12	40
city population europe	0	11	81	0	11	81	0	11	81	0	13	69	0	17	42
city population us	0	10	87	0	11	79	0	12	67	0	9	90	0	16	45
city population world	0	18	59	0	16	73	0	15	77	0	16	73	0	19	44
country landmass	0	7	76	0	5	95	0	5	95	0	5	95	0	7	76
country population	0	11	82	0	11	82	0	11	82	0	13	67	0	15	53
hardness	0	11	91	0	11	91	0	11	91	0	18	31	0	15	46
holidays	0	5	77	0	4	78	0	4	78	0	4	78	1	0	100
movies releasedate	0	2	95	0	2	95	0	2	95	0	2	95	0	2	95
oscar bestmovies	0	3	97	0	4	90	0	3	97	0	5	90	0	3	97
oscar movies	0	2	96	0	1	100	0	2	96	0	3	88	0	2	96
presidents	0	1	94	0	2	87	0	3	79	0	1	94	1	0	100
rivers	0	11	91	0	12	86	0	11	91	0	13	77	0	16	42
states westeast	0	1	97	0	2	88	0	3	78	0	1	97	0	1	97
superbowl	0	10	96	0	12	88	0	10	96	0	15	71	0	19	40
ten amendments	0	2	97	0	4	95	0	5	90	0	4	95	0	4	95
ten commandments	0	11	82	0	11	82	0	12	74	0	12	74	0	17	51
AVERAGE	0.0	7.1	87.9	0.0	7.4	86.9	0.0	7.5	85.2	0.0	8.3	80.4	0.1	9.7	68.2

report in the Rank column the percentage of participants who perform worse or the same as the group answer, as measured by τ. With the Rank statistic, we can verify the wisdom of crowds effect. In an ideal model, the group heuristic should perform as well as or better than all of the individuals in the group. Table 2 shows the results separately for each problem, and averaged across all the problems.

These results show that the mode heuristic leads to the worst performance overall in rank. On average, the mode is as good or better of an estimate than 68% of participants. This means that 32% of participants came up with better solutions individually. This is not surprising, since, with an ordering of 10 items, it is easily possible that only a few participants will agree on the ordering of items. The difficulty in inferring the mode makes it an unreliable method for constructing a group answer. This problem will be exacerbated for orderings involving more than 10 items, as the number of possible orderings grows combinatorially. The greedy count heuristic performs better than the mode overall, but it does not lead to the correct answer for any individual problem.

The Borda count and Kemeny scheme perform relatively well in Kendall's τ and overall rank performance measurements. On average, these methods perform with ranks of 85% and 88% respectively, indicating that the group answers from these methods score amongst the best individuals, although 10% of individuals still perform better.

A Thurstonian Model

Despite comparable statistical performances, the heuristic aggregation models create no explicit representation of each individual's working knowledge. Therefore, even though the methods can aggregate the individual pieces of knowledge across individuals, they cannot explain *why* individuals rank the items in a particular way, or how much confidence should be placed in the overall group ranking. To address this potential weakness, we develop a simple probabilistic model based on the seminal Thurstonian approach. Although the Thurstonian approach has often been used to analyze preference rankings (see Marden, 1997 for an overview), it has not been applied, as far as we are aware, to ordering problems where there is a ground truth.

In the Thurstonian approach, the overall item knowledge for the group is represented explicitly as a set of coordinates on an interval dimension. The interval representation is justifiable given that all the problems in our study involve one-dimensional concepts (e.g., the relative timing of events, or the lengths of items). Specifically, each item is represented as a value μ_i along this dimension, where $i \,\epsilon \,\{1, \ldots, N\}$. Each individual is assumed to have access to the group-level information. We assume, however, that individuals do not have precise knowledge about the exact location of each item. We model each individual's location of the item by a single sample from a distribution, centered on the item's group location. We represent the uncertainty associated with this value, μ_i, with a Normal distribution, $\mathrm{N}(\mu_\mathrm{i}, \sigma_\mathrm{i})$. In a fully specified Thurstonian model, once an individual draws samples for each item, the ordering for that individual is based on the ordering of the samples. Figure 1 shows an example of the group-level information for six items, A to G. A particular individual might sample values from these distributions such that some items are ranked correctly, but other items are transposed. In Figure 1, there is a larger degree of uncertainty for item C, making it likely that item C is placed incorrectly in the ordering.

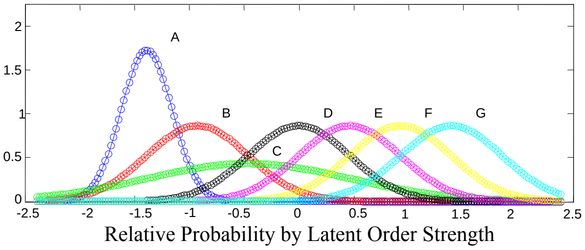
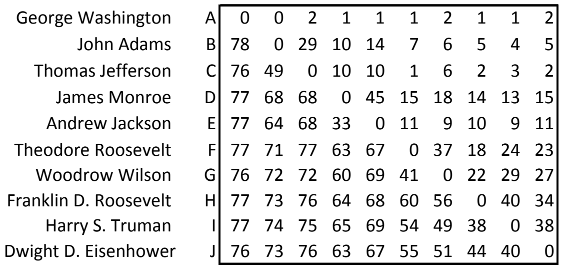


Figure 1. Example of group-level information for six items.

We apply Bayesian estimation techniques to infer the group representation from the individual orderings. Bayesian methods have been applied to Thurstonian models before (Yao, & Böckenholt, 1999), but here we present a simplified version of the Thurstonian model that facilitates more efficient Bayesian inference.

In the simplified model, we do not attempt to explain the particular orderings for each individual, but rather the pairwise orderings across all individuals. The data for this model consist of a N x N count matrix R, where $\mathrm{R}(i, j)$ contains the number of participants who ordered item i later than item j. For example, Figure 2 shows the matrix for the *Presidents* question with the Presidents in the correct order. Note that nearly all of the 78 participants correctly place George Washington earlier than any of the other Presidents, but that Dwight D. Eisenhower, who should be ranked last, is often placed earlier than other Presidents. The pairwise data therefore indicate some uncertainty about the ranking of Eisenhower relative to other Presidents.

In our model, when determining the relative order of two items i and j, a person samples a value from item i, $x_i \sim \mathrm{N}(\mu_i, \sigma_i)$, and also a value from item j, $x_j \sim \mathrm{N}(\mu_j, \sigma_j)$. These values are then compared to each other and item i is



		A	B	C	D	E	F	G	H	I	J
George Washington	A	0	0	2	1	1	1	2	1	1	2
John Adams	B	78	0	29	10	14	7	6	5	4	5
Thomas Jefferson	C	76	49	0	10	10	1	6	2	3	2
James Monroe	D	77	68	68	0	45	15	18	14	13	15
Andrew Jackson	E	77	64	68	33	0	11	9	10	9	11
Theodore Roosevelt	F	77	71	77	63	67	0	37	18	24	23
Woodrow Wilson	G	76	72	72	60	69	41	0	22	29	27
Franklin D. Roosevelt	H	77	73	76	64	68	60	56	0	40	34
Harry S. Truman	I	77	74	75	65	69	54	49	38	0	38
Dwight D. Eisenhower	J	76	73	76	63	67	55	51	44	40	0

Figure 2. Count matrix R for the 'Presidents' question.

ranked above j whenever $x_i > x_j$. Let θ_{ij} represent the probability of the outcome $x_i > x_j$. This probability can be determined exactly:

$$\theta_{ij} = p(x_i > x_j) = \Phi\left((\mu_i - \mu_j)/\sqrt{\sigma_i^2 + \sigma_j^2}\right), \tag{1}$$

where Φ is the cumulative normal distribution. This sampling process is repeated for each individual and all item pairs. Therefore, the number of times that item i is ranked before item j, across all individuals, is based on the binomial distribution:

$$\mathrm{R}_{ij} \sim \mathrm{B}(\theta_{ij}, K), \tag{2}$$

where K is the number of individuals.

In this probabilistic model μ_i and σ_i are the latent variables that can be estimated on the basis of the observed data R.[3] We applied MCMC techniques to estimate the latent parameters using a sequence of Metropolis Hasting steps. In order to prevent a drift in the items during estimation (as there is no natural zero point), we fixed the minimum of μ_i to 0 and the maximum of μ_i to 1. We ran 20 chains with a burn-in of 200 iterations. From each chain, we drew 20 samples with an interval of 10 iterations. In total, we collected 400 samples. To construct a single group answer, we analyzed the ordering of the items according to μ_i, separately for each sample, and then picked the mode of this distribution. This corresponds to the most likely order in the distribution over orders inferred by the model.

The result of this Thurstonian model is shown in Table 2. The model performs approximately as well as the Borda count method, but not quite as well as the Kemeny scheme. The model does not recover the exact answer for any of the 17 problems, based on the knowledge provided by the current 78 participants. It is possible that a larger sample size is needed in order to achieve perfect reconstructions of the ground truth.

Visualization of Group Knowledge One advantage of the Thurstonian approach is that it allows a visualization of group knowledge not only in terms of the order of items, but also in terms of the uncertainty associated with each item on the interval scale. Figure 3 shows the inferred distributions for four problems where the model performed relatively well. The crosses correspond to the mean of μ_i across all samples, and the error bars represent the standard deviations σ_i based on a geometric average across all samples.

These visualizations are intuitive, and show how some items are confused with others in the group population. For instance, nearly all participants were able to identify George Washington as the first President of the U.S., but many confused later Presidents whose terms occurred close to each other. Likewise, there was a large agreement on the proper placement of the right to bear arms in the amendments question — this amendment is often popularly referred to as “the second amendment”.

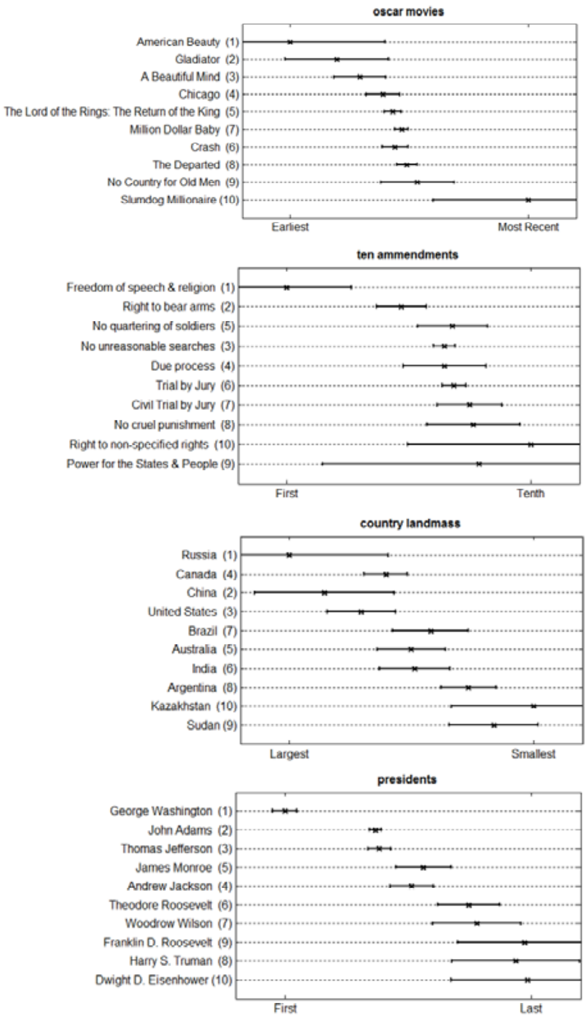


Figure 3. Sample Thurstonian inferred distributions. The actual order is the ground truth ordering, while the numbers in parentheses show the group answer.

Model Calibration Since the probabilistic model is estimated with MCMC techniques, we derive a posterior distribution over all group orderings, from which we select the mode as the best group answer. Because of this, we can also assess the posterior probability of this group answer. This probability has a natural interpretation as the model's measure of confidence. If the distribution over orders is very peaked, most posterior probability is concentrated on the modal answer, indicating a high confidence. If, on the other hand, the model is uncertain about any of the orderings, a low posterior probability, and therefore a low confidence, is given to the modal answer. We can then use this confidence measure to assess to what extent the model is calibrated. That is, we can ask: do confident answers come close to the ground truth?

Figure 4 shows an ordering of the problems according to their confidence values (i.e., the posterior probability of the

[3] Because of the simplified nature of the model, there is no need to explicitly estimate the particular draws *x*. These have been integrated out of the model by virtue of Equation (1)

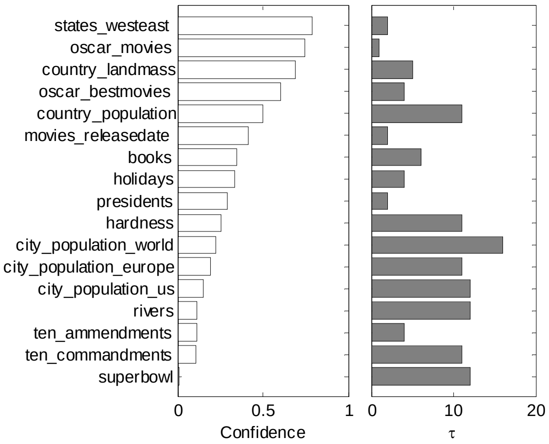


Figure 4. The relation between the confidence in the group answer and the Kendall τ distance of the group answer to the true answer.

modal answer). The right panel shows the Kendall τ distance between the group answer and the true answer. The correlation between confidence and Kendall τ is -.63, showing the expected relationship: high confidence responses are associated with orderings that are closest to the correct ordering. Calibration is important because, in practical situations, the ground truth is not available and a decision maker need to know how confident to be in the aggregated group answer.

Conclusion

We have presented four heuristic aggregation approaches, as well as a Thurstonian approach, for the problem of aggregating rank orders to uncover a ground truth. The model comparison showed that the mode is not a reliable approach for extracting the ground truth, because few individuals agree on the same ordering. We expect that in larger ordering tasks, involving more than 10 items, there might be no individuals that agree with any other on the item ordering. The other heuristic methods, such as the greedy count and the Borda count, analyze the orderings locally by counting the number of times items each occur at each position. This strategy seems to overcome some of the problems with using the mode. The Kemeny scheme extracted a group answer by finding an existing answer in the data that had the smallest combined distance to all other answers, as measured by Kendall's τ. This result suggests that the idea of finding "prototypical" orderings can lead to effective group answers.

We also presented a Bayesian model based on the classic Thurstonian approach. While this model did not outperform the heuristic models, it did perform well, and has some advantages over the heuristic models. The Bayesian model not only extracts a group ordering, but also a representation of the uncertainty associated with the ordering. This can be visualized to gain insight into mental representations and processes. The MCMC estimation procedure used for the Bayesian model leads naturally to a distribution over orderings. This distribution can be used to measure the confidence in any particular group answer. We found that this confidence relates to how close the group answer is to the true answer. Additionally, although not explored here, the Bayesian approach potentially offers advantages over heuristic approaches because the probabilistic model can be easily expanded with additional sources of knowledge, such as confidence judgments from participants and background knowledge about the items.

References

Bruce, R. (1935). Group Judgments in the Fields of Lifted Weights and Visual Discrimination. *The Journal of Psychology*, 1, 117-121.

Dani, V., Madani, O., Pennock, D.M., Sanghai, S.K., Galebach, B. (2006). An Empirical Comparison of Algorithms for Aggregating Expert Predictions. *Proceedings of the Conference on Uncertainty in Artificial Intelligence* (UAI).

Dwork, C., Kumar, R., Noar, M., Sivakumar, D. (2001). Rank aggregation methods for the Web. *Proceedings of the Tenth international conference on the World Wide Web* (pp. 613-622). New York, NY: Association for Computing Machinery.

Galton, F. (1907). Vox Populi. *Nature*, 75, 450-451.

Gordon, K. (1924). Group Judgments in the Field of Lifted Weights. *Journal of Experimental Psychology*, 7, 398-400.

Healy, A. F., Havas, D. A., Parker, J. T. (2000). Comparing Serial Position Effects in Semantic and Episodic Memory Using Reconstruction of Order Tasks. *Journal of memory and language*, 42, 147-167.

Lorge, I, Fox, D., Davitz, J., Brenner, M., (1957). A Survey of Studies Contrasting the Quality of Group Performance and Individual Performance. *Psychological Bulletin*, 55, 337-372.

Romney, K. A., Batchelder, W. H., Weller, S. C. (1987). Recent Applications of Cultural Consensus Theory. *American Behavioral Scientist*, 31, 163-177.

Marden, J. I. (1995). *Analyzing and Modeling Rank Data*. New York, NY: Chapman & Hall USA.

Shepsle, K. A., Bonchek, M. S. (1997*). Analyzing Politics*. New York, NY: Doubleday

Surowiecki, J. (2004*). The Wisdom of Crowds*. New York, NY: W. W. Norton & Company, Inc.

Yaniv, I. (1997). Weighting and trimming: Heuristics for Aggregating Judgments under Uncertainty. *Organizational behavior and human decision processes, 69*(3), 237-249.

Yao, G., & Böckenholt, U. (1999). Bayesian estimation of Thurstonian ranking models based on the Gibbs sampler. *British Journal of Mathematical and Statistical Psychology, 52*, 79–92.

Compound effect of probabilistic disambiguation and memory retrievals on sentence processing: Evidence from an eye-tracking corpus

Umesh Patil (umesh.patil@gmail.com)
Department of Linguistics, Karl-Liebknecht Str. 24–25
14476 Potsdam, Germany

Shravan Vasishth (vasishth@uni-potsdam.de)
Department of Linguistics, Karl-Liebknecht Str. 24–25
14476 Potsdam, Germany

Reinhold Kliegl (kliegl@uni-potsdam.de)
Department of Psychology, Karl-Liebknecht Str. 24–25
14476 Potsdam, Germany

Abstract

We evaluate the predictions of surprisal and cue-based theory of sentence processing using an eye-tracking corpus, the Potsdam Sentence Corpus. Surprisal is a measure of processing complexity based on a probabilistic grammar and is computed in terms of the total probability of structural options that have been disconfirmed at each input word. The cue-based theory characterizes processing difficulty in terms of working memory costs that derive from decay and interference arising during content-based retrieval requests of previously processed material (e.g., to incrementally build the sentence structure). We show that both surprisal and cue-based parsing independently explain difficulty in sentences processing and interestingly, they have an over-additive effect on processing when combined together.

Keywords: Sentence processing; eye-tracking; cue-based theory; surprisal; memory retrievals

Introduction

Research in psycholinguistics provides much evidence for probabilistic disambiguation in human language processing at various levels including lexical, syntactic and semantic processing (Jurafsky, 1996, 2003). More frequent words and structures are easier to comprehend than less frequent ones. Surprisal (Hale, 2001) is a proposal which characterizes processing difficulty in terms of the amount of work done in probabilistically disconfirming sentence continuations as a consequence of the information supplied by the current word. Consider, for example, the famous garden path sentence in (1). It has been observed that English speakers hearing this sentence have great difficulty at ”fell”. Hale (2001) demonstrates using probabilistic context-free grammar that the difficulty occurs because at ”fell” the parser has to disconfirm alternatives that together comprise a great amount of the probability mass.

(1) The horse raced past the barn *fell*.

Recent research in computational models of sentence comprehension has shown that surprisal is a significant predictor of eye movements while reading individual sentences and text (Boston, Hale, Kliegl, Patil, & Vasishth, 2008; Demberg & Keller, 2008). However, surprisal is likely to furnish only part of the explanation (Levy, 2008). As Lewis (1996) and Gibson (2000) argue, sometimes people take longer to process words that they need to connect to other words processed earlier. Resolving these linguistic relations seems to impose more processing effort even when the constructions are frequent or unsurprising. Grodner and Gibson (2005) provide evidence using self-paced reading study which involved reading sentences like (2) below. They observed monotonically increasing reading time at the verb ”supervised” as a function of its distance from the subject ”nurse”.

(2) a. The nurse *supervised* the ...
b. The nurse from the clinic *supervised* the ...
c. The nurse who was from the clinic *supervised* the ...

This difference between surprisal and integration cost was addressed by Demberg and Keller (2008), who compared the predictions of surprisal with Gibson’s (2000) Dependency Locality Theory (DLT), a theory of integration difficulty. They found that DLT’s predictions played a limited role in explaining processing difficulty. DLT was a significant predictor only for reading times at nouns and verbs. Here we show that surprisal and retrieval costs unequivocally play a role in determining processing difficulty. More interestingly, we observed a significant interaction of surprisal and memory retrievals, suggesting that a simple additive model of surprisal and retrieval processes will not suffice.

We compared surprisal’s predictions to the cue-based retrieval model of (Lewis & Vasishth, 2005) (LV05 henceforth) using the Potsdam Sentence Corpus (PSC) of German (Kliegl, Nuthmann, & Engbert, 2006). The cue-based retrieval theory characterizes processing difficulty in terms of working memory costs that derive from decay and interference arising during content-based retrieval requests of previously processed material, e.g., to complete dependencies, or to incrementally build structure.

We implemented cue-based retrieval models for sentences from the PSC, closely following the approach taken by LV05 and generated predictions for retrieval cost at each word. We also computed surprisal's predictions using a probabilistic phrase-structure parser. The main findings are that (1) retrieval cost furnishes better models of eye-fixation measures than models based on baseline predictors such as unigram and bigram frequency, word length, Cloze predictability plus surprisal, and (2) surprisal and retrieval cost show a significant interaction in predicting reading times.

Surprisal

Surprisal offers a theoretical reason why a particular word in a sentence should be easier or more difficult to comprehend on the basis of underlying probabilistic grammatical knowledge of the language. The idea of surprisal is to model processing difficulty as a logarithmic function of the probability mass eliminated by the most recently added word. This number is a measure of the information value of the word just seen, as rated by the grammar's probability model; it is nonnegative and unbounded. More formally, the surprisal of the n^{th} word (w_n) in a sentence is defined as the log-ratio of the prefix probability before seeing the word, compared to the prefix probability after seeing it. The prefix probability at word w_n is defined as the total probability of all grammatical analyses that derive the prefix string $w = w_1 \cdots w_n$ which is initial part of the bigger string wv. For grammar G and a set of derivations D the prefix probability α_n at word w_n can be expressed as:

$$prefix\_probability(w,G) = \sum_{d \in D(G,wv)} probability(d) \; = \; \alpha_n$$

Then, the surprisal at w_n is:

$$surprisal(w_n) = \log_2(\frac{\alpha_{n-1}}{\alpha_n})$$

Intuitively, surprisal and hence the difficulty of processing increases when a parser is required to build some low-probability structure.

Cue-based theory

The cue-based theory of sentence processing is derived from the application of independently motivated principles of memory and cognitive skills to the specialized task of sentence parsing. As a result, sentence processing emerges as a series of skilled associative memory retrievals modulated by similarity-based interference and fluctuating activation. The corresponding parsing model is implemented in the cognitive architecture ACT-R (Anderson et al., 2005) which formalizes the cognitive principles mentioned above in terms of the following set of equations:

1. The base activation (B_i) of chunk i, where t_j is the time since the j^{th} retrieval of the item, d is the decay parameter, and the summation is over all n retrievals, is

$$B_i = ln(\sum_{j=1}^{n} t_j^{-d})$$

2. Total activation (A_i) of a chunk i is defined as the summation of its base activation and strength of association. W_j is the amount of activation from the elements j in the goal buffer and S_{ji}s are the strengths of association from elements j to chunk i

$$A_i = B_i + \sum_j W_j S_{ji}$$

3. S_{ji} is defined in terms of fan_j which is the number of items associated with j

$$S_{ji} = S - ln(fan_j)$$

4. Retrieval latency of chunk i is defined in terms of A_i and F, a scaling constant

$$T_i = Fe^{-A_i}$$

The cue-based retrieval theory quantifies the processing difficulty at each word in terms of its *attachment time*, which is the sum of (i) the time required to retrieve the currently-built syntactic structure in order to attach the word into that structure, and (ii) a baseline cost of 100 milliseconds, which is the time required for the execution of the retrieval request and the subsequent attachment of the current word into the existing structure. See LV05 for details about data structures and the parsing algorithm used.

To summarize, the delay in retrieval of a prior syntactic element due to similarity based interference and fluctuating activation is assumed to induce difficulty in processing.

Experiment

The experiment involved a quantitative evaluation of the predictions of surprisal and cue-based theory using a corpus of eye movements during reading single sentences.

Methods

Data For the analyses in this paper, we selected 32 sentences from the Potsdam Sentence Corpus (PSC), which is an eye-tracking corpus consisting of fixation durations recorded from 222 persons, each reading 144 German sentences (Kliegl, Nuthmann, & Engbert, 2006). These 32 sentences were selected in a way that enabled us to cover a wide range of syntactic structures.

For generating surprisal values for each word in these selected sentences we used a probabilistic context-free phrase-structure parser from Levy (2008), which is an implementation of Stolcke's Earley parser (Stolcke, 1995). We unlexicalized the parser to avoid overlap of surprisal's predictions with the word frequency effect.

We hand-crafted an ACT-R model for each selected sentence, closely following the approach taken by LV05. The model of each sentence was run for 30 simulations and a prediction of attachment time for every word was generated by averaging across all simulations. All ACT-R parameter values were kept the same as those used by LV05 except for activation noise. In LV05, five out of six simulations were carried out without switching on the activation noise. They also noted from preliminary experiments that adding activation noise did not change their results significantly. Since, one of ACT-R's standard assumptions is that there is always some noise added to the activation value of a chunk at each retrieval which permits modeling various kinds of memory errors, we set its value to 0.45 (this was one of the values used in Vasishth, Bruessow, Lewis, & Drenhaus, 2008).

Statistical Analyses The statistical analyses were carried out using linear mixed-effects models (Bates & Sarkar, 2007; Gelman & Hill, 2007) and the *Deviance Information Criterion* or DIC (Gelman & Hill, 2007, 524–527) was used to compare the relative goodness of fit between simpler and complex models. Linear models were fit for the following "early" and "late" eye movements measures:

SFD - fixation duration on a word during first pass if it is fixated only once

FFD - time spent on a word, provided that the word is fixated during the first pass

FPRT - the sum of all fixations on a word during the first pass

TRT - the sum of all fixations

FPSKIP - the probability of skipping the word during the first pass

We considered following baseline predictors in addition to surprisal and attachment time:

unigram - logarithm of token frequency of a word in Das Digitale Wörterbuch der deutschen Sprache des 20. Jahrhunderts (DWDS) (Geyken, 2007; Kliegl, Geyken, Hanneforth, & Würzner, 2006)

bigram - logarithm of the conditional likelihood of a word given its left neighbor in DWDS (also called transitional probability)

word length - number of characters in conventional spelling

predictability - empirical predictability as measured in a Cloze task with human subjects (Taylor, 1953; Ehrlich & Rayner, 1981; Kliegl, Grabner, Rolfs, & Engbert, 2004)

Sentences and participants were treated as partially crossed random factors; that is, we estimated the variances associated with differences between participants and differences between sentences, in addition to residual variance of the dependent measures. For the analysis of FPSKIP (coded as a binary response for each word: 1 signified that a skipping occurred at a word, and 0 that it did not), we used a generalized linear mixed-effects model with a binomial link function (Bates & Sarkar, 2007; Gelman & Hill, 2007).

For each reading time analysis reported below, reading times more than three standard deviations away from the mean were removed before the analyses, excluding at most 1.7% of the data. Attachment time and all dependent measures except FPSKIP were log transformed. Word length, surprisal and attachment time were centered in order to render the intercept of the statistical models easier to interpret.

In the initial analyses, as expected, we found collinearity among the baseline predictors. Since collinearity can inflate the estimates of coefficients' standard errors leading to unreliable results, and can also lead to uninterpretable coefficient values, removal of collinearity between predictors was crucial before fitting the linear models for different fixation measures. For removing collinearity, we incrementally regressed each of these predictors against one or more baseline predictors and used residuals of the regressions as the predictors in the subsequent linear models. This was done in the following three steps:

1. Regression of unigram frequency against word length-
   uni.res = residuals (unigram ~ length)

2. Regression of bigram frequency against word length and residual unigram values obtained from step 1-
   bi.res = residuals (bigram ~ length + uni.res)

3. Regression of predictability against word length, residual unigram and bigram obtained from step 1 & 2-
   pred.res = residuals (predictability ~ length + uni.res + bi.res)

As a result, we had four baseline predictors — length, uni.res, bi.res, pred.res — which were completely noncollinear.

Table 1: Linear model coefficients, standard errors and t-values for surprisal, attachment time and interaction of attachment time and surprisal. An absolute t-value of 2 or greater indicates statistical significance at $\alpha = 0.05$.

	Coef	SE	t-value
SFD			
surprisal	0.021722	0.001195	18
att. time	0.084338	0.013722	6
att. time:surprisal	0.048706	0.009518	5
FFD			
surprisal	0.018304	0.001032	18
att. time	0.062361	0.012361	5
att. time:surprisal	0.039307	0.008327	5
FPRT			
surprisal	0.021520	0.001217	18
att. time	0.056154	0.014221	4
att. time:surprisal	0.050750	0.009743	5
TRT			
surprisal	0.028558	0.001389	21
att. time	0.058249	0.016197	4
att. time:surprisal	0.055988	0.011128	5

Table 2: Linear model coefficients, standard errors and t-values for baseline predictors for TRT.

	Coef	SE	t-value
TRT			
length	0.031052	0.000949	33
uni.res	-0.023228	0.002322	-10
bi.res	-0.011984	0.000879	-14
pred.res	-0.006162	0.002752	-2

Table 3: Linear model coefficients, standard error, z-scores and p-values with FPSKIP as the dependent measure.

	Coef	SE	z-score	p-value
att. time	-0.51588	0.09401	-5.5	<0.001
surprisal	-0.18235	0.01000	-18.2	<0.001
att. time:surp	-0.12521	0.08067	-1.6	0.121

Table 4: Deviance Information Criterion values for simpler model (baseline predictors + surprisal) vs. more complex model (simpler model + attachment time).

	Simpler model	Complex model
SFD	8624.7	8576.5
FFD	9908.0	9873.1
FPRT	22606.0	22581.9
TRT	30695.5	30674.6
FPSKIP	36140.8	36111.5

Results & Discussion

The results of the mixed-effects models are summarized in tables 1 to 3. We observed significant main effects of both surprisal and attachment cost across ”early” as well as ”late” measures and also on FPSKIP. The coefficient for FPSKIP is negative reflecting the fact that the probability of fixating a word increases with increase in surprisal and retrieval cost. These results illustrate that surprisal as well as retrieval cost can account for variance in eye-tracking measures independent of baseline predictors (such as unigram and bigram frequency, word length, Cloze predictability, etc.). For comparison, coefficients of baseline predictors for TRT are listed in table 2; similar coefficient values were obtained for other reading time measures.

The interaction of attachment time and surprisal is significant for all measures except for FPSKIP (though even in this case the coefficient has the expected sign), which indicates that there is a disproportionate increase in reading difficulty when both surprisal and retrieval cost are high.

Table 4 compares the DIC values for simpler models (baseline predictors + surprisal) and complex models (baseline predictors + surprisal + attachment time). For all dependent measures the predictive error (DIC value) was lower in the more complex model that included attachment time, which means that the complex models should be preferred to the simpler ones.

Retrieval cost, surprisal and their interaction show effects on ”early” as well as ”late” measures. This suggests that structure-building and retrieval processes start very soon after lexical access begins.

Implications for eye movement models Besides the contribution to psycholinguistic theories, this work can contribute towards extending models of eye movement control such as E-Z Reader (Pollatsek, Reichle, & Rayner, 2006) and SWIFT (Engbert, Nuthmann, Richter, & Kliegl, 2005) which despite being the two most fully developed models of eye movements, do not incorporate any theory of language processing. The latest version of E-Z Reader (Reichle, Warren, & McConnell, 2009) makes an attempt in this direction by augmenting the model with a post-lexical integration stage, named I. This stage is assumed to reflect all of the post-lexical processing like linking the word into a syntactic structure, generating a context-appropriate semantic representation, and incorporating its meaning into a discourse model. However, the amount of time to complete I, t(I), is independent of the language processing demands at that word; instead t(I) is sampled from a gamma distribution having a mean of 25 msec and standard deviation of 0.22. Models of sentence processing like the two evaluated here or, preferably, a systematic combination of them would offer a more realistic way of computing t(I). A similar approach of incorporating post-lexical processes can be taken in other eye movement models depending on the particular architecture of each model.

Conclusions

This work evaluated the combined contribution of two theories of sentence processing, viz., surprisal and cue-based retrieval theory. The two approaches capture different aspects of sentence processing, namely instantaneous probabilistic disambiguation and processing constraints due to memory retrievals. It was shown that when effects of these theories were combined together to predict eye movements measures, they emerged as significant predictors even when word length, n-gram frequency and Cloze predictability were taken into account. Moreover, they showed an over-additive effect on several eye movements measures. This needs to be taken into account in future models of sentence processing that integrate surprisal and retrieval costs. Also, models of eye movement could benefit from this work. Although the size of the evaluation corpus is small (total 32 sentences and 222 participants) and models of cue-base parsing were hand-crafted, this work serves as a first step towards developing a broad coverage model of sentence processing that combines the two processes – probabilistic disambiguation and memory retrieval – in a principled way.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2005). An integrated theory of mind. *Psychological Review*.

Bates, D., & Sarkar, D. (2007). lme4: Linear mixed-effects models using S4 classes (R package version 0.9975-11) [Computer software].

Boston, M. F., Hale, J., Kliegl, R., Patil, U., & Vasishth, S. (2008). Parsing costs as predictors of reading difficulty: An evaluation using the potsdam sentence corpus. *Journal of Eye Movement Research*, *2*(1), 1-12.

Demberg, V., & Keller, F. (2008). Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, *108*(2), 193-210.

Ehrlich, S. F., & Rayner, K. (1981). Contextual effects on word perception and eye movements during reading. *Journal of Verbal Learning and Verbal Behavior*, *20*, 641-655.

Engbert, R., Nuthmann, A., Richter, E. M., & Kliegl, R. (2005). Swift: A dynamical model of saccade generation during reading. *Psychological Review*, *112*(4), 777–813.

Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.

Geyken, A. (2007). The DWDS corpus: A reference corpus for the German language of the 20th century. In C. Fellbaum (Ed.), *Idioms and Collocations: Corpus-based Linguistic, Lexicographic Studies.* London: Continuum Press.

Gibson, E. (2000). Dependency locality theory: A distance-based theory of linguistic complexity. In A. Marantz, Y. Miyashita, & W. O'Neil (Eds.), *Image, language, brain: Papers from the first mind articulation project symposium.* Cambridge, MA: MIT Press.

Grodner, D., & Gibson, E. (2005). Some consequences of the serial nature of linguistic input. *Cognitive Science*, *29*(2), 261290.

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics* (pp. 1–8). Pittsburgh, PA.

Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, *20*(2), 137-194.

Jurafsky, D. (2003). Probabilistic modeling in psycholinguistics: Linguistic comprehension and production. In R. Bod, J. Hay, & S. Jannedy (Eds.), *Probabilistic linguistics.* MIT Press.

Kliegl, R., Geyken, A., Hanneforth, T., & Würzner, K. (2006). *Corpus matters: A comparison of German DWDS and CELEX lexical and sublexical frequency norms for the prediction of reading fixations.* Unpublished manuscript.

Kliegl, R., Grabner, E., Rolfs, M., & Engbert, R. (2004). Length, frequency and predictability effects of words on eye movements in reading. In *Eye movements and information processing during reading* (Vol. 16, p. 262-284). East Sussex: Psychology Press. (Special Issue)

Kliegl, R., Nuthmann, A., & Engbert, R. (2006). Tracking the mind during reading: The influence of past, present, and future words on fixation durations. *Journal of Experimental Psychology: General*, *135*, 12-35.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, *106*(3), 1126-1177.

Lewis, R. L. (1996). Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research*, *25*, 93115.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 1-45.

Pollatsek, A., Reichle, E. D., & Rayner, K. (2006). Tests of the e-z reader model: Exploring the interface between cognition and eye-movement control. *Cognitive Psychology*, *52*, 1 – 56.

Reichle, E., Warren, T., & McConnell, K. (2009). Using e-z reader to model effects of higher level language processing on eye-movements during reading. *Psychonomic Bulletin & Review*, *16*(1), 1– 21.

Stolcke, A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, *21*, 165–202.

Taylor, W. (1953). Cloze procedure: a new tool for measuring readability. *Journalism Quarterly*, *30*, 415–433.

Vasishth, S., Bruessow, S., Lewis, R. L., & Drenhaus, H. (2008). Processing polarity: How the ungrammatical intrudes on the grammatical. *Cognitive Science*, *32*(4).

Error and expectation in language learning: An inquiry into the many curious incidents of "mouses" in adult speech

Michael Ramscar & Melody Dye
Department of Psychology, Stanford University,
Jordan Hall, Stanford, CA 94305.

Abstract

Much can be learned about the world by examining the discrepancies between what is expected and what actually occurs. Although many formal learning theories make use of prediction error as an important— even necessary—component in explaining behavior, this source of evidence has been largely overlooked in the language-learning literature. In this paper, we show how incorporating prediction error into a model of plural word learning (Ramscar & Yarlett, 2007) can yield a surprising prediction: that at an appropriate point in learning, the tendency of children to over-regularize irregular plurals can be reduced, by exposing them to regular plurals alone. We report on an experiment, which was designed to test the model's predictions empirically. The findings indicate that memory testing on regular plurals led to significant reductions in the rates of plural over-regularization in six-year-olds.

Introduction

Gregory: "Is there any other point to which you would wish to draw my attention?"
Holmes: "To the curious incident of the dog in the night-time."
Gregory: "The dog did nothing in the night-time."
Holmes: "That was the curious incident."
"Silver Blaze," Sir Arthur Conan Doyle.

A racehorse vanishes on the eve of an important race, its trainer murdered. Sherlock Holmes lights upon a crucial piece of evidence: a dog on the premises has remained silent throughout the time in question. The fact that the dog did not bark – and thus, that an expected event did not occur – proves an important clue to the identity of the murderer. As the curious incident of the dog in the nighttime reminds us, much can be learned from discrepancies between what is expected and what actually occurs.

In what follows, we show how in the ordinary course of their lives, people use the discrepancy between what they expect and what they actually experience as a vital source of information in learning; and that often, as in the case of Sherlock Holmes and The Silver Blaze, the non-occurrence of expected events provides important negative evidence. That people use such evidence is only natural: expectation and prediction-error are important components of animal learning (Rescorla, 1988). However, these factors have been largely overlooked in discussions of children's learning, especially in relation to language. The extensive literature asserting the lack of negative evidence to children learning language (e.g., Chomsky, 1959; Pinker, 1984, 2004; Marcus, 1993) either ignores expectation and error-driven learning, or treats them superficially at best. Expectation is usually dismissed as a weak form of 'indirect negative evidence' that can offer little to no assistance in the complex process of language acquisition (Pinker, 2004). Here we show that prediction-error provides an abundant source of evidence in human learning, and in particular language learning, by testing and confirming an intriguing prediction that error-driven learning makes about children's plural over-regularization errors: namely, that at an appropriate point in learning, the tendency of children to over-regularize irregular plurals can be reduced through exposure to regular plurals alone.

Prediction error and learning theory

Formal learning models are able to account for a wide range of the effects associated with learning by assuming that learning is driven by the discrepancy between what is expected and what is actually observed (error-driven learning). The learned predictive value of cues produces expectations, and any difference in the value of what is expected versus what is experienced produces further learning. In the Rescorla-Wagner (1972) model, for example, the change in associative strength between a stimulus i and a response (or event) j on trial n is defined as: [1]

$$\Delta V_{ij}{}^{n} = \boldsymbol{\alpha}_i \boldsymbol{\beta}_j \, (\lambda_j - V_{total}) \qquad (1)$$

Learning is governed by the value of $(\lambda_j - V_{TOTAL})$ where λ_j is the value of the predicted event and V_{total} is the predictive value of a set of cues. In the ordinary course of learning, the discrepancy between λ_j and V_{total} reduces over repeated trials, producing a negatively accelerated learning curve, and asymptotic learning.

What is often overlooked is what happens when a predicted event does not occur. If a cue predicts something that doesn't follow, then λ_j will have a value

[1] n indexes the current trial. $0 \leq \alpha_i \leq 1$ denotes the saliency of cue $_i$, $0 \leq \beta_j \leq 1$ denotes the learning rate of event$_j$, λ_j denotes the maximum amount of associative strength that cue$_j$ can support, and V_{total} is the sum of the associative strengths between all cues$_j$ present on the current trial and event$_j$.

of zero for that trial. In this case the discrepancy (λ_j - V_{TOTAL}) will have a negative value, resulting in a reduction in the associative strength between the cues present on that trial and the absent feature *j*. For example, in modeling learning in a dog being trained to expect food when a bell is sounded, setting λ_j to 1 for training trials where food is given, and 0 for later trials when no food appears, allows for the characteristic patterns of training and extinction to be modelled. This means that latent learning about the relationship between cues and events that are not actually present occurs in these circumstances, and it is this process that is a key aspect of learning.

Thus, in error-driven learning, cues compete with one another for relevance, producing associative learning patterns that can differ greatly from those that would arise out of a record of the correlation between cues and outcomes (Rescorla, 1988). There is evidence for this mechanism at a neural level. Increases and decreases in the firing rates of monkeys' striatal dopamine neurons appear to track the degree to which the outcomes of training trials are under- or over-predicted (Hollerman & Schulz, 1998).

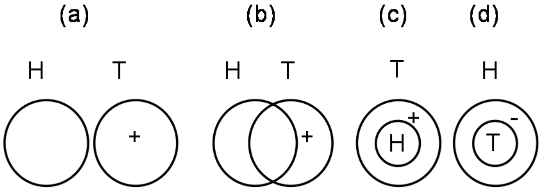


Figure 1. Four logical situations a child might arrive at while trying to "learn" a language (for the purposes of the example, language learning is assumed to be a process in which the child guesses the grammar that underlies that adult target language). Each circle represents the set of sentences constituting a language. "H" stands for the child's "hypothesized language"; "T" stands for the adult "target language." "+" indicates a grammatical sentence in the language the child is trying to learn, and "-" represents an ungrammatical sentence (Pinker, 1989).

Expectation in language learning

A good example of the considerations that have led to the widespread belief that much of the conceptual structure of language is innate (see e.g. Pinker, 1984) is the "logical problem of language acquisition" (LPLA). A classic statement of this is provided by Pinker (1984) and is depicted in Figure 1. According to the LPLA, in attempting, to learn language, children "hypothesize the grammar of the adult language" (strictly, the child's task is to guess guessing the set of grammatical sentences that comprise a language; Gold, 1967).

Possible languages are depicted as circles corresponding to sets of word sequences, and four logical possibilities for how a child's hypothesis might differ from adult language are given. In the first possibility (a), the child's hypothesis language, H, is disjoint from the language to be acquired (the "target language" - T). In terms of noun usage, on which we focus here, this corresponds to the state of a child learning English who cannot produce any well-formed noun plurals (the child might say things like "the mouses" but never "the mice."). In (b), the sets H and T intersect, corresponding to a child who has learned some nouns correctly but others incorrectly (the child uses nouns like "mice" alongside incorrect words like "gooses"). In (c), H is a subset of T, which means that the child has mastered usage of some but not all English noun plurals and never uses forms that are not part of English. Finally, in (d), H is a superset of T, meaning that the child has mastered all English nouns but nevertheless produces some forms that are not part of the English language (i.e., the child says both "mouses" and "mice" interchangeably).

Since the LPLA assumes that learners cannot recover from erroneous inferences without corrective feedback, and because children do not get the kind of feedback required (Brown & Hanlon, 1970), in addition to the fact that they through stage (d), it follows accordingly that, children cannot acquire language simply by attending to the input. (Indeed, the idea that language is learned purely from experience is often regarded as having been effectively disproved; see Baker, 1979; Gold, 1967; Pinker, 1989)

However, the assumption that explicit negative feedback is needed for children to correct errors is entirely inconsistent with the principles of error-driven learning described above, and Ramscar and Yarlett (2007) provide an account of the way that general error-driven learning principles can give rise to the patterns of children's plural inflection acquisition. Ramscar and Yarlett's (2007) model represents plural items as semantic cues to phonological outcomes. Each item is an exemplar comprising an associatively linked semantic and a phonological component. For example, the plural noun CARS is represented by a couplet encoding the association between the general semantics of cars, including their plurality, and the phonological form */carz/*. The model assumes that learning is driven both by what the child has heard, and what the child expects to hear based on prior experience.

Over-regularization – children saying *foots* instead of *feet*, for example – arises in the model out of an initial failure to discriminate the individual semantic cues to particular plural words. In early learning, this lack of discrimination results in interference when shared cues activate frequent (and thus strongly learned) regular forms during the production of infrequent (and thus weakly learned) irregular forms. Interference thus results from prediction error generated by shared semantic cues. Accordingly, the associative values of these shared cues get weakened as learning progresses, which results in irregular forms becoming better discriminated and a decline in interference. Because regular and irregular forms are learned at different rates (there are far more regular than irregular plurals) and require different degrees of discrimination (regular plurals are supported by other regulars, but interfere with irregulars) the model predicts that interference effects will worsen for a time in the earliest stages of learning (because of the speed with which regular forms are learned), before slowly resolving as irregular forms become better learned. The model thus predicted that older children could improve their production of correct irregulars by repeatedly generating plurals (indeed even if they produce over-regularizations), but that this might be less beneficial to younger children.

These predictions were supported by the outcomes of several empirical tests (Ramscar & Yarlett, 2007). In one study, children repeatedly named plurals (correctly and incorrectly) for several blocks of regular and irregular items. The older children converged on the correct irregular plurals (e.g., production of "child" decreased, while "children" increased), without corrective feedback, however under the same conditions, younger children's over-regularization worsened, consistent with 'U-shaped' learning. A similar pattern of data was obtained when a semantic memory task for pictures was interspersed between pre- and post- tests of plural production: older children who performed an old/new task on pictures of regular and irregular plural items over-regularized less on the post-test, while younger children over-regularized more.

Can over-regularization be reduced by exposure to regular items alone?

A strong, very counterintuitive prediction that arises out of the principles of error-driven learning was not tested in Ramscar & Yarlett's (2007) studies. This is that at an appropriate point in learning, children's tendency to over-regularize irregular plurals will be reduced if they given training on only regular plurals. The way that this surprising prediction arises can be explained as follows: because regular nouns in English are frequent (both in terms of the number of regular plural noun types, and the overall number of plural noun tokens that are regular), the majority of plural forms cued by "plurality" will be plural forms which resemble their singular forms, but which end in + /S/. Since over-regularization is a failure to discriminate the appropriate cues to individual items present, (i.e., generalization) – if children encounter the cues of to regular plurals (e.g., a group of dogs), poor discrimination will result in the prediction of irregulars. The resultant prediction error will lead to children learning to negatively associate regular cues with irregular forms, which will increase the discrimination of regulars and irregulars. This increased discrimination of irregular plurals will in turn lead to a reduction in over-regularization. Further, although prediction errors for irregular items are caused by the activation of the cues for regular items, the erroneous prediction of irregulars is a function of how well the irregular items have been learned. Early in development, when irregulars are weakly learned, exposure to regular plurals will generate little irregular prediction error as compared to later in development, when irregulars will be better learned.

Simulation Experiment

To formally test these ideas, we implemented a simple model of how children might learn to discriminate plural forms over time (see also Ramscar & Yarlett, 2007). The model assumes that plural items are represented as semantic cues to phonological outcomes. In early learning, over-regularization arises because the semantic representations of irregular plural items are not sufficiently discriminated from those of regular plurals, i.e., children initially tend to pluralize in response to general plurality, rather than in response to specific plural items (Ramscar & Yarlett, 2007). In the simulation, this was represented in terms of two competing hypotheses, which were reinforced whenever an irregular plural item was presented. One hypothesis was item specific (e.g., plural mouse is the cue to mice), while the other was more general (i.e., e.g., plurality is the cue to mice). Simultaneously, we simulated the learning of regular plurals. Due to the fact that regular plurals occur more frequently, and because their singular and plural forms overlap, we assumed that they offer more support to the general plural semantic hypothesis than irregular plurals, which instead offer support to more item-specific hypotheses.

Learning about the couplets was simulated using the Rescorla-Wagner (1972) rule described above. In the simulation, the learning rate, β_j , for the semantic hypotheses (cues) was set at a constant, and λ_j was set at 100% for the semantic-phonological couplets, which included both regular and irregular plurals forms. To simulate the high type and token frequency of regular plurals, V_{ij} for the regular plurals was learned with α_i set to a high value (i.e., in the Rescorla-Wagner model, α_i effectively serves as a separate learning rate for each cue$_i$) while V_{ij} for the irregular plurals was learned with α_i set to a low value.[2] This allowed training to be simulated by alternately presenting the model with regular and irregular items in training, to simulate a child's exposure to regular and irregular plurals at different frequency levels.

To examine the effect of exposure to regular plurals alone at different stages in learning, the presentation of irregular plurals was withheld for 10 trials, the first of

[2] In the simulation: $\boldsymbol{\beta}_j$=0.3 $\boldsymbol{\alpha}_I$ regular=0.4; $\boldsymbol{\alpha}_I$ irregular=0.15.

these coming early in the model's training, and the second later in training, after the response to regular plurals had asymptoted. Figure 2 shows the learning of the two irregular hypotheses (general and specific) and the general regular hypothesis.

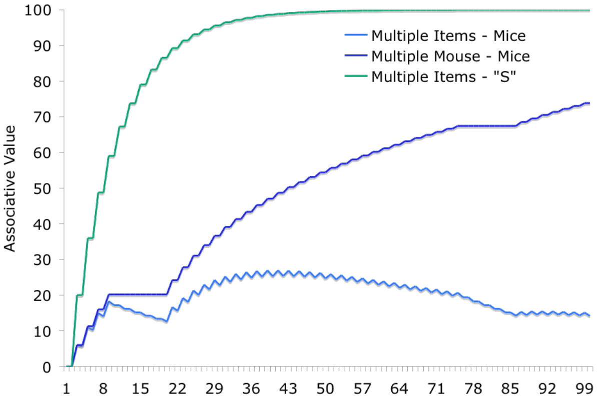


Figure 2. Learning of the semantic cues to an irregular item such as *mice* and the regular /S/. The periods in which no irregular trials occured appear as horizontal lines on the plot representing the *multiple mouse items⇒ mice* hypothesis.

As in Ramscar & Yarlett (2007) the likelihood of over-regularization (i.e. failure to produce the learned response) was modeled as a result of response competition, caused by spreading activation to items in memory that are activated by the semantics of the situation but which correspond to different phonological forms. This activation is modeled as a function of the degree to which the competing semantic-phonological couplets have been learned, the strength of the semantic cue that co-activates them and a spreading activation parameter *S* (Ramscar & Yarlett, 2007). Figure 3 shows the strength of this interference signal across the training period, and Figure 4 shows the effect this competition has on the likelihood that a learned irregular response will be reproduced. In Figure 4, response propensity is calculated by subtracting the value of the interference signal from the value of the correct response (Ramscar & Yarlett, 2007).

As can be seen from Figures 3 and 4, prediction errors for irregular items are caused by the activation of cues related to regular items, which results in the unlearning of the *multiple items⇒irregular* cue. Early in development, when irregulars are weakly learned, exposure to regular plurals will generate less overall irregular prediction error, and the overall frequency of regulars will result in a steady increase in the level of interference that produces over-regularization. Later in development, exposure to regular plurals produces more irregular prediction error, and interference no longer increases. As a result, the model predicts that depending on the overall prior exposure a child has had to plurals, exposure to regular plurals alone can lead to opposite effects (e.g., 'U-shaped' learning; Ramscar & Yarlett's 2007 model and empirical data showed that interspersing regular and irregular items produced this pattern of learning).

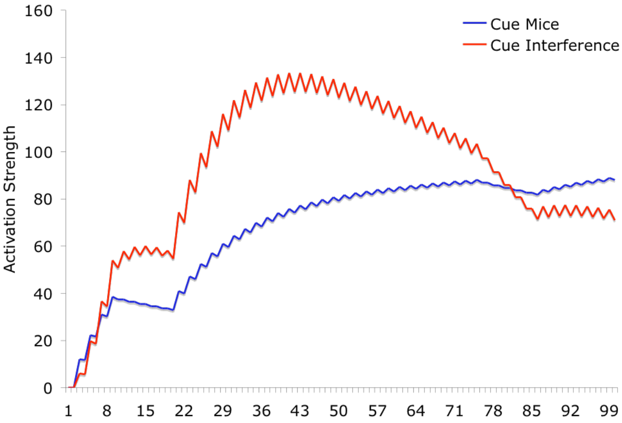


Figure 3. Interference and imitation in training. These parameter values were chosen to best illustrate our predictions; the important thing to note is the wunderlying *relationship* that arises out of the different learning rates.

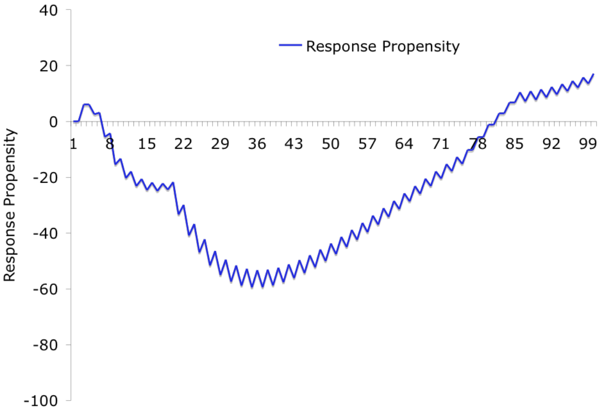


Figure 4. Response propensity levels over training. Over-regularization will be likely when this value is negative.

Human Experiment

We tested these predictions using a semantic old/new task to expose children to regular plurals, and a test-train-test paradigm to establish a baseline rate of over-regularization for each child. This allowed us to examine the effect of children's exposure to regular plurals has on later irregular plural production (see Ramscar & Yarlett, 2007). Semantic priming (e.g., where priming the semantics of "doctor" yields shorter response latencies in a lexical decision task on "nurse";

Meyer & Schvaneveldt, 1971) indicates that phonological and orthographic representations can be activated by cueing their semantic features. The Ramscar & Yarlett (2007) model assumes that until the representation of a phonological–semantic association reaches asymptote, the activation of an association can strengthen its representation (see Roediger & Karpicke, 2006). Thus explicitly priming the semantics of the nouns, even in the absence of any overt naming responses by the child, was expected to be sufficient to produce errors in prediction and subsequent latent learning. Furthermore, by not having children explicitly name items, we aimed to reduced the effects of perseveration on spoken motor responses have in children's performance during a post-test. This we expected would allow for a better measure of their representation of the items tested.

Participants

24 four and 23 six year old children living resident in the vicinity of Palo Alto, California, and recruited from a database of volunteers. The average ages were 4 years and 7 months for the four year olds, and 6 years and 7 months for the six year olds.

Methods and materials

The children were randomly assigned to two groups, both of which were pre-tested on plural production.[3] In the elicitation test the children were asked to help a cookie monster puppet name a series of six irregular nouns, and six regular pairings of plural nouns. The children sat with the experimenter and named the nouns first from singular and then from plural depictions that were presented on a laptop computer.

In the experimental condition the children then performed an old/new task in which they were asked to tell a cookie monster whether or not they had seen depictions similar to those they had named in the pre-test. All depictions of the "old" items in training were novel, which required children to make categorization judgments to generate the correct answers. The children were asked to help the cookie monster identify them "By telling him, yes or no" to indicate whether they had already seen these depictions or not. When an object appeared, the experimenter asked the child to "Look at those – did cookie monster see those before?" Children who did not spontaneously respond were prompted, "Did cookie see these? Yes? No?". If no response was forthcoming, the experimenter proceeded to the next item. Half of the presented items were new depictions of the regular items in the pre-test and half were foils. The children were thus tested on 12 new and 12 old items per block. All of the items were presented as depictions on a computer screen.

In the control condition, the children were shown 6 color slides after the pre-test, and then asked to tell the cookie monster whether they had seen that particular color before in an old/new task that contained an equal number of foils. The colors were presented as blocks filling the computer screen to avoid cuing any notion of plurality. The total time to complete each was equal. Both sets of children were then post-tested on exactly the same set of depictions that were used in the pre-test.

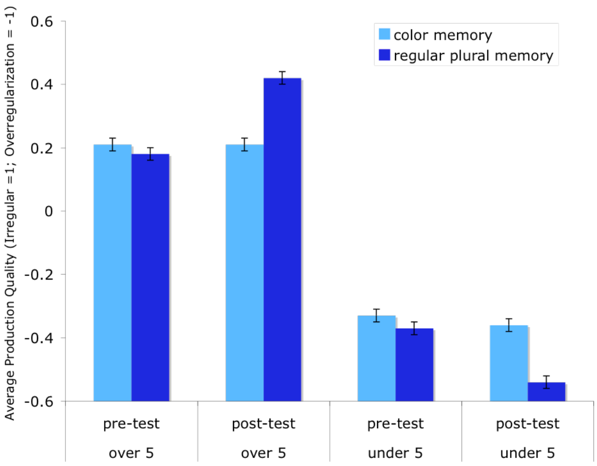


Figure 5. Pre and post test performance by age and condition

Results

The results overwhelmingly supported our predictions. The performance of the older children in the experimental condition improved between pre-and post test (t(64)=2.256, p<0.05) while the performance of the younger children declined (t(66)=1.955, p<0.05). There was little change in the performance of either age group in the control condition (see Figure 5). A 2 (pre- to post- test) x 2 (age) x 2 (condition) repeated measures ANOVA of the children's plural production revealed a significant interaction between age and pre- to post-test performance (F(1,43) = 8.32, p<0.01), and a significant interaction between age, training type and pre- to post-test performance (F(1,266) = 4.235, p=.05).

General Discussion

We found that testing memory for regular plurals significantly reduced the rates of plural over-regularization in six-year-olds. Though the strength of these results is likely to have been influenced by recency (children named the irregulars immediately prior to regular training), what is clear that the children learned about irregular plurals, and improved their

[3] The irregular items were MOUSE-MICE, CHILD-CHILDREN, SNOWMAN-SNOWMEN, GOOSE-GOOSE, TOOTH-TEETH and FOOT-FEET; the regular matches were RAT, DOLL, COW, DUCK, EAR, and HAND. Ramscar & Yarlett (2007) Experiment 1 revealed that although children of these ages over-regularize these irregular plurals, they have reliable knowledge of their correct forms.

production of them, even though none were present during the training trials. We feel that, to the extent that this result is surprising, this surprise is due to the lack of widespread understanding of error-driven learning processes (see also Rescorla, 1988).

Overwhelmingly, research into language learning has pre-occupied itself with the observable: that is, with what a child hears or sees. Researchers have variously touted "the lack of negative evidence" in language learning as a constraint on theory (Marcus, 1993; Pinker, 2004), and much virtue is attributed to models that learn from "positive evidence" alone. We feel this is regrettable. There is good reason to believe that error-driven learning describes the principal mechanism by which people acquire information about their environment (Miller, Barnet & Grahame, 1995; Siegel & Allen, 1996; Ramscar & Yarlett, 2007; Ramscar, et al, in submission). The basic principles of error-driven learning are supported both by animal (e.g., Kamin, 1969; Rescorla & Wagner, 1972) and neurobiological models (e.g., Hollerman & Schultz, 1998; Barlow, 2001). In developing accounts of human learning, error-driven learning ought to be primarily considered when it comes to establishing conceptual and theoretical constraints and default hypotheses.

Extrapolating from the findings presented here (see also Ramscar & Yarlett, 2007; Ramscar et al, in submission), it seems likely that the processes involved in verbal learning – reducing prediction-error between semantic cues in the world and linguistic forms – are critical to the development of our use of language as an abstract representational device in communication.

Understanding language in terms of learning may, in the future, involve a reassessment of what human communication involves, requiring and inspiring new theories of language and its role in culture (Wittgenstein, 1953; Quine, 1960; Tomasello, 1999). At the very least, we would argue that simply reversing the trend ~~to~~ of ignoring learning in human development, we can and will reap many important scientific benefits.

References

Baker, C. (1979). Syntactic theory and the projection problem. *Linguistic Inquiry*, 10, 533–581

Barlow H. (2001). Redundancy reduction revisited, *Network: Computation in Neural Systems*, 12, 241-253

Brown, R., & Hanlon, C. (1970). Derivational complexity and order of acquisition in child speech. In J. R. Hayes (Ed.), *Cognition and the development of language*. New York: Wiley.

Hollerman J.R., Schultz W. (1998) Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature Neuroscience*, 1: 304-309.

Kamin L.J. (1969). Predictability, surprise, attention, and conditioning. In: Campbell B, Church R (eds). *Punishment and Aversive Behaviour*. Appleton-Century-Crofts: New York.

MacWhinney, B. (2004). A multiple process solution to the logical problem of language acquisition. *Journal of Child Language,* 31, 883–914

Marcus, G. F. (1993). Negative evidence in language acquisition. *Cognition*, 46, 53–85

Meyer, D.E. and Schvaneveldt, R.W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90, 227-234

Miller R.R., Barnet R.C. and Grahame N.J. (1995). Assessment of the Rescorla-Wagner Model, *Psychological Bulletin*, 117(3), 363-386.

Pinker, S. (1984). *Language learnability and language development.* Cambridge, MA: Harvard University Press

Pinker, S. (2004) Clarifying the logical problem of language acquisition. *Journal of Child Language*, 31, 949-953.

Quine, W.V. (1960) *Word and Object*, Cambridge, MA: MIT Press.

Ramscar, M. & Yarlett, D. (2007) Linguistic self-correction in the absence of feedback: A new approach to the logical problem of language acquisition. *Cognitive Science*, 31, 927-960

Ramscar, M. Yarlett, D., Dye, M., Denny, K., & Thorpe, K. (in submission) The Feature-Label-Order Effect In Symbolic Learning

Rescorla R.A. (1988). Pavlovian Conditioning: It's Not What You Think It Is, *American Psychologist*, 43(3), 151-160.

Rescorla R.A. and Wagner A.R. (1972). *A Theory of Pavlovian Conditioning: Variations in the Effectiveness of Reinforcement and Nonreinforcement.* In Black & Prokasy (Eds.), Classical Conditioning II: Current Research and Theory. New York: Appleton-Century-Crofts.

Roediger, H.L., & Karpicke, J.D. (2006). Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science*, 17, 249-255

Rudy, J. W. (1974). Stimulus selection in animal conditioning and paired-associate learning: Variations in the associative process. *Journal of Verbal Learning & Verbal Behavior,* 13, 282-296

Rumelhart D. E. & McClelland J. L (1986) On learning past tenses of English verbs. In Rumelhart & McClelland (eds) *Parallel Distributed Processing: Vol 2*: Psychological & Biological Models. Cambridge, MA: MIT Press.

Siegel, S.G., & and Allan, L.G. (1996). The widespread influence of the Rescorla-Wagner model, *Psychonomic Bulletin and Review*, 3(3), 314-321

Tomasello, M. (1999). The cultural origins of human cognition. Cambridge, MA: Harvard University Press

Wittgenstein, L (1953). *Philosophical Investigations* Oxford: Blackwell

Towards Explaining the Evolution of Domain Languages with Cognitive Simulation

David Reitter (reitter@cmu.edu) and **Christian Lebiere**
Carnegie Mellon University
Department of Psychology, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA

Abstract

We simulate the evolution of a domain language in small speaker communities. Data from experiments (Garrod et al., 2007; Fay et al., 2008) show that human communicators can evolve graphical languages quickly in a constrained task (Pictionary), and that communities converge towards a common language even in the absence of feedback about the success of each communication. We postulate that simulations of such horizontal evolution have to take into account properties of human memory (cue-based retrieval, learning, decay). We implement a model that can draw abstract concepts through sets of non-abstract, related concepts, and recognize such drawings. The knowledge base is a network with association strengths randomly sampled from a natural distribution found in a text corpus; it is a mixture of knowledge shared between agents and individual knowledge. In three experiments, we show that the agent communities converge, but that initial convergence is stronger when communities are structured so that the same pairs of agents interact throughout. Convergence is weaker in communities when agents do not swap roles (between recognizing and drawing), predicting the necessity of bi-directional communication in domain language evolution. Average and ultimate recognition performance depends on how much of the knowledge agents share initially.

Keywords: Alignment; Language Evolution; Domain Languages; Microevolution; Cognitive Architectures, Multi-Agent Simulation

Introduction

Languages evolve: like biological systems, they undergo mutation and selection as they are passed on between speakers and generations. Similar to its biological counterpart, human communication evolves under environmental constraints. Fitness of a communication device (software) is a function also of the cognitive hardware: cognitive facilities constrain the language system. In this paper, we use an independently motivated cognitive memory architecture to constrain an evolutionary process that produces a communication system.

Recent models of dialogue describe how interlocutors develop representation systems in order to communicate; such systems can, for instance, be observed using referring expressions that identify locations in a maze. Experiments have shown that referring expressions converge on a common standard (Garrod & Doherty, 1994). Pickering & Garrod's (2004) Interactive Alignment Model suggests that explicit negotiation and separate models of the interlocutor's mental state aren't necessary, as long as each speaker tends to adapt to themselves and their interlocutors, as they are known to do on even simple, linguistic levels (lexical, syntactic).

Some evolutionary models (vertical models) see the transmission of cultural information as a directed process, in which information is passed only from the older to the younger generation. Horizontal models explain the emergence of language as a continuous process within generations. *Individualistic* models of language evolution assume that innate learning and processing systems set a prior, towards which language converges. Interaction and the cultural environment do not leave marks in the resulting language. *Collaborative* models, on the other hand, accept that language mutates and converges within generations as well. They claim that meaning-symbol connections spread between collaborating agents and ultimately converge on a predominant one. It is the dichotomy between individual and community-based learning that motivated the experiments by Garrod et al. (2007) and Fay et al. (in prep.), which serve as the basis for the model presented here.

In the horizontal society of cognitive agents in our study, agents adapt their communication system collaboratively to environmentally shaped and cognitively constrained needs of each individual. With our model, we aim to use a cognitive framework – specifically a memory model – to reflect processes in the individual that give rise to emergent convergence and learning within the community. By this, we acknowledge the fact that cultural evolution is constrained by individual learning; each agent learns according to their cognitive faculty (cf., Christiansen & Chater, 2008). The possibility of cultural language evolution has been supported by computational simulations (e.g., Kirby & Hurford, 2002; Brighton et al., 2005).

It is because adaptation according to experience is determined by human learning behavior that simulation in validated learning frameworks is crucial. Griffiths & Kalish (2007) for instance model language evolution among rational learners in a Bayesian framework; the purpose of the present project is to simulate the evolution of a communication system using an architecture with an accurate account of memory access and a concrete experimental design. We will introduce a cognitive model that simulates a participant in the experiment; multiple models interact as a community of participants. The purpose of this paper is to observe how a compositional language system is created between collaborating agents in a computational, cognitive simulation. We will show that the model demonstrates learning behavior similar to the empirical data. We assume these agents share a common reference system initially, display cooperative behavior and adopt mixed roles as communicators. Therefore, we explore different scenarios that test the necessity of our preconditions, in particular the initial common ground and the fact that each agent can be both on the sending and the receiving end of the communications.

The Task

The Pictionary experiment (Garrod et al., 2007) involves two participants, a *director*, who is to draw a given meaning from a list of concepts known to both participants, and a *matcher*, who is to guess the meaning. Director and matcher do not communicate other than through the drawing shared via screens of networked computers; the matcher is able to draw as well, for instance to request clarification of a part of the picture. Each trial ends when the matcher decides to guess a concept. Garrod et al.'s set of concepts is divided into five broad categories (e.g., actor, building); the concepts within each are easily confusable (e.g., drama, soap opera). Each game involves several trials, one for each concept on the list, in randomized order. The director is not informed of the guess made by the matcher, and neither participant receives feedback about whether the guess was correct. Participants switch roles after each trial. Participants to play many games so that the emergence of consistent drawings can be observed.

We implement the experiment in a form applied by Fay et al. (in prep., 2008), where 16 concepts (plus 4 additional distractors) were used in a design with two conditions. In the *isolated pair* condition, participants were split into fixed pairs. They played seven rounds of six games each with the same partner. In the *community* condition, participants changed partners after each round. Each community consisted of eight participants. The pattern of pairings was designed so that after the first round, four sub-communities existed, after the second round, two sub-communities. After round four, the largest separation between partners was 2 (i.e., each agent has interacted via another one with every other agent); it was 1 after round seven. Fay et al. evaluated the iconicity of drawings, showing that isolated pairs developed more idiosyncratic signs, while the signs emerging within communities were more metaphoric (i.e. deducible) and easier to understand for new (fictitious) members of the language community. As idiosyncracy increases with each drawing-recognition cycle, but resets (to some degree) when communication partners change, communities may end up evolving similar idiosyncracy once every pair of participants played the same number of games.

The simplest measure and the one crucial for the evaluation of models like ours is *identification accuracy*. Fay et al. found that their participants generally converged quickly to a common meaning system. Convergence reached a ceiling of around 95% in both community and isolated-pair conditions. Changing interaction partners from round to round, as in the community condition, reduced accuracy during the initial changes; however, the community reached good ID accuracy after just a few rounds. We will use the development of ID accuracy as one way to evaluate the model.

The Model

ACT-R (Anderson, 2007) is an architecture for specifying cognitive models, one of whose major components is memory. ACT-R's memory associates symbolic chunks of information (sets of feature-value pairs) with subsymbolic, activation values. Learning occurs through the creation of such a chunk, which is then reinforced through repeated presentation, and forgotten through decay over time. The symbolic information stored in chunks is available for explicit reasoning, while the subsymbolic information moderates retrieval, both in speed and in retrieval probability. The assumption of rationality in ACT-R implies that retrievability is governed by the expectation to make use of a piece of information at a later point. Important to our application, retrieval is further aided by contextual cues. When other chunks are in use (e.g., *parliament*), they support the retrieval of related chunks (*building*).

A single ACT-R model implements the *director* and *matcher* roles. As a director, the model establishes new combinations of drawings for given target concepts. As a matcher, the model makes guesses. In each role, the model revises its internal mappings between drawings and target concepts. Table 1 gives an example of the process. The model is copied to instantiate a community of 64 agents, reflecting the subjects that took part in the Pictionary experiments.

Our model uses a scalable and efficient re-implementation of ACT-R called *ACT-UP*, letting us underspecify model elements such as the production-rule system, which would neither introduce nondeterminism nor carry explanatory weight in this particular model.

Maintaining a communication system

The simplest form of keeping a communication system in ACT-R memory *chunks* is a set of signs. Each sign pairs a concept with a set of drawings. Competing signs can be used to assign multiple drawings for one concept, this would create *synonyms*; multiple concepts can also combine with the same drawings, creating *homonyms* and ambiguity.

To create new concepts, we need to introduce a subsymbolic notion of relatedness. We use ACT-R's spreading activation mechanism and weights between concepts to reflect relatedness. Spreading activation facilitates retrieval of a chunk if the current context offers cues related to the chunk. Relatedness is expressed as a value in log-odds space (S_{ji} *values*).

When the model is faced with the task to draw a given concept such as *Russell Crowe* (one of the concepts in the experiment) that has no canonical form as a drawing, a related but drawable concept (*drawing*) is retrieved from declarative memory. Similarly, we request two more concepts, deferring any desire of the communicator to come up with a distinctive rather than just fitting depiction of the target concept. The case of a model recognizing a novel combination of drawings is similar; we retrieve the concept using the drawings as cues that spread activation, making the target concept the one that is the most related one to the drawings.

After drawing or recognizing, the target or guessed concept, along with the component drawings, is stored symbolically in memory as a chunk for later reuse (*domain sign*). These signs differ from the pre-existing concepts in the network, although they also allow for the retrieval of suitable

Director	Matcher
Fails to retrieve domain sign for A.	
Retrieves related concept: ⇒ component drawings 123	
Draws components 1, 2, and 3	Requests related concept with cues 123 ⇒ concept B
	Guesses B
Learns domain sign A-123	Learns domain sign B-123
Retrieves domain sign for target concept B	
⇒ component drawings 345	
Verifies that B is retrieved when drawings 345 are activated	
Draws components 3, 4 and 5	Requests related concept with cues 345
	⇒ concept B
	Guesses B
	Verification: Requests domain sign for B
	⇒ domain concept B-123
	345 spread more activation to B than do 123,
Learns domain sign B-345	thus, learns domain sign B-345

Table 1: A protocol of two model instantiations, first failing to communicate concept A through three related drawings 1, 2 and 3, then successfully communicating concept B via drawings 3,4 and 5. The Matcher first adopts B-123 as a domain sign, then revises it to B-345.

drawings given a concept, and for a concept given some drawings. When drawing or recognizing at a later stage, the memorized domain signs are preferred as a strategy over the retrieval of related concepts. The system of domain signs encodes what is agreed upon as a language system between two communicators; they will be reused readily during drawing when interacting with a new partner, but they will be of only limited use when attempting to recognize a drawing combination that adheres to somebody else's independently developed communication system.

Knowledge

Agents start out with shared world knowledge. This is expressed as a network of concepts, connected by weighted links (S_{ji}). The distribution of link strengths is important in this context, as it determines how easily we can find drawing combinations that reliably express target concepts. Thus, the S_{ji} were sampled randomly from an empirical distribution: log-odds derived from the frequencies of collocations found in text corpus data. In a corpus comprising several years worth of articles that appeared in the *Wall Street Journal*, we extracted and counted pairs of nouns that co-occurred in the same sentence (e.g., "market", "plunge"). As expected, the frequencies of such collocations are distributed according to a power law. We found that the empirical log-odds resulting from these that form $S_{ji} = \log(P(J|I)/P(J|notI))$ (Anderson, 1993) (J and I being the events that J and I appear) can be approximated by a Generalized Inverse Gaussian-Poisson distribution (given in Baayen, 2001).

Such knowledge is, however, not fully shared between agents. Each agent has their own knowledge network resulting from life experience. This difference is essential to the difficulty of the task: if all agents came to the same conclusions about the strongest representation of target concepts, there would be little need to establish the domain language. We control the noise applied to the link strengths between concepts j and i for agent M (S_{Mji}) by combining the common ground S_{ji} (shared between all agents) with a random sample N_{Mji} in a mixture model: $S_{Mji} = (1-n)S_{ji} + nN_{Mji}$. Then, n [0;1] sets the proportion of noise. For Experiments 1 and 2, the noise coefficient is set to 0.2.

Adaptation pressure

Notably, participants in the experiment converged to a common sign system fairly quickly. This happened even though there was no evident, strong pressure to do so. Agents received no explicit feedback about the quality of their guesses or drawings. The only weak clue to the success of a set of drawings was whether the partner made a guess quickly. A helpful strategy for the matcher is to assume consistency between matching and drawing.

Invariably, the model will mistake a set of drawings for a reference to the wrong target. Lacking a feedback loop in this experiment, the model has no choice but to acquire even flawed domain signs and boost their activation upon repetition. Under these conditions, there is little pressure to converge. It is difficult to see how interaction partners could ever agree on a working communication system, given that there is no benefit for a model in choosing the concept-drawing associations of its interaction partner. However, the model does leverage *consistency* as proposed in Grice's maxims of manner, "Avoid ambiguity" and "Avoid obscurity of expression" (Grice, 1975). To do so, it assumes that a given set of drawings is associated with only one target concept, and, conversely, that a given target concept is associated with only three drawings. Suppose, for example (Table 1), that the model associates concept B with drawings $1,2,3$ (short: B-123). Later on, it comes across drawings $3,4,5$ as another good way to express B. In fact $3,4,5$ serve as convincingly stronger cues to retrieve B than do $1,2,3$. Thus, the model not

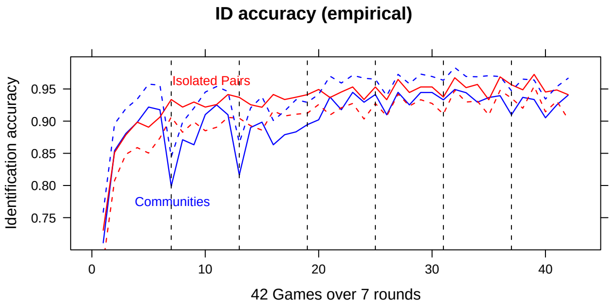


Figure 1: Identification accuracy for isolated pairs and communities (human data) as provided by Fay (p.c.). One-tailed 95% confidence intervals are given (upper bounds for communities, lower bounds for pairs), based on standard error (normality assumption).

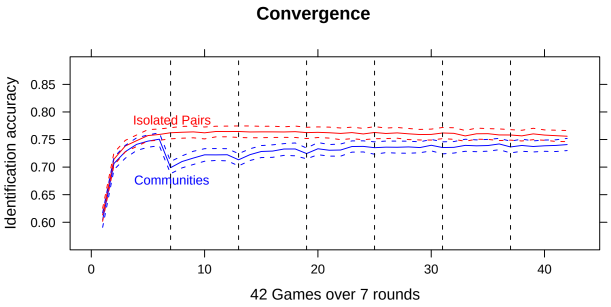


Figure 2: Mean identification accuracy in model simulations: As in the human data, both community pairs and isolated pairs gain most of their ID accuracy in the first game, but community pairs lose much accuracy when switching partners. 95% C.I., bootstrapped. 100 runs.

only correctly recognized B, but also learns the new preferred combination B-345. In the following rounds, B-345 will likely shadow the alternative in a winner-take-all paradigm, since B-345 is newer than B-123 and, thus, has stronger activation due to activation decay (noise and reinforcement may keep B-123 as a winner for longer). The decay mechanism counteracts the creation of synonyms.

In evolving the domain language, the model will avoid creating homonyms as well. Suppose a concept C is to be drawn, and 345 are retrieved as closely related and highly active drawings. Here, the model attempts to verify that 345 cannot be understood as any other concept than C. As the most strongly active concept for 345 is B, these drawings are ruled out to express C. With this mechanism, the model is able to cheaply modify the system of signs without extensive reasoning about the optimal combination every time a concept is added.

Algorithm

Directing The model is given a target concept A to convey. It uses *domain signs* and general knowledge to decide about a sign. At the end, the composed concept is committed to declarative memory as a domain sign. Domain knowledge is explicitly accessible and overrides subsymbolically derived compositions. As a consequence, the model acts with consistency: once a combination has first been used to convey a concept, the model will be more likely to use it. The director proceeds with the following algorithm.

1. Attempt to retrieve a domain sign for A of form $A-\alpha\beta\gamma$. If successful, verify by retrieving a domain sign B for the same three drawings $\alpha\beta\gamma$ is retrieved ($B-\alpha\beta\gamma$). Only if $A=B$, accept the domain sign $A-\alpha\beta\gamma$ and continue with step 3; otherwise choose another domain sign.
2. If no acceptable domain sign is found, use subsymbolic knowledge to combine concepts to express related target meanings. Using the target meaning as cue, retrieve three drawings $\alpha\beta\gamma$. The most active drawings are retrieved preferentially.
3. Draw $\alpha\beta\gamma$.
4. Learn $A-\alpha\beta\gamma$ (ACT-R buffer clearing action, repeated multiple times during the drawing process).

Matching Recognizing a drawing takes place in a similar fashion: domain knowledge is preferred over associative guesses. The model is given three drawings $\alpha\beta\gamma$. It proceeds with the following algorithm.

1. Attempt to retrieve a domain sign for $\alpha\beta\gamma$, resulting in $C-\alpha\beta\gamma$. If successful, verify by retrieving a domain sign of form $C-\delta\varepsilon\zeta$. Only if $\alpha,\beta,\gamma=\delta,\varepsilon,\zeta$, accept the domain sign $C-\alpha\beta\gamma$ and continue with step 3.
2. If no acceptable domain sign is found, retrieve a concept C using cues $\alpha\beta\gamma$ (spreading activation).
3. Guess C.
4. Learn $C-\alpha\beta\gamma$ (ACT-R buffer clearing action, repeated multiple times during the drawing process, but less often than during directing.)

ACT-R memory parameters were set to values consistent with the literature (transient noise 0.2, base-level constant 1.0, base-level learning and spreading activation enabled, retrieval threshold 1.0).

Experiment 1: Learning and Convergence

In the first experiment, we evaluate whether the model exhibits similar learning and convergence behavior, and whether there are differences in learning between the isolated-pair and community condition, as observed in Fay et al.'s experiment. The model uses the same number of concepts, trials and simulated participants as in the experiment.

Results

As shown in Figure 2, the learning behavior differs in the two conditions. *Isolated pairs* and *Community pairs* show a learning effect, i.e. they converge in their communication systems. However, unlike isolated pairs, community pairs dis-

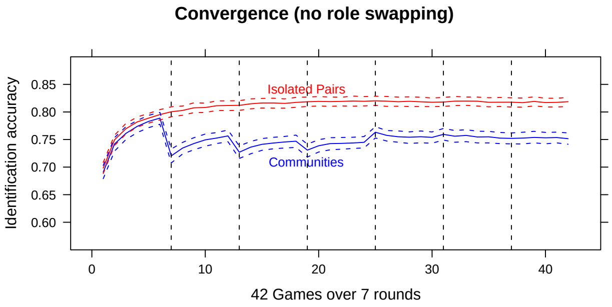


Figure 3: As in Figure 2, but without swapping roles.

play lower ID accuracy after the 7th game (game 1 of round 2), i.e. after switching partners.

We fitted a linear model to test some of the predictions more explicitly. The linear regression model treating round, game and condition (isolated pairs vs. communities) as independent variables, predicting log-transformed ID accuracy showed expected effects for round ($\beta = 0.03, p < 0.0001$) and game ($\beta = 0.02, p < 0.0005$), indicating improving accuracy with each game and round. An interaction of round and game ($\beta = -0.0046, p < 0.0005$) showed that the convergence leveled off in later rounds (as expected). There was no main effect of condition ($p = 0.45$), but an interaction of condition (isolated pairs) and round in the predicted direction ($\beta = -0.008, p < 0.05$), suggesting that convergence continued on for longer in the communities condition, and leveled off sooner in the isolated pairs condition. (All β in log space.)

Discussion

The results demonstrate, first, that agents converge both when retaining partners and when interacting with changing partners. Second, the results show that partner switching results in a setback in performance, but that agents continue to optimize their communication systems. This demonstrates that different dyads indeed converge on different signs for the same concepts. Notably, the setback appears to be smaller for rounds 3 through 7, i.e., through repeated partner switching, agents converge to a more common language.[1]

Overall, the model behaves similarly in many ways to the empirical data; however the initial and final accuracy achieved by the model is consistently lower than the approximately 70% and 95% accuracy (respectively) achieved by human subjects in the Pictionary experiments.

Experiment 2: Director and Matcher roles

Garrod et al. (2007) compared the performance of their participants in a comparable Pictionary task when a single director remained in that role throughout the experiment (single director, SD condition), vs. when participants swapped roles after each round (double director, DD condition). Identification accuracy was slightly higher for the role-swapping, double-director condition than in the single-director condition (significantly so only in the final rounds 5 and 6). This condition is similar to the *isolated pairs* condition in our model. Our model can not only simulate the role-swapping conditions, but also predict contrasts between isolated pairs and communities. The general question here is whether unidirectional communication would be sufficient to develop a community language. So, in this experiment, agents did not switch roles after every concept conveyed, i.e. they remained either director or matcher throughout the game. (Note that, unlike Fay et al.'s experiments and our simulation, Garrod et al.'s study involved feedback about the guesses.)

100 instances of Fay et al's experimental design were run.

Results

Identification accuracy for isolated pairs converged to a higher level than in Experiment 1. Interestingly, communities failed to achieve the same level of accuracy when director and matcher roles were not swapped (Figure 3).

Discussion

This experiment showed that turn-taking is essential for the development of a common community language. Isolated pairs benefit from uni-directional communication (as in Garrod et al's data), presumably converging towards the director's chosen language system. Communities are predicted by the model to require bi-directional communication to converge towards a similarly reliable communication system.

Experiment 3: Noise in Common Ground

A crucial assumption of the compositional semantics in this model is that the agents start out with common knowledge. For instance, both director and matcher need to accept that ambulances and buildings are strongly related to the concept *hospital*. However, the strength of the links between those concepts may differ without precluding the matcher from making the right inference.

The model allows us to test the importance of this assumption and predicts the results of a lower overlap between the knowledge bases of each agent.

Results

Figure 4 shows that mean identification accuracy (7th round, all games) decreases with increased levels of noise in the subsymbolic knowledge state common to the agents. The model appears to deal reasonably well with noise levels of up to 0.3 (coefficient in the noise mixture) for both isolated pairs and communities configurations. This generally holds when taking all rounds into account. (At high noise levels, the initial acquisition of domain signs still works, but agents fail to converge further beyond the initial game or beyond a lower ceiling.) Further work should reveal whether further learning

[1] Note that Figure 2 suggests an effect of condition on the ceiling that is achieved; the regression analysis does not support this. We believe it is due to randomization of the concept order; further work is needed here. Note that in these initial experiments, we simulated only the same number of subjects and communities as in the experiments.

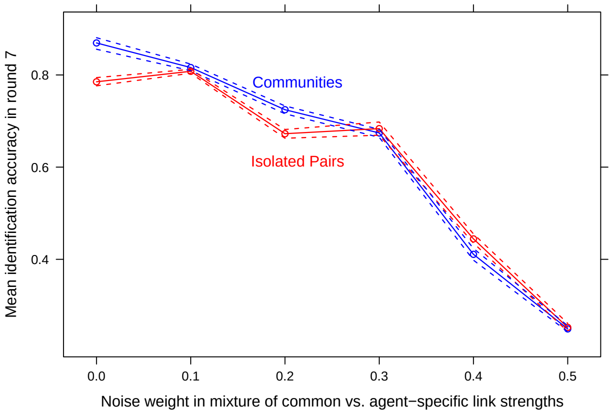


Figure 4: Mean identification accuracy at round 7 is reduced with noise between the knowledge bases of each agent. Bootstrapped 95% confidence intervals.

cycles can make up for the effect, i.e., medium noise levels lead to slower convergence and the failure to converge here is due to the limited number of games.

General Discussion

The model replicates several of the characteristics of the *communities* compared to the *isolated pairs* condition; specifically the set-backs after switching partners for the first few times and the ultimate convergence, despite very limited feedback. We also arrive at a clear prediction: bi-directionality is essential for linguistic convergence in communities.

At this point, we do not attempt to estimate optimal parameters in order to achieve a better fit to the empirical data. We believe that adaptation rates and the convergence ceiling depend both on the difficulty of the task, the specific materials (concepts) and the higher-level reasoning tools employed to optimize the language system. The task in Fay et al.'s experiment structured the list of concepts into a tree (e.g., there were four actors), making the job of drawing and guessing easier. Rather than just drawing what seems most closely related to the target concept, the experimental design invites them to choose a component concept that best disambiguates the drawing in the light of competing concepts (a head and a movie screen may be descriptive of Robert De Niro, but they do not distinguish him from Brad Pitt). Neither specific differentiation nor the precise choice of materials are modeled. Thus, we may overestimate the difficulty of the task. As a further simplifying assumption, our model always produced three component drawings before a guess is made. Garrod et al.'s (2007) design had participants give one another feedback about whether a drawing was thought to be recognized. However, our simplification is not expected to influence the character of the outcome.

Conclusion

We have demonstrated the use of validated, cognitively plausible constraints to explain an emergent, evolutionary group process via multi-agent simulation. Subsymbolic and symbolic learning within a validated human memory framework can account for rapid adaptation of communication between dyads and for the slower acquisition of a domain language in small speaker communities despite very limited feedback about the success of each interaction. Bi-directional communication is predicted to be necessary for a common language system to emerge from communities. The effects are robust against some divergence in prior common ground between agents.

Our model of the horizontal emergence of a common language in multi-agent communities is a first step to a computational, cognitive analysis of the learning processes involved in creating combined signs and acquiring links between them and arbitrary concepts, in order words, the evolution of language. Firm predictions can be drawn from this simulation only once robust convergence in much larger communities can be demonstrated, which will go beyond the empirical data that served as basis for this study.

Acknowledgments We thank Nicolas Fay and Simon Garrod for making their data and manuscripts available and Ion Juvina for comments. This work was funded by the Air Force Office of Scientific Research (FA95500810356).

References

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford, UK: Oxford University Press.

Baayen, R. H. (2001). *Word frequency distributions.* Dordrecht: Kluwer.

Brighton, H., Smith, K., & Kirby, S. (2005). Language as an evolutionary system. *Physics of Life Reviews*, *2*(3), 177-226.

Christiansen, M. H., & Chater, N. (2008). Language as shaped by the brain. *Behavioral and Brain Sciences*, *31*(5), 489-509.

Fay, N., Garrod, S., & Roberts, L. (2008). The fitness and functionality of culturally evolved communication systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *363*(1509), 3553-3561.

Fay, N., Garrod, S., Roberts, L., & Swoboda, N. (in prep.). *The interactive evolution of communication systems.* in preparation.

Garrod, S., & Doherty, G. M. (1994). Conversation, co-ordination and convention: An empirical investigation of how groups establish linguistic conventions. *Cognition*, *53*, 181-215.

Garrod, S., Fay, N., Lee, J., Oberlander, J., & Macleod, T. (2007). Foundations of representation: Where might graphical symbol systems come from? *Cognitive Science*, *31*(6), 961–987.

Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Speech acts* (Vol. 3, p. 41-58). New York: Academic Press.

Griffiths, T. L., & Kalish, M. L. (2007). Language evolution by iterated learning with Bayesian agents. *Cognitive Science*, *31*(3), 441-480.

Kirby, S., & Hurford, J. (2002). The emergence of linguistic structure: An overview of the iterated learning model. In A. Cangelosi & D. Parisi (Eds.), *Simulating the evolution of language* (p. 121-148). London: Springer Verlag.

Pickering, M. J., & Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, *27*, 169-225.

Passing the Test: Improving Learning Gains by Balancing Spacing and Testing Effects

Hedderik van Rijn (D.H.van.Rijn@rug.nl)[a,b], Leendert van Maanen (leendert@ai.rug.nl)[b], Marnix van Woudenberg[b]

[a] Experimental Psychology
University of Groningen

[b] Artificial Intelligence
University of Groningen

Abstract

Where the spacing effect promotes longer intervals between facts that need to be memorized, the testing effect argues for intervals that are short enough to recall the facts. As the ease by which facts are memorized differs greatly between students, an individual assessment of how well certain facts are represented in memory is required to successfully balance spacing and testing effects. We present a model that adapts itself to the abilities of the student, and show in a real-world experiment that this model outperforms other approaches to spacing.

Keywords: spacing-effect; testing-effect; subsymbolic model tracing; cognitive model.

Introduction

The last couple of years have seen a renewed interest in applying insights from fundamental memory research in real-world settings. One of the most visible lines of work are studies to the application of the spacing effect. The spacing effect, first described by Ebbinghaus (1913/1885) at the end of the 19th century, is the positive effect on factual recall that is observed when study trials are temporally separated. Thus, the probability of recall of facts learned in a spaced sequential order (e.g., abcabcabc or abc-*break*-abc-*break*-abc) is higher than the probability of recall of facts that are learned massed (e.g., aaabbbccc). The consequence of this finding is that the presentation sequence of a to-be-memorized list of facts partly determines how well these facts will be recalled on a later test: items on a list that presents the items with wider spacing will be recalled better than items on a list that presents the items as many times as the first list, but massed instead of spaced.

This observation was central to much applied research in the 1960s and early 1970s. Using the possibilities provided by digital computers, scientists tried to construct optimal learning schedules. Although some of this work has stood the test of time from an applied or commercial point of view (e.g., the Pimsleur and Leitner methods are still available commercially), the methods used by these early systems are relatively simple and the learning gains often did not outweigh the extra investment associated with using these systems. This led to a decline in applied research on the spacing effect, although over the decades, more fundamental research on this effect has thrived (for reviews see Dempster, 1988, Cepeda, Pashler, Vul, Wixted, & Rohrer, 2006). Only recently has attention again shifted to using algorithms to determine the optimal schedule for learning (Wozniak & Gorzalanczyk, 1994, Pavlik, 2007, Pavlik & Anderson, 2008).

Another finding that has a potentially large effect on how an optimal sequence has to be constructed, is the testing effect. This effect can be described as: “If students are tested on material and successfully recall or recognize it, they will remember it better in the future than if they had not been tested” [but merely studied the same material] (Roediger & Karpicke, 2006, p.249, see also Carrier & Pashler, 1992). As it is generally assumed that memory decays over time, increasing the interval between successive presentations makes it more likely that an item cannot be recalled. Therefore, spacing beyond a certain interval will be associated with lower learning gains because of failing the testing effect (c.f., the inverse u-shape often observed when the performance on a test is plotted as a function of the interval between two presentations, Cepeda, Vul, Rohrer, Wixted, & Pashler, 2008).

When it comes to computing an optimal presentation sequence for fact learning, spacing and testing have different interests. For the spacing effect, increased spacing is theoretically preferred. But for the testing effect, small to no spacing would theoretically provide the best results. One of the aims of the study reported here is to reconcile these seemingly conflicting requirements.

An interesting observation in almost all work on the spacing effect is that the “optimal schedule” is defined as the schedule that reaches the best performance (often defined as the highest probability of recall) over a longer timeframe. Although this is of course what the goal of all learning *should be*, the goal of learning in a real-world situation is often more pragmatic: passing the next day’s test by studying for a limited, often more or less fixed amount of time. So, although the results of more than a century of spacing results can be used for the real-world situation of having to learn numerous vocabulary word pairs for a foreign language test that is scheduled a couple of weeks or months in advance (c.f., Wozniak & Gorzalanczyk, 1994), these results do not necessarily tell us anything about the pragmatic goal of learning: What method should a student use to learn a set of 20 vocabulary word pairs for a potential test tomorrow, knowing that, because of soccer practice, favorite TV-shows and other homework, all he or she has is 15 minutes to spare?

Note that this real-life situation differs quite a bit from typical experimental setups: First, to prevent effects of prior knowledge, the learning materials in experimental contexts are often selected in such a way that none of the participants has any relevant prior knowledge (by either learning sequences of nonwords, e.g., Ebbinghaus, 1913/1885, very obscure facts, e.g., Cepeda et al, 2008, or word-pairs from languages previously unstudied, e.g., Pavlik & Anderson, 2008). In contrast, when learning for a vocabulary test, most students bring additional knowledge to the learning session from earlier experiences with that language. Second, in most studies the list of word pairs presented to the participants is much longer than the 10 to 30 words that typically have to be learned in a single real-life learning session. Third, the retention interval (defined as the time between the final test on the learned materials and the last study of the materials) in most studies is less than a day (221 out of 254 studies reviewed in Cepeda et al, 2006, used an interval less than a day). Fourth, where many experimental studies aim for finding a general law that describes the effects of different types of spacing on performance in general, the goal of an individual student is not striving for the best performance of a larger group, but for the optimal results on his or her test. As the speed and ease by which vocabulary is learned differs greatly between individuals (e.g., Baddeley, 2003), settings that are optimal for the group as a whole might not be the optimal settings for an individual. These differences are less substantial with respect to the spacing effect than with respect to the testing effect. That is, irrespective of the individual expertise in vocabulary learning, the spacing effect predicts that increased spacing provides better scores. However, with respect to the testing effect, individual differences greatly determine the probability of recall of a particular item. Since successful recall is associated with better learning gains, it is important to account for individual differences in such a way that facts are presented before they cannot be recalled anymore.

To test whether the general findings associated with spacing and testing effects hold when these issues are taken into account, we ran an experiment that closely mimics everyday learning contexts. In this experiment, pre-university level students were asked to memorize Dutch translations of French words in a computer-supported learning session of 15 minutes. During learning, the schedule of presentations of the Dutch-French word pairs was computed according to one of four algorithms.

Algorithm 1 was based on a flashcard strategy: the study items were clustered in sets of 5 which were presented individually until all items in the set had been responded to correctly once. After all sets had been presented, the sequence was started anew until time ran out. **Algorithm 2** is an implementation of the spacing method proposed by Pavlik and Anderson (2005), which will be discussed below. **Algorithm 3 and 4** are adaptations of the original Pavlik and Anderson algorithm in that the model that is used to determine the optimal sequence is dynamically adapted on the basis of the observed performance of the student while taking the testing effect in account. Before turning to these algorithms, we will first discuss Pavlik and Anderson's spacing model and how this model can be applied to provide an optimal learning sequence.

Pavlik & Anderson's Spacing Model

The spacing model proposed by Pavlik and Anderson (2005, referred to as the PA model) is based on the work of Anderson and Schooler (1991). Anderson and Schooler demonstrated that the "availability of human memories for specific items shows reliable relationships to frequency, recency, and pattern of prior exposures to the item" (Anderson & Schooler, 1991, p.396). Eventually, the following formula was proposed to express the availability (or activation) *A* of a certain item *i* at a certain time (*t*) as a function of prior encounters:

$$A_i(t) = \sum_{j=1}^{n} (t - t_j)^{-d_j}$$

According to this equation, which has become central to all memory related models created in the ACT-R cognitive architecture (Anderson, 2007), all previous encounters ($t_1..t_n$) of the item *i* contribute to its current activation. However, the older an encounter (t_j represents the time of encounter *j*), the smaller the contribution of that encounter to the total activation. The speed of this decline is expressed by $-d_j$, the decay parameter. Although initially $-d_j$ was assumed to be variable for different encounters *j* (Anderson and Schooler, 1991, provided an equation to account for some spacing effects but downplayed its importance by noting that "its exact form is a bit arbitrary", p.407), it quickly became a parameter that was treated as a constant (d=.5) as different values for different encounters did not add much explanatory power for most tasks to which this equation was applied. However, in contrast to the original work of Anderson and Schooler, in none of these later tasks was spacing a factor of importance. To account for a broader range of spacing phenomena, the PA model reintroduced individual decay values for individual items.

Pavlik and Anderson proposed to relate the decay values for the individual encounters to the activation of that particular item at the time of the encounter (c.f., Rescorla-Wagner's, 1972, model of learning). As recently presented items have a high activation, the second encounter of an item presented twice in quick succession will be associated with a high decay value. Therefore, the long-term influence of this item will be small as its activation will decay quickly. On the other hand, an encounter of an item of which the last presentation was longer ago (and therefore has a lower activation) will receive a lower decay value, resulting in more long term impact on the activation of that item. The proposed equation calculates the decay, *d*, for encounter *j* of item *i* by calculating the activation of that item (A_i) at the time of encounter *j*.

$$d_{ji} = ce^{A_i(t_j)} + \alpha$$

In this equation, alpha represents the decay intercept. This intercept is the minimum decay for an encounter that will also be used as decay value for the first encounter. The decay scale parameter *c* determines the relative contribution of the activation dependent component. Pavlik and Anderson (2003, 2005, 2008, Pavlik, 2007) have shown in a series of studies that these equations account for a wide range of spacing-related learning phenomena.

In the PA model the activation of a fact determines both the probability of recall of that fact and the latency associated with recalling that fact. For the probability of recall, the activation of the fact is compared to the retrieval threshold while taking into account the noise that is associated with declarative memory. If the activation of a fact is higher than the retrieval threshold, that fact can be recalled. However, if the fact is below the retrieval threshold, it is unavailable for further processing. Apart from the probability of recall, the activation also determines the latency of a retrieval at each point in time (t) according to the following formula:

$$L_i(t) = Fe^{-A_i(t)} + \text{fixed time}$$

In this equation, F is a scaling factor and "fixed time" refers to the time cost of all non-fact-retrieval processes required in giving the answer.

Applying the Spacing Model

In Pavlik and Anderson (2008), the spacing model is used to actively determine the optimal sequence for learning a list of Japanese-English word-pairs. In this paper, Pavlik and Anderson do not explicitly discuss the testing effect (although it is partly accounted for), but instead focus on presenting a sequence of items that have the highest activation gain per second of practice. Thus, the positive effects of increased spacing intervals on the probability of recall are balanced against the negative effects that increased intervals have on accuracy of immediate recalls. This results in a series of complex formulae to determine the learning gains of test-trials and study-trials.

An alternative and simpler approach is to determine the optimal sequence on the basis of the activation of the word-pairs in relation to the retrieval threshold. That is, if we assume on the basis of the combination of spacing and testing-effects that the time between two encounters is optimal *just before* the activation of the fact drops below the retrieval threshold, an optimal sequence can be determined on the basis of the activation of all facts.

Algorithm 2: Default PA

On the basis of the approach discussed above, the default PA model (i.e., pre-2008) can be used to determine the optimal spacing sequence: as soon as a fact is about to fall below the retrieval threshold, it has to be presented again. If no previously presented fact is close to the threshold, a new fact can be introduced. More precisely, as it could be that a fact drops below the retrieval threshold while another fact is being tested, the algorithm computes the activation of all facts 15 seconds ahead to determine whether to introduce a new fact or present a previous one. If all facts have been introduced, the fact with the lowest activation is selected for presentation. The performance of this algorithm is highly dependent on the accuracy of the internal activation representations, which are in turn dependent on the choice of parameter values. Although the PA model has been tested extensively, no fixed set of parameter settings have emerged yet. The values for the decay scale (c) range (Pavlik & Anderson, 2005, 2008) from 0.143 to 0.495, and for the decay intercept (alpha) from 0.058 to 0.300. The threshold parameter is typically set at -0.704. As these parameters have been fit to experiments with longer study session than used in the current experiment, we explored the effects of different settings on the resulting sequences. As a threshold that is too low results in extended spacing (e.g., in our explorations, sometimes all word-pairs were presented before the first word-pair was repeated), we decided to raise the retrieval threshold to -0.500. Following similar reasoning, the decay intercept and the decay scale were set at .25. With respect to the latency equation, we decided against separate estimations for F and the "fixed time". In Pavlik and Anderson (2008), F is set at a value larger than 1 (1.29) indicating an enhanced effect of A_i on the latency. At the same time, using a "fixed time" diminishes the effect of A_i on the latency. Therefore, we set F to 1, and the "fixed time" to 0.

Using the default PA algorithm, we can create an optimal schedule. However, this schedule will be similar for all participants: if the first word-pair is repeated after 5 trials because it will drop below the retrieval threshold within 15 seconds, this holds for all participants. Obviously, this does not match real performance profiles: some participants will have a higher overall performance level than other participants, but it might also be that some words are recalled better by some participants, but a different set of words is recalled better by other participants. However, each time an item is presented the learner provides us with additional behavioral data, which we can use to dynamically adapt the model to the individual learner. This approach can be described as subsymbolic model tracing.

Subsymbolic Model Tracing

In the traditional model tracing account (Anderson, Boyle, Corbett, & Lewis, 1990), the behavior of a student is matched against all knowledge available in a tutoring system. For example, if a student has shown accurate performance in a number of subtraction problems in which carrying is required, the knowledge in the tutoring system that represents carrying is marked as mastered. Thus, the tutoring system keeps a representation of all knowledge the student has mastered by updating the internal representation each time new behavioral information becomes available. The behavior that the learner displays can similarly be used to update the subsymbolic activation of facts (Jastrzembski, Gluck, & Gunzelmann, 2006).

Given that each time a student has to answer a test trial both accuracy and latency information becomes available, we can, in principle, use this information to determine what the current activation of the retrieved chunk is. If we know the latency and therefore the activation at the time of encounter j, and we also know the latency/activation at the time of encounter j-1, we can calculate what the decay for encounter j-1 should have been. By this rationale, we can minimize the difference between the predicted activation and the observed latency and use the behavioral data of the student to update our model that represents the state of the student.

However, given the general assumption that the retrieval process is inherently noisy, using this direct relation might be problematic when the response is fast. That is, when t_j - t_{j-1} is relatively long and the latency for t_j is short because of a temporal boost in activation due to noise, the calculated decay for t_{j-1} will be very low (or even negative). As a very low decay results in facts that are predicted to be highly active over a very long period of time, this temporal noise-boost will ruin the scheduling of the fact. Therefore, we have chosen not to use the outcome of the algorithm described here directly, but instead change the d_{j-1} with a fixed, small amount in the direction indicated by the mismatch between predicted activation and observed latency (c.f., hill-climbing optimization algorithms).

Algorithm 3: Threshold-based Adaptation

Given the issues related to the noisy observations, using the more fine-grained subsymbolic model tracing method described above might result in overfitting. To minimize the chances of overfitting, a coarser algorithm might prove beneficial. Therefore, Algorithm 3 adapts the PA model by only modifying the decay parameter for a certain encounter when at test the word-pair cannot be correctly recalled (c.f., Pavlik & Anderson, 2008). As the system always presents word-pairs of which the estimated activation is above the retrieval threshold, a failure to recall indicates that the estimated activation was too high. Thus, the decay for that particular item should be higher, which is reflected in increasing the alpha parameter with 0.01.

Algorithm 4: Latency-based Adaptation

The threshold-based adaption algorithm focusses on maximizing the testing-effect. Each time a fact cannot be recalled, its decay is increased, ensuring that it will be presented with shortened spacing in subsequent trials. Although this will result in better testing effects because of shorter spacing for facts that could not be recalled, this algorithm does not adapt itself to the inverse situation when facts are better learned than expected. That is, where a failure to retrieve is a marker of lower than expected activation, a faster response than expected is a marker of a higher than expected activation. This idea is captured in the latency-based adaptation of Algorithm 4 which extends the threshold-based adaptation algorithm by comparing the expected latency with the observed latency. To prevent overfitting, the decay intercept is only changed if the difference between expected and observed latency is more than 0.5 seconds. Instead of a constant modifier, the decay intercept is changed according to:

$$\Delta\alpha = \max(0.01, \frac{\text{observed} - \text{expected}}{1000})$$

where observed and expected are the latencies expressed in seconds.

Experiment

Four classes of approximately 15-year old pre-university level pupils were asked to memorize Dutch translations of 20 French words. Each word pair was presented first in a study trial in which both the French and the Dutch word were presented. During a test trial, only the French word was presented, and the participant had to type the Dutch translation. After the initial presentation, the next trial was scheduled on the basis of one of the four algorithms discussed above.

Procedure Study trials were presented for 5 seconds. After each initial study trial, a test trial of the same word-pair was presented. During a test trial, only the French word was presented and students had 15 seconds to reply by typing in the correct Dutch translation. After pressing Enter, students were presented with a 2-second feedback screen stating “Correct”, “Incorrect” or “Almost correct” (which was given if the Levenshtein-distance to the correct answer was smaller than 3). If the participant did not respond in time, or an incorrect answer was given, the study trial was presented to refresh the participant’s memory. The four algorithms determined which word pair to present next. The learning session lasted 15 minutes, irrespective of the number of trials or words presented. After the learning session on Day 1, all words were tested by means of a traditional paper-and-pencil test on Day 2. The post-test was graded on a scale from 0 to 10. Each incorrect response deducted .5 point from the maximum score of 10. Participants were naive with respect to the experimental manipulation and did not know that they would be tested on Day 2.

On Day 1, participants were tested in groups in a classroom equipped for computer-supported education. Each participant operated his or her own computer. The paper-and-pencil test on Day 2 was conducted during normal class-hours.

Materials A list of 20 words was compiled for each class separately. All words were selected from a textbook chapter that would not be discussed until one week after the experiment.

Participants Ninety-one pre-university-education level students (all students of four 3rd year HAVO/VWO classes) of approximately 15 years of age participated, of which 85 took part in both tests. Participants were semi-randomly distributed over conditions to ensure that in each class an equal number of participants used each algorithm. All participants were instructed that their results would be stored anonymously and that the results would not be communicated to their school or teachers on an individual level.

Results

Of the 85 students who took part in both sessions, six students were removed from further analyses because they did not respond in more than 5% of all trials and gave a number of answers that did not fit the instructions (e.g., “I’m bored”, or names of rock bands). Four participants were removed because their performance in terms of correct responses during the learning session deviated more than 2 standard deviations from the average of their group. One participant was removed because of scores on the final test that deviated more than 2 standard deviations from the average score for his or her group. This leaves 74 participants, 18 in the flashcard condition, and 19 participants in each of the three spacing conditions.

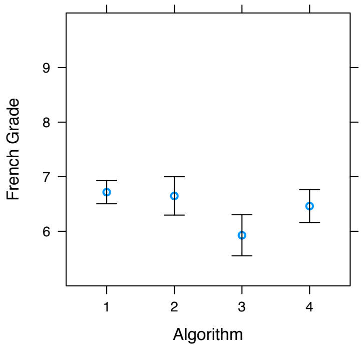


Figure 1: Average grades on French per algorithm condition

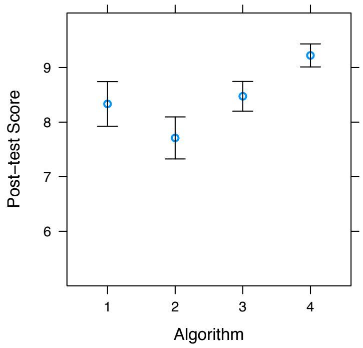


Figure 2: Raw scores on post-test per algorithm condition

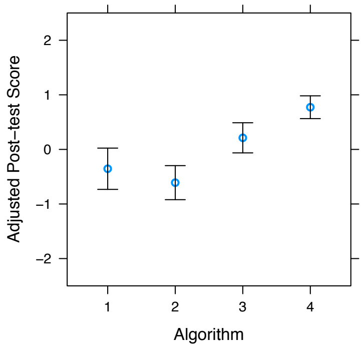


Figure 3: Post-test scores adjusted for covariates mentioned in text

All errorbars depict standard errors.

Covariates As we tested participants in a domain in which they have a significant amount of prior knowledge, it is important to control for potential differences in prior knowledge between groups. Hereto, we analyzed the students' school grades for French (graded on a theoretically linear scale from 0 to 10, with a 6 representing the grade required to pass that class), see Figure 1 (F(3,73)= 1.27, p=0.29). Although this effect is far from significant, the value of the F-statistic is larger than we hoped for. Therefore, we decided to include the grades for French as covariate in all subsequent analyses.

Given that we limited the amount of time to learn 20 word-pairs and the algorithm determined when a new word pair was introduced, not all participants might have seen all 20 word pairs in the non-flashcard conditions (algorithm 2 to 4). This did indeed turn out to be the case in all three conditions. The average number of word pairs presented to the participants was 19.5, 19.6 and 19.8 for the default PA, the threshold-based and the latency based conditions respectively. Although these differences (when compared to the 20 words seen by the students in the flashcard condition) fail to reach significance (ANOVA F(3,69)=2.6, p=0.057, post-hoc pairwise t-tests with pooled standard deviations: flashcard vs default PA algorithm, p=0.08, all other comparisons p > .1), this does give the flashcard-based condition an advantage when comparing scores on the post-test, as some participants in the other conditions will not have seen all word pairs. Therefore, we also included the number of words seen by the student as covariate in subsequent analyses.

To account for possible effects associated with the session in which the study was run or peculiarities of a particular class, another factorial covariate was included representing group.

Post-test Scores Figure 2 shows the raw scores on the post-test, and Figure 3 shows the scores on the post-test adjusted by the covariates French grade, group and number of words seen.

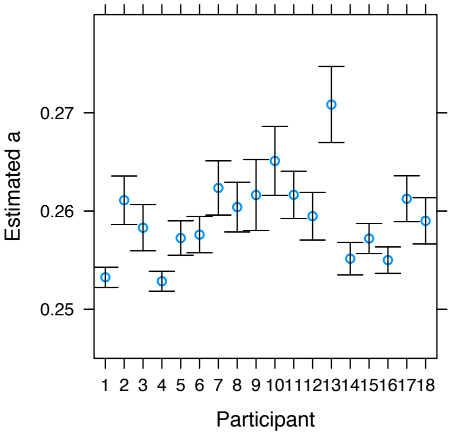


Figure 4: Effects of adaption per participant

Analyzing the data presented in Figure 3 shows that the algorithm has a significant effect on the post-test scores (F(3,70)=4.19, p=0.009). Testing the individual effects by conducting pairwise comparisons using t-tests with pooled standard deviation and Benjamini and Hochberg's (1995) p-value adjustment method showed that students in the latency-adaptation group, Algorithm 4, score significantly higher than students in the flashcard (p=0.032) or in the PA model (p=0.010) group. None of the other comparisons reached significance (p>0.100).

Adaptions The observed differences between the more static PA model (Algorithm 2) and the latency adaptation condition (Algorithm 4) suggests that the adaptations resulted in different decay patterns for different participants. Figure 4 shows the average estimation of the alpha parameter associated with the last encounter per word-pair. As can be seen, different participants required different alphas, with, for example, participant 1 and 4 requiring relative low decay values and participant 13 requiring a very high decay value. If these three participants would have been set at the average alpha (0.259), the estimated activation for participant 13 would be too high, resulting in many retrieval failures - and violating the testing-effect constraints. At the same time, participants 1 and 4 would have had a too low estimated activation, resulting in a sequence with too low spacing, violating the spacing-effect constraints.

Discussion

The current study set out to answer two questions. The first was to test whether the results obtained in the scientific literature on the spacing effect would also hold in the more real-life case of learning a small set of vocabulary items in a small period of time. The second question was to examine whether the learning gain would improve when the algorithms that construct the learning sequence take individual differences into account. With respect to the first goal, the significant difference between the flashcard and the latency adaption conditions illustrates that a learning sequence that is based on an algorithm that takes spacing and testing-effects into account outperforms a more traditional flashcard sequence. However, *only* the condition that optimizes the sequence on the basis of individual latency differences significantly outperforms the flashcard condition, answering the second question.

It is striking to see that the default (pre-2008) PA spacing condition scores - in absolute terms - worse than the flashcard condition. This result for the default PA spacing condition might be caused by the parameter settings chosen for this study: alternative parameter settings might improve the PA model. However, it is difficult to come up with the parameter settings required. The first candidate for change would be the retrieval threshold, as in most PA studies the threshold is set at -0.704 instead of -0.5. However, decreasing the threshold would increase the spacing between two presentations of the same item. This will probably have a negative effect on the data as the sole difference between the default PA algorithm (2) and the threshold adaptation algorithm (3) is *decreased* spacing and algorithm 3 fares considerably better than the PA algorithm. With respect to changes in the parameters involved in calculating d_{ji}, it is most likely that these changes would benefit the other algorithms as well. Thus, although changes in the parameter settings might diminish the gap between the different spacing algorithms, it is hard to imagine how the default PA model would outperform the alternative algorithms proposed here.

With respect to Algorithm 3 and 4, although the differences in performance are not significant, the performance profiles favor the latency-based Algorithm 4 over the accuracy-based Algorithm 3. This suggests that Pavlik and Anderson's 2008 implementation might be further refined by incorporating the information that can be deduced from the latencies (c.f., Pavlik, Presson, & Koedinger, 2007).

Finally, it is interesting to note that Pavlik and Anderson (2008, p.102) discuss a very similar approach they call "performance tracking" and mention that this method will "add considerable power". Nevertheless, they conclude that this approach will make scheduling much more complex.

In this study we have shown that performance tracking is possible, but also that adapting the sequence to the characteristics of individual learners improves learning gains considerably, even if the learning session takes only 15 minutes.

Acknowledgments

Parts of this work has been conducted in the context of Van Woudenberg's (2007) Master thesis. The authors would like to thank the students and teachers of het Belcampo College, het Gormarus College and het Werkman College for their participation and Philip Pavlik and two anonymous reviewers for helpful remarks regarding the argument and clarity of the paper.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R., Boyle, C. F., Corbett, A., & Lewis, M. W. (1990). Cognitive modelling and intelligent tutoring. *Artificial Intelligence, 42*, 7–49.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2 (6), 396–408.

Baddeley, A. (2003). Working memory: looking back and looking forward. *Nature Reviews Neuroscience*, 4 (10), 829–839.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, 289–300.

Carrier, M., & Pashler, H. (1992). The influence of retrieval on retention. *Memory and Cognition*, 20, 633–633.

Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132 (3), 354.

Cepeda, N. J., Vul, E., Rohrer, D., Wixted, J. T., & Pashler, H. (2008). Spacing effects in learning: A temporal ridgeline of optimal retention. *Psyc.Science*, 19, 1095-1102.

Dempster, F. N. (1988). The spacing effect. *American Psychologist*, 43, 627–634.

Ebbinghaus, H. (1885/1913). *Memory: A contribution to experimental psychology*. Translated by Henry A. Ruger and Clara E. Bussenius (1913).

Jastrzembski, T. S., Gluck, K. A., & Gunzelmann, G. (2006). Knowledge tracing and prediction of future trainee performance. In *Proceedings of the 2006 interservice/ industry training, simulation, and education conference* (p. 1498-1508). National Training Systems Association.

Pashler, H., Rohrer, D., Cepeda, N., & Carpenter, S. (2007). Enhancing learning and retarding forgetting: Choices and consequences. *Psychonomic Bulletin & Review*, 14 (2).

Pavlik Jr, P. I. (2007). Understanding and applying the dynamics of test practice and study practice. *Instructional Science. 35*, 407-441

Pavlik Jr, P. I., & Anderson, J. R. (2003). An ACT-R model of the spacing effect. In Proceedings of the 5th international conference of cognitive modeling (pp. 177–182).

Pavlik Jr, P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29 (4).

Pavlik Jr, P. I., & Anderson, J. R. (2008). Using a model to compute the optimal schedule of practice. Journal of Experimental Psychology: Applied, 14 (2), 101–117.

Pavlik Jr., P. I., Presson, N., & Koedinger, K. R. (2007). Optimizing knowledge component learning using a dynamic structural model of practice. In R. Lewis & T. Polk (Eds.), Proceedings of the eighth international conference of cognitive modeling. Ann Arbor: University of Michigan.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II* (p. 64-99). Appleton-Century-Crofts: Appleton-Century-Crofts.

Roediger, H. L., & Karpicke, J. D. (2006). Taking memory tests improves long-term retention. *Psychological Science*, 17 (3), 249–255.

Van Woudenberg, M. (2008). *Optimal word pair learning in the short term: Using an activation based spacing model*. Unpublished master's thesis, University of Groningen.

Wozniak, P. A., & Gorzelanczyk, E. J. (1994). Optimization of repetition spacing in the practice of learning. *Acta Neurobiologicae Experimentalis*, 54 (1), 59–62.

Towards a New Cognitive Hourglass: Uniform Implementation of Cognitive Architecture via Factor Graphs

Paul S. Rosenbloom (Rosenbloom@USC.Edu)
Department of Computer Science and Institute for Creative Technologies, 13274 Fiji Way
Marina del Rey, CA 90292 USA

Abstract

As cognitive architectures become ever more ambitious in the range of phenomena they are to assist in producing and modeling, there is increasing pressure for diversity in the mechanisms they embody. Yet uniformity remains critical for both elegance and extensibility. Here, the search for uniformity is continued, but shifted downwards in the cognitive hierarchy to the *implementation level.* Factor graphs are explored as a promising core, with initial steps towards a reimplementation of Soar. The ultimate aim is a uniform implementation level for cognitive architectures affording both heightened elegance and expanded coverage.

Keywords: Cognitive architecture; implementation level; factor graphs; graphical models; production match; Soar

From Architecture to Implementation

A *cognitive architecture* is a hypothesis about the fixed structures underlying thought in active intelligent beings, whether natural or artificial. It consists of a set of interacting mechanisms that can combine with appropriate knowledge to model human intelligent behavior and/or generate artificial intelligent behavior. In the large, a cognitive architecture is a theory about one or more *systems levels* comprising an intelligent being. Newell (1990) discussed a hierarchy of levels (organelles, neurons, neural circuits, deliberate acts, operations, etc.) across four bands of human action: biological, cognitive, rational, and social. At each level, a combination of structures and processes implements the basic elements at the next higher level.

One controversial attribute of systems levels in cognitive architecture is their *girth*; i.e., their uniformity versus diversity. Diversity always exists across levels, but individual levels may consist of anything from a small number of very general elements to a wide diversity of more specialized ones. Uniformity appeals to simplicity and elegance. In caricature, it is the physicist's approach, where a broad diversity of phenomena emerges from interactions among a small set of general elements. Diversity appeals to specialization and optimization. It is the biologist's approach, in which many specialized structures, each locally optimized, jointly yield a robust and coherent whole.

Across a hierarchy of levels, there is no a priori reason to assume they are all of comparable girth. While physicists and biologists may expect uniformity within their fields, the networking community trumpets the *Internet hourglass* to explain their protocol stack (Deering, 1998). At the narrowed waist is the Internet Protocol (IP). Above is an increasingly diverse sequence of levels enabling "everything on IP". Below is an increasingly diverse sequence of levels enabling "IP on everything". The hourglass yields a diversity of applications and implementations that are united via a core of *mesoscale uniformity*. Domingos's (In press) recent call for an *interface layer* in AI is an appeal for a similar sort of mesoscale uniformity in AI.

Intelligence clearly entails diversity in the cognitive hierarchy across levels, but what about within levels? At the top, the extraordinary range of possible behaviors and applications is one of the core phenomena cognitive architectures are developed to explain. At the bottom, the mind is grounded in the diverse biology of the brain and, at least according to *strong AI*, could also be grounded in a diversity of alternative technologies (with adjustments in Newell's lower levels for grounding in such technologies). But is there an hourglass or a rectangle in between?

The question of the existence of a *cognitive hourglass* has traditionally been cast in terms of whether the cognitive architecture is uniform. Among architectures for cognitive modeling, Soar (Rosenbloom, Laird & Newell, 1993) has been a standard exemplar of uniformity and ACT-R (Anderson, 1993) of diversity. Recently, based on both functional and modeling considerations, Soar 9 (Laird, 2008) has shifted strongly towards diversity, and is helping to tip the community balance in this direction.

As a scientist, one can respond to a demonstrated need for diversity by simply accepting it, or by hypothesizing an underlying uniformity and simplicity that explains it. Anderson, for example, developed a background theory of cognitive rationality to justify ACT-R's mechanisms as optimal adaptations to the environment (Anderson, 1990). The theory's uniformity is not in the architecture itself, but does yield a simple, well-motivated explanation for it. Yet something significant is lost when the uniformity is not in the cognitive hierarchy, as diversity negatively impacts both the elegance of the resulting system and the ease with which new capabilities can be integrated into a unified whole. Historically, diverse architectures have been tough to unify. To the extent such a system remains disunified, it is more of a toolkit or language than a hypothesis about the fixed structures of thought (i.e., an architecture).

Another alternative is not simply to accept diversity, or try to justify it, but to continue a search for uniformity – the narrow waist of the hourglass – elsewhere in the cognitive hierarchy. This is an application of the *uniformity-first* research strategy (a variant of *Ockham's razor*): begin by assuming uniformity and accept diversity only upon overwhelming evidence. To the extent uniformity is possible, it yields elegance and facilitates unification and extension. Beginning instead with diversity removes the pressure to search for hidden commonality, and may lead down an irrevocable path of complexity.

The history of Soar well illustrates the uniformity-first strategy (Laird & Rosenbloom, 1996). For years it had a single procedural, rule-based, long-term memory and a single learning mechanism (Laird, Rosenbloom & Newell, 1986), while investigations continued into their ability to support a diversity of memory (e.g., procedural, semantic, and episodic (Rosenbloom, Newell & Laird, 1991)) and learning (e.g., skill and knowledge acquisition, generalization and transfer, and learning from observation (Rosenbloom, 2006)) behaviors. A wide range of such behaviors proved feasible, but they never could be fully unified with the rest of the system to yield pervasive utility across all activity. This evidence against the existing uniformity, amassed over years of experimentation, inspired the development of Soar 9, a diverse architecture that adds new long-term memories (semantic and episodic) and learning mechanisms (semantic, episodic and reinforcement), while also incorporating other new capabilities (emotion and imagery) (Laird, 2008).

Uniformity-first, however, entails that acknowledging a need for diversity at the architecture level should be accompanied by a continued search for uniformity at other levels. In this article, the particular focus is on burrowing beneath the diversity at the architecture level to look for uniformity at the *implementation level* just below. The goal is still an hourglass, albeit one with a lower waistline.

The implementation of cognitive architectures, while critical for efficiency and usability, is usually extra-theoretic and not part of the architectural hypothesis. Characteristic examples include the COGENT (Cooper & Fox, 1998) and Storm (Pearson, Gorski, Lewis & Laird, 2007) environments for cognitive modeling/architectures, both of which are coarse-grained, graphical tools intended to assist the developer rather than theoretical hypotheses about the implementation level. The primary exception is systems like SAL (Jilk, Lebiere, O'Reilly & Anderson, 2008) or Neuro-Soar (Cho, Dolan & Rosenbloom, 1991), where a cognitive architecture is implemented via neural networks. Neural approaches remain interesting possibilities for the implementation level, but the focus here is on the related but more general class of *graphical models* (Jordan, 2004).

Graphical models share the core graphical/network nature of neural networks and graphical modeling environments, but focus on representing independence across variables in complex functions such as joint probability distributions and communication codes. They include Bayesian networks (Pearl, 1988) and Markov networks, with origins in probabilistic reasoning. But they also include factor graphs (Kschischang, Frey & Loeliger, 2001), which take a broader multivariate-function view. Interestingly, a variety of neural network algorithms – such as supervised Boltzmann machines, radial basis functions, and unsupervised learning algorithms – can be mapped onto graphical models (Jordan & Sejnowski, 2001). A core premise of this article is that graphical models provide untapped potential for cognitive architectures. They may also ultimately forge a principled bridge between neural and symbolic architectures.

The work in this article is based on factor graphs. Although originating in coding theory, where they underlie the "astonishing performance" of turbo codes, factor graphs are particularly promising for cognitive architecture because of the diversity of important problems and algorithms they subsume in a uniform manner when combined with their canonical sum-product algorithm. Factor graphs are relevant for *signal processing*, where they are useful in vision (Drost & Singer, 2003) and subsume Kalman filters, the Viterbi algorithm, and the forward-backward algorithm in hidden Markov models; *probabilistic processing*, where they subsume belief propagation in Bayesian and Markov networks; and *symbol processing*, where they yield arc consistency for constraint problems (Dechter & Mateescu, 2003). There is also significant work on *mixed* approaches combining symbolic and probabilistic processing, *hybrid* approaches combining discrete and continuous processing, and *hybrid mixed* approaches (Gogate & Dechter, 2005).

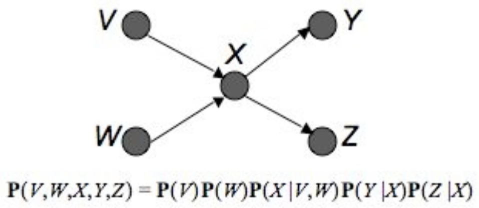
Factor graphs raise the possibility of a uniform implementation level that elegantly explains the diversity seen in existing cognitive architectures while going beyond them to yield an effective and uniform basis for: unifying cognition with perception and motor control, breaking down the barriers between central and peripheral processing by bringing the latter within the cognitive inner loop and making each potentially penetrable by the other; fusing symbolic and probabilistic reasoning to provide general reasoning under uncertainty; and providing a conceptual bridge from symbolic to neural architectures, by mapping them onto a common intermediary. They provide a tantalizing combination that is particularly appropriate at the implementation level of: (1) *generality*, in the range of capabilities they can uniformly support in a state-of-the-art manner; and (2) *constraint*, in the ways that these capabilities can reasonably be supported.

The remainder of this article introduces factor graphs, begins exploring their utility for cognitive architectures via a first step towards a graphical reimplementation of Soar, and summarizes the path forward from here. The focus is not on a specific cognitive model, but on the possibility of a radically new generation of integrated cognitive models.

Factor Graphs

Factor graphs provide a form of divide and conquer with nearly decomposable components for reducing the combinatorics that arise with functions of multiple variables. The function could be a joint probability distribution over a set of random variables; e.g., $\mathbf{P}(V,W,X,Y,Z)$, which yields the probability of $V{=}v \wedge W{=}w \wedge X{=}x \wedge Y{=}y \wedge Z{=}z$ for every value v, w, x, y and z in the variables' domains. Or the function could represent a constraint satisfaction problem, $\mathbf{C}(A,B,C,D)$, over a set of variables, yielding 1 if a combination of values satisfies the constraints and 0 otherwise. Or the function could represent a discrete-time linear dynamical system, as might typically be solved via a Kalman filter. The problem formulation here would involve a *trellis* structure, where the graph for one time step is repeated for each, with four variables per time step – *State*, *Input*, *Output* and *Noise* (Kschischang, Frey & Loeliger, 2001) – $\mathbf{K}(S_0,I_0,O_0,N_0,\ldots, S_n,I_n,O_n,N_n)$.

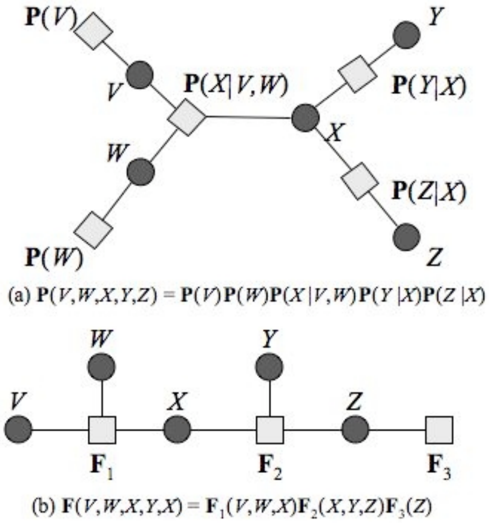
The prototypical factor graph operation is the computation of *marginals* on variables. For a joint probability distribution, this is simply the marginal of a random variable, computed by summing out the other variables: $\mathbf{P}(Y) = \Sigma_{v,w,x,z}\mathbf{P}(v,w,x,Y,z)$. The key to tractability is avoiding the explicit examination of every element of the cross product of the variables' domains. For probabilities, the joint distribution is decomposed into a product of conditional (and prior) probabilities over subsets of variables: $\mathbf{P}(V,W,X,Y,Z) = \mathbf{P}(V)\mathbf{P}(W)\mathbf{P}(X|V,W)\mathbf{P}(Y|X)\mathbf{P}(Z|X)$. Such decompositions derive from the *chain rule* plus *conditional independence* assumptions. Using commutative and distributive laws then enables more efficient marginal computation: $\mathbf{P}(Y) = \Sigma_x\mathbf{P}(Y|x)\Sigma_z\mathbf{P}(z|x)\Sigma_v\mathbf{P}(v)\Sigma_w\mathbf{P}(x|v,w)\mathbf{P}(w)$. This provides the basics of Bayesian networks (Figure 1).



$\mathbf{P}(V,W,X,Y,Z) = \mathbf{P}(V)\mathbf{P}(W)\mathbf{P}(X|V,W)\mathbf{P}(Y|X)\mathbf{P}(Z|X)$

Figure 1. Example Bayesian network

Factor graphs generalize this to arbitrary multivariate functions; e.g., $\mathbf{F}(V,W,X,Y,Z) = \mathbf{F}_1(V,W,X)\mathbf{F}_2(X,Y,Z)\mathbf{F}_3(Z)$. The function becomes a bipartite graph, with a *variable node* for each variable, a *factor node* for each function use, and undirected links between factors and their variables (Figure 2).



(a) $\mathbf{P}(V,W,X,Y,Z) = \mathbf{P}(V)\mathbf{P}(W)\mathbf{P}(X|V,W)\mathbf{P}(Y|X)\mathbf{P}(Z|X)$

(b) $\mathbf{F}(V,W,X,Y,X) = \mathbf{F}_1(V,W,X)\mathbf{F}_2(X,Y,Z)\mathbf{F}_3(Z)$

Figure 2. Example factor graphs

The core inference algorithm for factor graphs is the *sum-product* (aka *summary-product* or *belief-propagation)* algorithm, which passes messages along links. A message from a source node to a target node along a link summarizes the source node's information about the domain of the link's variable node. A message from a variable node to a factor node is the pointwise product of the messages into the variable from all of its neighbors except the target node. A message from a factor node to a variable node starts with this same product but also includes the factor node's own function in the product, with all variables other than the target variable then being summed out to form the outgoing message. A key optimization here, as in Bayesian networks, is to use the commutative and distributive laws to redistribute multiplicative factors outside of summations.

For tree-structured graphs in which only a single marginal is desired, the factor graph can be reduced to an *expression tree* in which the products and sums are computed unidirectionally upwards in the tree. Beyond this simplest case, the algorithm works iteratively by sending output messages from nodes as they receive input messages. For *polytrees*, which have at most one undirected path between any two nodes, this iterative algorithm always terminates and yields the correct answer. For arbitrary graphs with loops, neither correct answers nor termination are guaranteed. However, it does often work quite well in practice, as has been most strikingly evident for turbo codes.

The sum-product algorithm uses two specific arithmetic operations: sum and product. However, the same generic algorithm works for any pair of operations forming a *commutative semi-ring*, where: both operations are associative and commutative and have identity elements; and the distributive law exists. Max-product, for example, is key to computing maximum a posteriori (MAP) probabilities. OR-AND also works, as do other pairs.

To improve the efficiency of the algorithm, various optimizations can be applied, and alternative algorithms can be used (such as survey propagation (Mézard, Parisi & Zecchina, 2002) and Monte Carlo sampling (Bonawitz, 2008)). A connection exists between factor graphs and statistical mechanics, revealing that the sum-product algorithm minimizes the *Bethe free energy*, and yielding algorithmic innovations (Yedidia, Freeman & Weiss, 2005).

Factor Graphs for Cognitive Architecture

The key question for us is whether factor graphs can yield a uniform level for implementing, understanding and exploring cognitive architecture, while ultimately yielding novel architectures that are more uniform, unified, and functional. Existing work on hybrid mixed methods is encouraging, as is work on general languages for mixed probabilistic and logical reasoning. FACTORIE (McCallum, Rohanemanesh, Wick, Schultz & Singh, 2008), for example, combines factor graphs with an imperative programming language to support relations and other capabilities, while BLOG (Milch, Marthi, Russell, Sontag, Ong & Kolobov, 2007) and Alchemy (Domingos, Kok, Poon, Richardson & Singla, 2006) combine probability and logic via Bayesian and Markov networks, respectively.

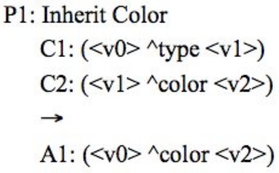
The particular approach advanced here is to: (1) re-implement existing architectures to help understand factor graphs, existing architectures, and the implications of implementing architectures via factor graphs; (2) go beyond existing architectures by hybridization and simplification, both within and across architectures; and (3) integrate in new capabilities that don't mesh well with existing architectures, such as perception and motor control.

The initial focus is on Soar because of familiarity with it, its history in cognitive modeling, and its dual status as both a uniform (early) and a diverse (latest) architecture. We can make a quick start on reimplementing the uniform core, and build towards a more uniform integration of later diversity.

The inmost core of "uniform Soar" is the *reactive layer*, where working memory (WM) is elaborated via associative retrieval of relevant information from a parallel production system. During a single *elaboration cycle*, match computes all legal production instantiations, which then fire in parallel to modify WM. Match is the core of the elaboration cycle, so it is the natural focus for initial reimplementation efforts.

In Soar, match is based on *Rete* (Forgy, 1982), comprising a discrimination network for sorting working memory elements (wmes) to production conditions, a join network to determine which combinations of wmes yield production instantiations (while attending to across-condition variable equality), and support for both incremental match across cycles and shared match across productions. Most individual productions match efficiently, although worst-case match cost is exponential in the number of conditions.

A factor graph implementation of Rete has been designed, where factor nodes handle discriminations and joins, variable nodes represent wmes that match production conditions and their combinations – analogous to Rete's α and β *memories* – and unidirectional message passing over an expression tree enables incremental and shared match. But, rather than imposing Rete on factor graphs, the primary focus here has been on algorithms that arise more naturally from viewing production match as a multivariate function.



P1: Inherit Color
C1: (<v0> ^type <v1>)
C2: (<v1> ^color <v2>)
→
A1: (<v0> ^color <v2>)

Figure 3. Example rule

Consider the rule in Figure 3. This is not exactly Soar's representation, although it does retain its object-attribute-value scheme, with conditions testing wmes via constants and variables (in angle brackets). The simplest mapping of this production to a factor graph views it as a Boolean function of the three production variables – $\mathbf{P_1}(v_0,v_1,v_2)$ – which, for each combination of variable values, yields 0 or 1 depending on whether the combination defines a legal instantiation. The production's conditions then specify how the function is to factor: $\mathbf{P_1}(v_0,v_1,v_2) = \mathbf{C_0}(v_0,v_1)\mathbf{C_2}(v_1,v_2)$ (Figure 4).

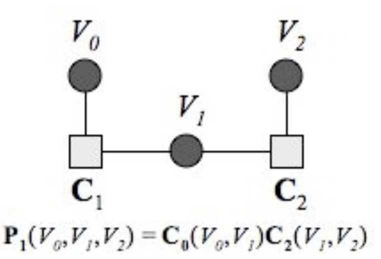


Figure 4. Example rule graph

This mapping has been implemented. In it, WM is a 3D Boolean array – objects × attributes × values – with 1s for every wme in WM and 0s elsewhere; and messages are Boolean vectors with 1s for valid bindings of the link's variable and 0s elsewhere. In essence, productions define graphs while WM defines distributions over graph variables.

This initial approach showed the feasibility of implementing match via factor graphs, but it also raised three issues: (1) both WM and tests of constants were hidden within the condition factors; (2) production match ignored conditions without variables; and (3) it led to errors from *binding confusion* (Tambe & Rosenbloom, 1994). Solutions for these issues have been implemented, but as the first one didn't affect correctness – only how much factor graphs were leveraged – and the second couldn't actually occur in Soar because all of its conditions must be linked via variables, only the third issue is discussed here.

Binding confusion arose because the graph independently tracked the legal bindings of each variable – called *instantiationless match* in (Tambe & Rosenbloom, 1994) – rather than maintaining Rete's explicit combinations of condition instantiations. Suppose (A ^type B), (C ^type D), (B ^color Red) and (D ^color Blue) are in working memory. The match bound v_0 to A & C, v_1 to B & D, and v_2 to Red & Blue, but it couldn't, for example, distinguish which color (v_2) to associate with object A (v_0), despite the fact that a correct match would mandate Red rather than Blue.

This problem is a direct consequence of the interaction between two types of constraint imposed by factor graphs: (1) the *locality* of processing in the network; and (2) the limitation on message content to the values of one variable. Approaches to binding confusion must either work around these constraints to yield correct combinations or redefine match to live within them. Correct combinations can be yielded, for example, by post-extraction (Dechter & Pearl, 1987) or by implementing Rete. If instead match is to be redefined to be what is produced, we must then determine how to write rules that yield the desired overall behavior given the new semantics. This approach could also be further refined by replacing Boolean array values with apportioned fractional values for ambiguous bindings.

The most promising approach at this point modestly redefines the semantics of match to produce the needed combinations of bindings for action variables, while still avoiding the need for Rete's full instantiations. In the process, it eliminates binding confusion, alters the worst-case match cost for a production to exponential in its *treewidth*, and further reduces costs and potential confusion by eliminating redundant instantiations that would otherwise generate equivalent results (when some condition-variable bindings differ while action-variable bindings do not).

This approach enables local processing of variable combinations by using variable nodes in the graph that represent combinations of production variables rather than individual ones. To start, an ordering is imposed on the production's conditions and actions to yield a sequence of factor nodes. A variable node is then added between each successive pair of factor nodes. To finish, the first and last condition/action that uses each production variable is determined, and that variable is added to each variable node between these two factor nodes (Figure 5). The approach is based on *stretching* in factor graphs, which itself maps onto junction trees (Kschischang, Frey & Loeliger, 2001). Its implementation eliminates binding confusion by tracking combinations of variable bindings just as they are needed.

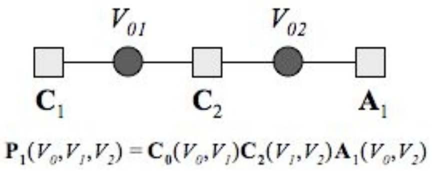

V_{01}
V_{02}
$\mathbf{C}_1$
$\mathbf{C}_2$
$\mathbf{A}_1$
$\mathbf{P}_1(V_0,V_1,V_2) = \mathbf{C}_0(V_0,V_1)\mathbf{C}_2(V_1,V_2)\mathbf{A}_1(V_0,V_2)$

Figure 5. Modified rule graph

Since each variable node in the graph may now represent multiple production variables, multi-dimensional arrays result that can be expensive to process without further optimization. The most critical optimization here is factor rearrangement. Without it, the full factor graph for the rule in Figure 3 – comprising 8 factor nodes and 8 variable nodes when all three problem solutions are included plus the *goal memory* to be described later – exhausts heap space before match completes (in LispWorks PE). With factor rearrangement, match takes only 1.7 sec.

A second critical optimization leverages the uniformity of WM and message arrays (which are almost all 0s or 1s) via an N-dimensional generalization of *region quad/octrees* (à la CPT-trees in Bayesian networks (Boutilier, Friedman, Goldszmidt & Koller, 1996)). If an array is uniform, it becomes a single-valued unit. Otherwise, each dimension is bisected – yielding 2^N sub-arrays – and the process recurs. The sum and product algorithms are trickier here, but have been worked out. With this optimization, match time is reduced by a further factor of ~7 (from 1.7 to .25 sec.). It also enables comparing match cost without rearrangement, yielding a factor of ~500 (132 vs. .25 sec.).

One interesting implication of representing WM via trees is a view of it as a piece-wise constant function. If this proves extensible to piece-wise linear functions, it may be effective for variables with continuous domains and ranges (as used in mixed and hybrid systems). It may also be possible to employ more intelligent partitioning algorithms for WM, including adaptive clustering methods.

Conclusion and Next Steps

Despite the increasing trend towards diversity in cognitive architectures, uniformity at the implementation level may yet provide leverage in exploring, understanding and improving existing architectures; and in developing novel architectures with increased elegance and broader coverage. Factor graphs, and graphical models more generally, are intriguing for this level because they yield a wide diversity of capabilities in a uniform and constrained manner.

An initial step has been taken towards reimplementing Soar by factor graphs, with the demonstration of the latter's ability to implement (symbolic) production match via an interesting new algorithm. The key next step is extending beyond match to the rest of Soar's cognitive inner loop – the *deliberate layer* (or *decision cycle*) – where elaboration cycles repeat until quiescence (the *elaboration phase*) followed by a decision. One approach to the elaboration phase is to alter WM between cycles, as in standard production systems. This has been implemented, but a more promising alternative is to arrange the elaboration phase's temporal structure in space rather than time, as a trellis. With a trellis, perceptual and motor processing may be integrated directly into the cognitive inner loop rather than being walled off into a separate I/O system. A trellis would also enable bidirectional information propagation across the elaboration phase to ensure correct graphical probability calculations. For the process of decision making itself, *influence diagrams* are a natural strategy to explore first.

Beyond reimplementing Soar's cognitive inner loop is the challenge of extending the loop to include a more uniform integration of Soar 9's semantic and episodic memories, plus probability and signal processing. The lead candidate for semantic memory blends Prolog's view of facts as condition-less rules that are triggered backwards by a goal probe, with the statistical view of retrieving the most probable semantic memory element given the probe (Anderson, 1990). A goal memory – in analogy to working memory – has been implemented to enable backwards access to production actions; but appropriate control of backwards vs. forward processing in the inner loop is still needed, as is restricting retrieval to the most probable element (based on MAP probabilities and the max-product algorithm). For episodic memory, two approaches have potential: (1) adding long-term trellises to the graph; and (2) extending WM to a fourth, temporal dimension.

Adding probabilities to the inner loop is being explored via experiments with extant mixed languages, such as Alchemy and BLOG (Rosenbloom, 2009). Signal processing will be investigated via trellises and piecewise-linear quad/octrees (for representing continuous functions).

Still, this is all only the beginning. It will also be critical to: (1) reimplement Soar's *reflective layer* and learning mechanism(s); (2) implement and integrate in other cognitive capabilities, such as planning, emotion, social cognition and language; (3) reexamine the implementation of a broader range of architectures (such as ACT-R); and (4) forge a bridge to neural architectures. Success should yield both a uniform implementation level for architecture development – i.e., a narrow waist for the cognitive hourglass – and better architectures for cognitive modeling.

Acknowledgments

This effort was made possible by sabbatical support from the USC Viterbi School of Engineering plus funding from the Institute for Creative Technologies (ICT). ICT's Cognitive Architecture Working Group has been invaluable for semi-public exploration of these ideas. (Kschischang *et al*., 2001) first attracted my attention to factor graphs and is the basis for much of the general material here on them.

References

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1993). *Rules of the Mind.* Erlbaum.

Bonawitz., K. A. (2008) *Composable Probabilistic Inference with Blaise*. Doctoral Dissertation, Department of EECS, MIT, Cambridge, MA.

Boutilier, C., Friedman, N., Goldszmidt, M. & Koller, D. (1996). Context-specific independence in Bayesian networks. *Proceedings of the Twelfth Conference on Uncertainty in AI* (pp. 115-123). Morgan Kaufman.

Cho, B., Rosenbloom, P. S. & Dolan, C. P. (1991). Neuro-Soar: A neural-network architecture for goal-oriented behavior. *Proceedings of the 13th Annual Conference of the Cognitive Science Society* (pp. 673-677). Erlbaum.

Cooper, R. & Fox, J. (1998). COGENT: A visual design environment for cognitive modeling. *Behavior Research Methods, Instruments & Computers*, *30*, 553-564.

Dechter, R. & Mateescu, R. (2003). A simple insight into iterative belief propagation's success. P*roceedings of The Nineteenth Conference on Uncertainty in Artificial Intelligenc*e (pp. 175-183). Morgan Kaufman.

Dechter, R. & Pearl, J. (1987). Network-based heuristics for constraint-satisfaction problems. *Artificial Intelligence*, *34*, 1-38.

Deering, S. (1988). *Watching the waist of the protocol hourglass*, Keynote address at ICNP '98.

Domingos, P. (In press). What is missing in AI: The interface layer. In P. Cohen (Ed.), *Artificial Intelligence: The First Hundred Years*. Menlo Park, CA: AAAI Press.

Domingos, P., Kok, S., Poon, H., Richardson, M. & Singla, P. (2006). Unifying logical and statistical AI. *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (pp. 2-7). AAAI Press.

Drost, R. J. & Singer. A. W. (2003). Image segmentation using factor graphs. *Proceedings of the 2003 IEEE Workshop on Statistical Signal Processing* (pp. 150-153).

Forgy, C. L. (1982). Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence*, *19*, 17-37.

Gogate, V. & Dechter, R. (2005). Approximate Inference Algorithms for Hybrid Bayesian Networks with Discrete Constraints. *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence* (pp. 209-216).

Jilk, D. J., Lebiere, C., O'Reilly, R. C. & Anderson, J. R. (2008). SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, *20*, 197-218.

Jordan, M. I. (2004). Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, *19*, 140-155.

Jordan, M. I. & Sejnowski, T. J. (2001). *Graphical Models: Foundations of Neural Computation*. MIT Press.

Kschischang, F. R., Frey, B. J. & Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, *47*, 498-519.

Laird, J. E. (2008). Extending the Soar cognitive architecture. In *Artificial General Intelligence 2008: Proceedings of the First AGI Conference*. IOS Press.

Laird, J. E. & Rosenbloom, P. S. (1986). Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning, 1*, 11-46.

Laird, J. E. & Rosenbloom, P. S. (1996). The evolution of the Soar cognitive architecture. In D. M. Steier. and T. M. Mitchell (Eds.), *Mind Matters: A Tribute to Allen Newell*. Mahwah, NJ: Erlbaum.

McCallum, A., Rohanemanesh, K., Wick, M., Schultz, K. & Singh, S. (2008). FACTORIE: Efficient probabilistic programming via imperative declarations of structure, inference and learning. *Proceedings of the NIPS workshop on Probabilistic Programming*.

Mézard, M., Parisi, G. & Zecchina, R. (2002). Analytic and algorithmic solution of random satisfiability problems. *Science*, *297*, 812-815.

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L. & Kolobov, A. (2007). BLOG: Probabilistic models with unknown objects. In L. Getoor and B. Taskar, (Eds.) *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufman,

Pearson, D. J., Gorski, N. A., Lewis, R. L. & Laird, J. E. (2007). Storm: A framework for biologically-inspired cognitive architecture research. In *Proceedings of the 8th International Conference on Cognitive Modeling*.

Rosenbloom, P. S. (2006). A cognitive odyssey: From the power law of practice to a general learning mechanism and beyond. *Tutorials in Quantitative Methods for Psychology*, *2*, 43-51.

Rosenbloom, P. S. (2009). A graphical rethinking of the cognitive inner loop. In *Proceedings of the IJCAI International Workshop on Graph Structures for Knowledge Representation and Reasoning.*

Rosenbloom, P. S., Laird, J. E. & Newell, A. (1993). *The Soar Papers: Research on Integrated Intelligence*. Cambridge, MA: MIT Press.

Rosenbloom, P. S., Newell, A. & Laird, J. E. (1991). Towards the knowledge level in Soar: The role of the architecture in the use of knowledge. In K. VanLehn (Ed.), *Architectures for Intelligence*. Erlbaum.

Tambe, M. & Rosenbloom, P. S. (1994). Investigating production system representations for non-combinatorial match. *Artificial Intelligence*, *68*, 155-199.

Yedidia, J. S., Freeman, W. T. & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, *51*, 2282-2312.

Spiking Neurons and Central Executive Control: The Origin of the 50-millisecond Cognitive Cycle

Terrence C. Stewart (tcstewar@uwaterloo.ca)
Chris Eliasmith (celiasmith@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Abstract

A common feature of many cognitive architectures is a central executive control with a 50-millisecond cycle time. This system determines which action to perform next, based on the current context. We present the first model of this system using spiking neurons. Given the constraints of well-established neural time constants, a cycle time of 46.6 milliseconds emerges from our model. This assumes that the neurotransmitter used is GABA (with GABA-A receptors), the primary neurotransmitter for the basal ganglia, where this cognitive module is generally believed to be located.

Keywords: cognitive cycle time; central executive; LIF neurons; neural production system; neural engineering framework; cognitive architectures

Introduction

ACT-R, Soar, GOMS, EPIC, and a variety of other approaches to modelling human cognition all contain a common assumption about the central control of cognitive operations. This is usually regarded as a production system where IF-THEN rules are applied sequentially. This imposes a serial bottleneck where low-level decisions as to which cognitive action should be performed next are made, requiring approximately 50 milliseconds per decision (Anderson et al., 1995).

While this 50 millisecond cognitive cycle time leads to models that match empirical data, the neurological basis for this time constraint has not been previously established. This paper develops a model of low-level rule application using realistic spiking neurons. The 50 millisecond cycle time is then shown to be the result of well-established neuron membrane and neurotransmitter properties. The result is not only a realistic, neurally plausible model of a core component for cognition, but also an explanation for why this characteristic time appears across architectures.

The model presented here is not meant to be complete. In particular, we do not provide a model of the developmental process which leads to the decision making system. We also do not currently include any learning capabilities, although this is part of our ongoing research. Instead, our model uses fixed mathematically derived synaptic connection weights, in contrast to most neural network models. These derived weights are meant to be the final result of a learning process, and weights derived in this manner have been shown to be realistic and highly robust to noise and neuron death (Eliasmith & Anderson, 2003).

Recent results have suggested that the brain area that corresponds to the system we are modelling is the basal ganglia (e.g. Anderson et al., 2004). This provides us with constraints as to the neural properties and neurotransmitters involved. However, since we are not yet modelling all aspects of this system and its interactions with other brain areas, we do not present our work as a complete model of the basal ganglia.

We start by describing the basic components of our model: the standard leaky integrate-and-fire (LIF) neuron and a model of post-synaptic current caused by a neural spike. From these, we construct a simplistic minimal model of neural decision making. We then add a competition system so that only one option at a time is represented. Finally, we build a working memory system so that context can be stored over time.

Neural Model

The standard basic model of spiking neurons is the leaky integrate-and-fire (LIF) model. While computationally simple, it provides a good approximation to real neurons over a wide range of conditions (Koch, 1999). It uses a point neuron, as opposed to more complex compartment models where ion flows within the neuron are modelled at a sub-millisecond level. Current is constantly leaking out of the neuron as per the membrane resistance *R*, but if enough input current is gathered to cause the voltage to be above a certain threshold, then the neuron will fire. After firing, the voltage is set to 0 for a fixed refractory period (2 milliseconds) before starting to gather current again.

Given a constant current input *J* and membrane resistence *R*, the voltage level of the LIF neuron changes over time as given in Equation 1 and shown in Figure 1. The timing of this behaviour is controlled by τ_{RC}, the membrane time constant of the neuron. Larger values cause the neuron's voltage to change more slowly, making it slower to respond to changes in input current. Interestingly, many real neurons are well-characterized by LIF neurons with membrane time constants in the range of 20 milliseconds, so this value is used for all simulations reported here.

$$V(t) = J R(1 - e^{-t/\tau_{RC}}) \qquad (1)$$

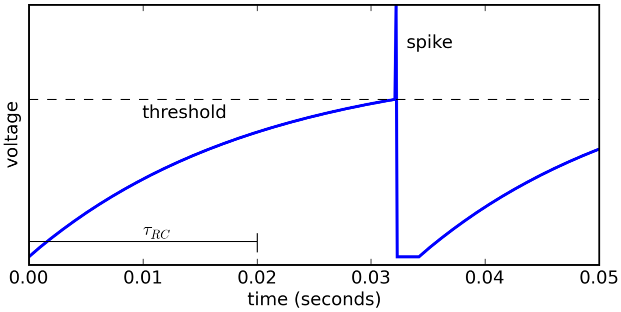


Figure 1: LIF neuron with constant input current.

When a neuron fires, it affects the input current to all of the neurons to which it is connected. This current *h(t)* can be characterized by Equation 2, where τ_s captures the effects of neurotransmitter re-uptake and dispersal. As shown in Figure 2, a small τ_s provides a fast, short-lasting effect (~10ms), while others last for hundreds of milliseconds.

$$h(t)=t e^{-t/\tau_s} \qquad (2)$$

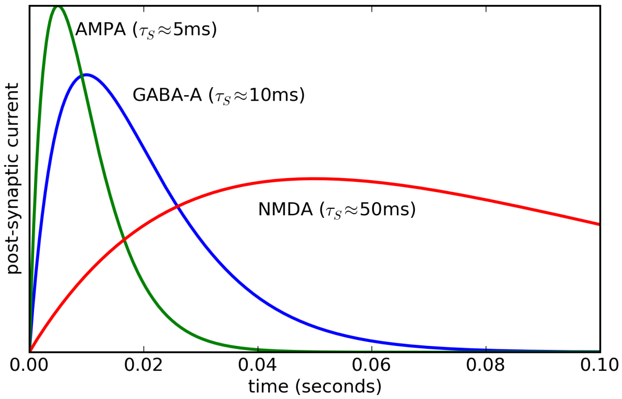


Figure 2: Post-synaptic currents for common synapses.

The τ_s values used here are approximate, based on available neurophysiology data. Gupta et al. (2000) estimate τ_s for GABA-A to be 10.41ms. AMPA is generally found to be between 1ms and 10ms and NMDA between 50ms and 150ms (e.g. Moreno-Bote & Parga, 2004).

While the neural model we are using is not a perfect replication of real neurons, we find it sufficient for our purposes. The LIF neurons allow us to explore the timing of neural processing, unlike the typical rate neurons used in most neural models. These not only do not spike, but also do not have any temporal dynamics at all, responding instantly to any change in input. Furthermore, given that the basic neural behaviour is well captured by the LIF model, switching to a more detailed model should not significantly impact the large-scale behaviour of the system (on the order of tens of milliseconds). That said, our model does not rely on the use of LIF neurons, and other more complex models could be used.

Neural Representation

Any model of a central executive control system where particular actions are chosen based on the current context must confront the issue of neural representation. The current context must be represented in such a manner as to appropriately affect the behaviour of other neurons. The approach described here to define and create such models is known as the Neural Engineering Framework (NEF; Eliasmith and Anderson, 2003).

To be as general as possible, we make the minimal assumption that representations can be distributed across a group of neurons, but leave open the question of the exact nature of this distribution. Within a neural group, each neuron has a preferred value $\tilde{\phi}$ to which it responds most strongly, and this response is reduced as the difference between this preferred value and the actual value increases.

If we assume that any value the neurons can represent can be thought of as a vector ***x***, this behaviour can be captured in terms of the input current *J* as shown in Equation 3. Adjusting the neuron gain α, the background input current J_{bias}, and the preferred direction vector $\tilde{\phi}$ allows us to capture a wide range of known neural representation schemes.

$$J=\alpha\tilde{\phi}\cdot\boldsymbol{x}+J_{bias} \qquad (3)$$

In the simplest case, 100 neurons could represent a 100 dimensional vector ***x*** by having each $\tilde{\phi}$ be a different unit vector in each of the 100 dimensions. This would provide a completely local representation of each value in the vector. More realistically, 100 neurons could represent one or two dimensions by having $\tilde{\phi}$ values chosen randomly (i.e. uniformly distributed around the unit hypersphere in that many dimensions). This approach has been observed in numerous areas of visual and motor cortex (e.g. Georgopoulos et al., 1986). By having more neurons per dimension, the representation error can be decreased to arbitrarily low levels (error is inversely proportional to the number of neurons).

Since Equation 3 can be used as the input to a model of an LIF neuron, we can determine the sequence of spikes that would be generated for a group of neurons if a particular vector ***x*** is being represented. We can also perform the opposite operation: given a sequence of spikes we can estimate the original vector. As shown elsewhere (Eliasmith & Anderson, 2003), this can be done by deriving the decoding vectors ϕ as per Equation 4, where a_i is the average firing rate for neuron *i* with a given vector ***x***, and the integration is over all values of ***x***.

$$\phi=\Gamma^{-1}\Upsilon \quad \Gamma_{ij}=\int a_i a_j dx \quad \Upsilon_j=\int a_i x dx \qquad (4)$$

The resulting vectors ϕ can be used to determine an estimate of the represented value using Equation 5. This is an estimate that varies over time based on the individual spikes. Importantly, it is the optimal estimate when under the constraint that the estimate must be built by linearly adding the effects of the post-synaptic currents caused by each spike. This is precisely the constraint that other neurons are under, so Equation 5 indicates the best that the original vector can be reconstructed by another neuron.

$$\hat{\boldsymbol{x}}(t)=\sum_{i,n}\delta(t-t_{i,n})*h_i(t)\phi_i=\sum_{i,n}h(t-t_{i,n})\phi_i \qquad (5)$$

This result further provides a method for determining optimal synaptic connection weights between groups of neurons if one group is to perform a linear transformation on the value represented by the other. If one group of neurons represents ***x*** and the other group should represent ***Mx***, then this can be achieved by setting the connection weights *w* as per Equation 6.

$$w_{ij}=\alpha_j\tilde{\phi}_j\boldsymbol{M}\phi_i \qquad (6)$$

We can also use a variant of Equation 4 to determine connection weights for arbitrary nonlinear transformations of ***x*** (see Eliasmith & Anderson, 2003 for details).

The Task

As a baseline for the construction and demonstration of our model, we use a simple minimal sequential decision making task. This is meant to show that the model is capable of responding appropriately to different contexts, and is capable of modifying the context itself.

The current context is represented by a large group of neurons (at least 2000 in all models shown here), as per the representation system described in the previous section. The preferred direction vectors $\tilde{\phi}$ are chosen randomly from the unit hypersphere, and the neuron gains α and background currents J_{bias} are chosen to give a uniform distribution of maximum firing rates between 100Hz and 200Hz and an average background firing rate of 40Hz, consistent with many cortical neurons. At the beginning of a simulation, this context is fixed to represent the initial state of the model, but after this initialization period (50ms) there is no external input. That is, the model must be capable of maintaining and changing its own internal state.

We arbitrarily choose five vectors to represent five different internal states referred to as *A*, *B*, *C*, *D*, and *E*. The model's task is to implement the set of state change rules that will cause it to cycle through these five states. If the system is in state *A*, it should change to state *B*; if it is in *B*, it should change to *C*, and so on.

In terms of the cognitive architecture ACT-R, this would involve five production rules. Each production rule would match on a particular goal buffer state (*A* through *E*), and if that production fires it would modify the goal buffer to contain the next state in the sequence. In ACT-R (and in most other cognitive architectures), this process is externally fixed to require 50 milliseconds. As will be seen, in our models this timing will emerge from neural properties.

Figure 3 shows an idealized (non-neural) model of this process. The five different colours indicate the five different representational states over a period of 500 milliseconds. This is enough time for the system to repeat the cycle twice. At each moment in time, we measure the represented vector ***x*** and compare it to the arbitrarily chosen patterns *A* through *E*. This comparison is done by taking the dot product of the represented value (from Equation 5) and each of the five target patterns.

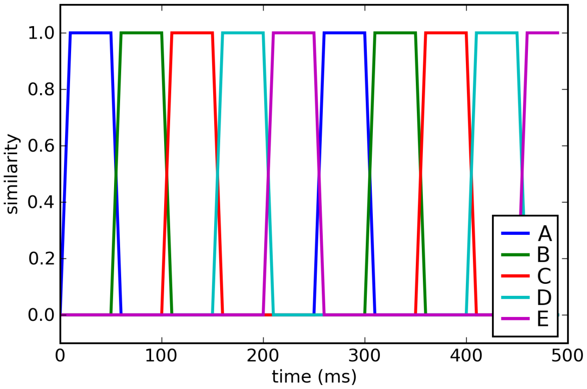


Figure 3: Behaviour of an ideal model cycling through five states, fixed to have a 50 millisecond cycle time.

Model 1: Basic Sequential Decisions

Our first model is created by adding a separate population of neurons for each of the rules to be implemented. These neurons must be connected to the main context neurons so that they will only fire when the value being represented is the same as (or very close to) the desired state (*A* through *E*). When a particular group of neurons starts to fire, their connections back to the context neurons are such that they will drive its firing towards the desired next state. This structure is shown in Figure 4. For clarity, this diagram shows only three rules: A→B, B→C, and C→A.

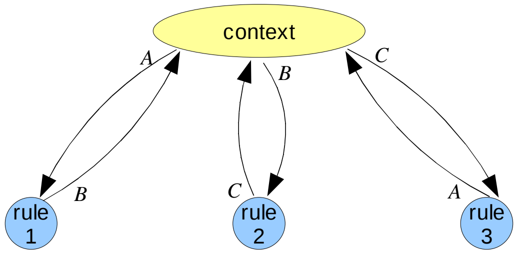


Figure 4: Neural groups and connections for Model 1.

To form the synaptic connections from the context to the rule neural groups, we can use Equation 6. For example, for the connection to the first rule, we set *M* to be the pattern *A*. As per Equation 6, this means that the neural group will be driven to represent the value *A****x***, which is the dot product of the represented context value with *A*. This will be large (near 1) when *A* is being represented, and small (near 0) when another pattern is being represented.

The properties of the neurons in the rule groups must also be set. Here, we can make use of the fact that we want these neurons to not fire at all when representing 0, but should be sensitive to values near 1. This can be achieved by having a large negative J_{bias} (with some random variation). The corresponding neuron tuning curves are shown in Figure 5. These show the average spiking rate of ten different neurons for different contexts ***x***. To see the actual spiking patterns over time, Figure 6 shows the spikes caused by varying the input to this neural group from 0 to 1 and back to 0 over one second.

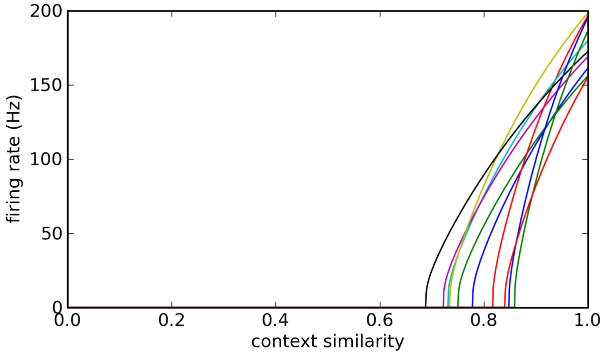


Figure 5: Average firing rates for neurons detecting the presence of pattern *A*. Different context patterns are on the x-axis: far left is a context unlike *A* (dot product of 0), far right is a context of exactly *A*. Each curve shows a different neuron with different values of α and J_{bias}.

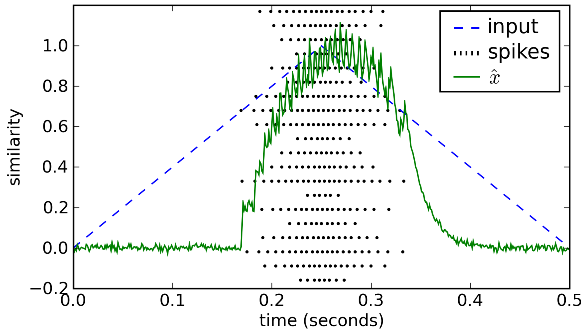


Figure 6: Individual neuron spikes for the neural group detecting *A*. Each neuron's spikes are separated along the y-axis. Dot product of the context with *A* is varied from 0 to 1 and back to 0 over one second (dotted line). The value $\hat{x}$ decoded using Equation 5 is shown (solid line).

We use a similar process to form synaptic connections from the individual rule groups back to the context neurons. Here, the weights encode the effect of each rule, indicating how the context should be changed if this rule is applied. These are again calculated using Equation 6. The resulting model has a variety of parameters, given in Table 1.

Table 1: Parameters of the model

Parameter	Default value
# of context neurons	2000
# of neurons per rule group	20
membrane time constant (τ_{RC})	20ms
synaptic time constant for context (τ_{SC})	10ms
synaptic time constant for rules (τ_{SR})	10ms

The behaviour of the resulting model is shown in Figure 7. As can be seen, it successfully cycles between the five states. For this particular model, each change requires an average of 27.5ms, making this much faster than the expected 50 millisecond cycle time. Furthermore, this rate is not sensitive to the numbers of neurons in each group: increasing these values by a factor of 10 causes only a slight decrease (<2ms) in the cycle time, since adding more neurons decreases the representational error in the system.

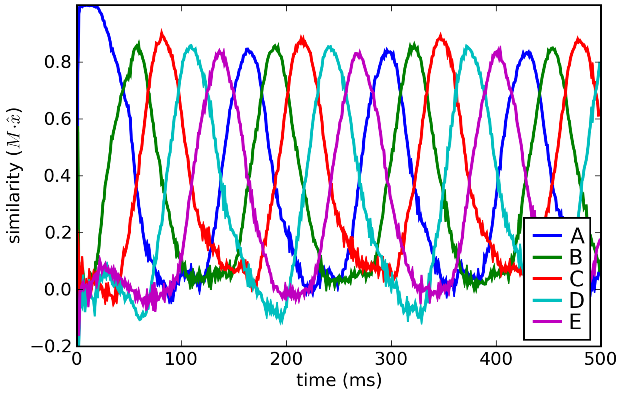


Figure 7: Behaviour of Model 1. Similarity is determined by the dot product of $\hat{x}$ (calculated from the spikes of the context neurons using Equation 5) with the vectors *A* to *E*.

The main effect on behaviour is seen by adjusting the synaptic time constants. As shown in Figure 2, different neurotransmitter/receptor pairs have different time constants. We can adjust the synapses from context neurons to rules separately from the ones from rule neurons to the context. These parameters are varied in Figure 8. The membrane time constant is known to be approximately 20ms for a wide range of neurons, so it is not adjusted here.

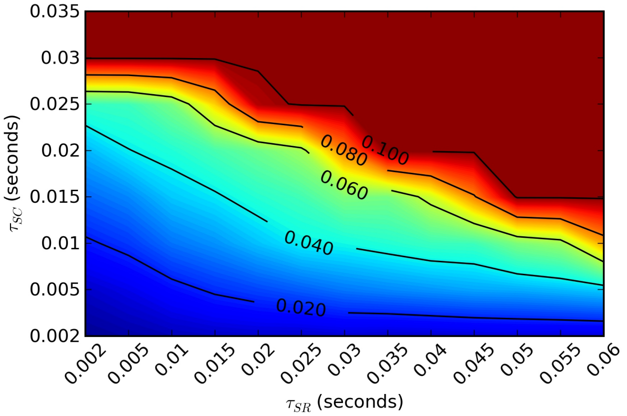


Figure 8: Average cycle time in seconds for varying τ_{SC} and τ_{SR} in Model 1. Values above 0.1 indicate either a cycle time above 100ms or no cycling (an infinite cycle time).

Given the results in Figure 8, the model is successful when the synaptic time constant for the context neurons is below 30ms, which is consistent with both GABA-A and fast AMPA synapses. This limit decreases as the synaptic time constant of the rule neurons increases.

While this model is successful at cycling across five different states, it fails in many other cases. For example, Figure 9 shows the behaviour when cycling between three states. Here, cycling behaviour is initially evident, but over time the system converges to a static representation. In particular, it converges to representing *all three* states at the same time. The final context value is the superposition (vector sum) of *A*, *B*, and *C*. This is clearly not the desired behaviour.

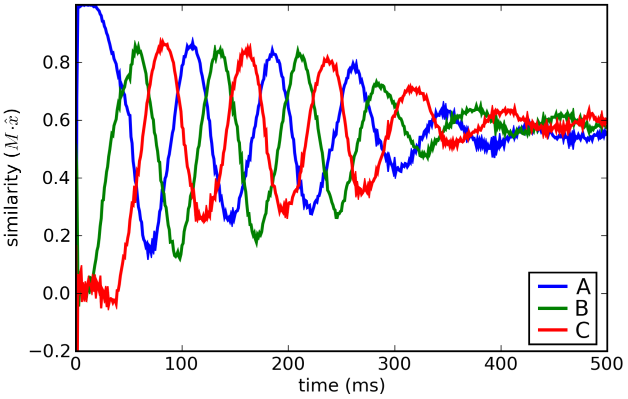


Figure 9: Behaviour of Model 1 when there are only three states. Similarity is determined by the dot product of $\hat{x}$ (calculated from the spikes of the context neurons using Equation 5) with the vectors *A*, *B*, and *C*.

Model 2: Inhibition Between Rules

To improve on Model 1 and fix the behaviour shown in Figure 9, we needed to add a mechanism to encourage the application of only one rule at a time. This was accomplished by adding inhibition between the groups of neurons responsible for each rule. That is, if the neurons in the first group are firing, this should decrease the activity in the other four groups. This is accomplished with Equation 6, where *M* is simply the value -w_i (the strength of the inhibition). We must also add a self-excitatory connection of strength w_e within the neurons of each rule group, so as to counteract this inhibitory current. This new model is shown in Figure 10.

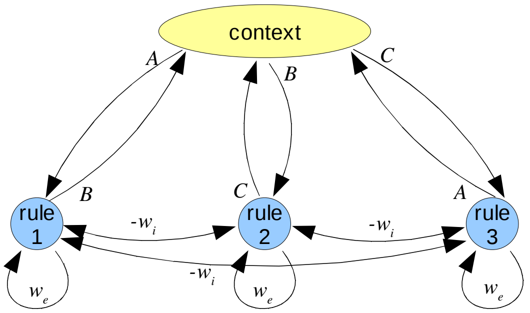


Figure 10: Neural groups and connections for Model 2.

For w_i of 0.5 and a w_e of 1, the model is successfully cyclic for cycles of 2 through 20 (which was as high as was tested). That is, the resulting behaviour looks like Figure 7, rather than Figure 9. The precise effects of these parameters will be explored in future work, as they are likely to impact any reinforcement learning system which might be used to bias one rule over another (such as in the ACT-R utility learning system). With these parameter values, the behaviour of the model for varying τ_{SC} and τ_{SR} is shown in Figure 11. We can see that Model 2 is slightly slower, but more stable over a wider range of synaptic time constants.

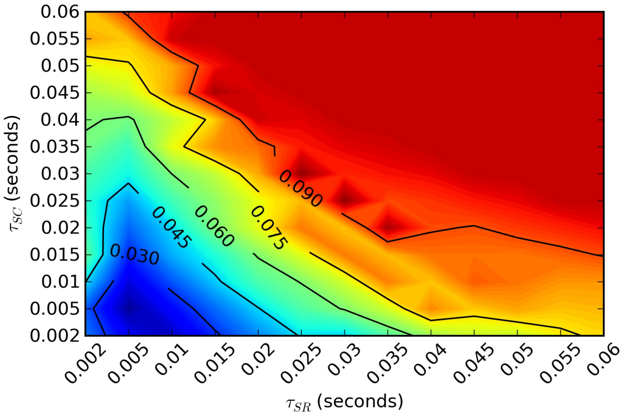


Figure 11: Average cycle time in seconds for varying τ_{SC} and τ_{SR} in Model 2. Values above 0.1 indicate either a cycle time above 100ms or no cycling (an infinite cycle time).

While this model eliminates the problem of convergence onto a superposition of states, there is a further difficulty present in both Model 1 and Model 2. So far, we have been assuming that this rule-following system is completely self-sufficient. In particular, once an action is chosen, the context is modified, and the system is then immediately able to start identifying the next rule to apply.

However, in real cognitive models, the central production system is only one of many components that can affect the current context. For example, in ACT-R, it is common for the production system to request that the declarative memory system recall a fact. While that fact is being recalled, the production system may not be doing anything, as no rules may apply until that fact is found (which may take hundreds of milliseconds). During that time, no rules are applied, but the context must be maintained.

Figure 12 shows the behaviour of Model 2 when no rules can be found that apply to the current context. This is done by removing the rule that transitions from *E* to *A*. As can be seen, when no rule can be applied, the system *forgets* the current context, since no rule is firing to set it in the context population. Model 1 behaves similarly.

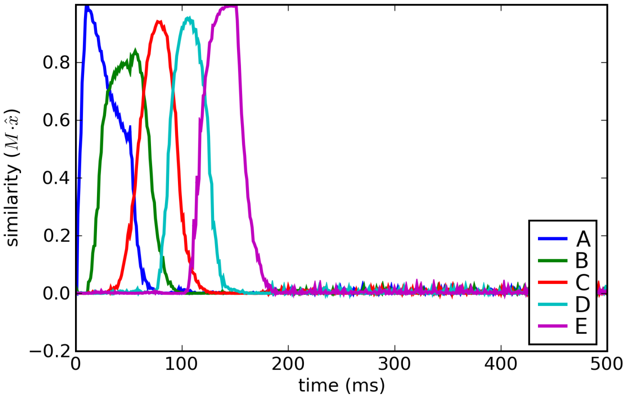


Figure 12: Behaviour of Model 2 when the rule to go from *E* to *A* is removed. The context information is lost.

Model 3: Maintaining Working Memory

To eliminate the forgetting effect shown in Figure 12, we add recurrent connections among the neurons representing context. This approach has previously been used to model working memory (Singh & Eliasmith, 2006), and is a generic method for storing information over time in spiking neurons. This is done by using Equation 6 to determine synaptic weights from the context population back into itself, with *M* set to be the identity matrix *I*. The resulting model is shown in Figure 13.

The behaviour of this model when the rule to transition from *E* to *A* is removed is shown in Figure 14. In contrast to Model 2 (Figure 12), the system is now capable of maintaining context information over time.

Adding this new recurrent connection allows information to be stored, but it also slows down the process of modifying this information. The behaviour for varying τ_{SC} and τ_{SR} is shown in Figure 15.

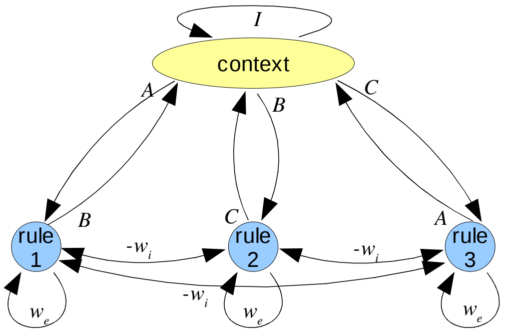


Figure 13: Neural groups and connections for Model 3.

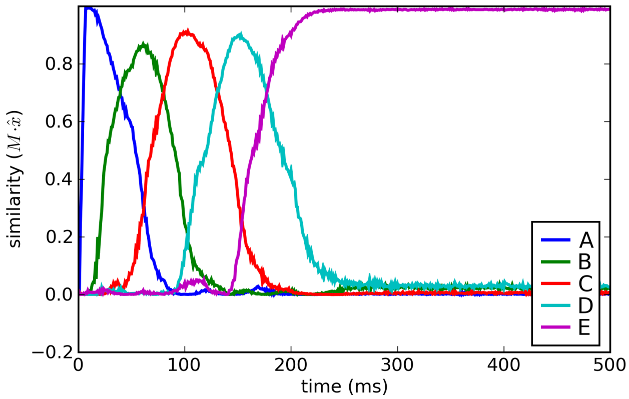


Figure 14: Behaviour of Model 3 when the rule to go from *E* to *A* is removed. The context information is maintained.

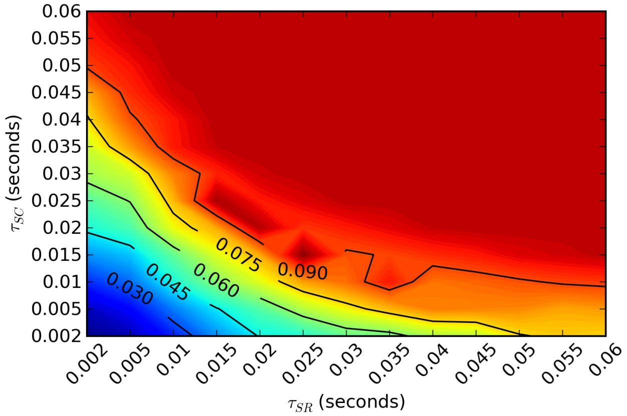


Figure 15: Average cycle time in seconds for varying τ_{SC} and τ_{SR} in Model 3. Values above 0.1 indicate either a cycle time above 100ms or no cycling (an infinite cycle time).

Discussion

Our Model 3 successfully identifies the rule appropriate to the current context and modifies the context appropriately. It is able to keep the patterns for each context separate (unlike Model 1) and store information over time (unlike Model 2). Furthermore, if the synaptic time constants for both the context neurons and the rule neurons are set to be 10ms, the average cycle time is 46.6ms, very close to the standard of 50ms. As noted above, 10ms is the synaptic time constant for GABA-A receptors. These are the primary synaptic receptors in the basal ganglia, which is the postulated location responsible for sequential rule selection.

While our model closely matches the generally accepted cycle time of 50 milliseconds, more is needed before it can be accepted as a neural model of central executive control. Most crucially, cognitive architectures generally postulate rules that are much more complex than “if A then B”. We have shown elsewhere (Stewart & Eliasmith, 2008) how complex symbolic rules can be translated into vectors appropriate for our model. This would require the addition of a new neural population capable of combining the output of the rule neurons with the existing context. Preliminary results indicate that such a system would increase the cycle time by 5-10ms if AMPA or GABA-A are used.

We are also in the process of directly mapping our model onto the architecture of the basal ganglia and its connection to the cortex via the thalamus. In this case, the context may be stored using faster AMPA connections in various cortical areas and then gathered in the striatum for matching. The thalamus would then apply the complex rules mentioned in the previous paragraph. This is a direct match to the mapping from modules to brain areas found in ACT-R (Anderson et al., 2004). Furthermore, a learning system is required (likely using a dopamine-based expected reward signal) to identify how these synaptic connections arise.

Although our model is incomplete, it provides the first neural explanation for the 50 millisecond cognitive cycle. This time is a direct result of the properties of GABA-A receptors, along with the requirements that the system be able to recognize appropriate rules in a given context, apply rules separately, and store context information over time.

References

Anderson, J. R., John, B. E., Just, M. A., Carpenter, P. A., Kieras, D. E., & Meyer, D. E. (1995). *Production system models of complex cognition*. 17th Annual Meeting of the Cognitive Science Society.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review 111*(4). 1036-1060.

Eliasmith, C. & Anderson, C. (2003). *Neural Engineering: Computation, representation, and dynamics in neurobiological systems.* Cambridge: MIT Press.

Georgopoulos, A.P., Schwartz, A., & Kettner, R.E. (1986). Neuronal population coding of movement direction. *Science*, 233, 1416-1419.

Gupta, A., Wang, Y., & Markram, H. (2000). Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex. *Science*, 287. 273-278.

Koch, C. (1999). *Biophysics of computation: Information processing in single neurons.* Oxford University Press.

Moreno-Bote, R., & Parga, N. (2005). Simple model neurons with AMPA and NMDA filters: role of synaptic time scales. *Neurocomputing* 65-66. 441-448.

Singh, R., & Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *Journal of Neuroscience*, 26, 3667-3678.

Stewart, T.C. and Eliasmith, C. (2008). *Building production systems with realistic spiking neurons.* 30th Annual Meeting of the Cognitive Science Society.

A Biologically Realistic Cleanup Memory: Autoassociation in Spiking Neurons

Terrence C. Stewart (tcstewar@uwaterloo.ca)
Yichuan Tang (y3tang@uwaterloo.ca)
Chris Eliasmith (celiasmith@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo
Waterloo, ON, N2L 3G1

Abstract

Methods for cleaning up (or recognizing) states of a neural network are crucial for the functioning of many neural cognitive models. For example, Vector Symbolic Architectures provide a method for manipulating symbols using a fixed-length vector representation. To recognize the result of these manipulations, a method for cleaning up the resulting noisy representation is needed, as this noise increases with the number of symbols being combined. While these manipulations have previously been modelled with biologically plausible neurons, this paper presents the first spiking neuron model of the cleanup process. We demonstrate that it approaches ideal performance and that the neural requirements scale linearly with the number of distinct symbols in the system. While this result is relevant for any biological model requiring cleanup, it is crucial for VSAs, as it completes the set of neural mechanisms needed to provide a full neural implementation of symbolic reasoning.

Keywords: Autoassociative memory; Neural engineering framework; Vector symbolic architectures; Holographic reduced representation;

Autoassociative Memory

A fundamental component of many cognitive architectures is an autoassociative memory. This is a system that can be provided with a partial or noisy version of a previously stored memory and will in turn provide a complete and more accurate version of that memory. This can be seen in ACT-R's declarative memory system (Anderson & Lebiere, 1998), CLARION's non-action-centered subsystem (Sun, 2006), RAAM's compressor and reconstructor (Pollack, 1988), and many other cognitive models. This capability can be implemented using a wide variety of approaches, including multilayer perceptrons, Hopfield networks, and any prototype-based classifier.

The particular use of autoassociative memory of importance to this paper is as a *cleanup memory* for cognitive operations. Recent research has shown that the storage and manipulation of cognitive symbol systems can be implemented as mathematical operations on fixed-length, high-dimensional vectors. These approaches are known as Vector Symbolic Architectures (VSAs; Gayler, 2003) and include Holographic Reduced Representation (HRR; Plate, 2003), MAP Coding (Gayler & Wales, 2000), Binary Splatter Codes (Kanerva, 1997), and others. Each of these provides an alternate method for converting symbols and symbol trees into vectors, combining vectors to perform symbolic manipulations, and extracting out the original components of that symbol tree.

In previous research we have shown how VSAs can be implemented in biologically realistic spiking neurons (Eliasmith, 2005; Stewart & Eliasmith, 2008). This approach is many orders of magnitude more efficient[1] than alternate theories of how symbolic manipulations could be performed by the brain (Stewart & Eliasmith, in press). However, one common criticism is that this approach does not yet show how these systems can clean up their representations. Performing symbol manipulations using VSAs is an inherently noisy process, and these operations must be performed by spiking neurons, adding a significant amount of random variation. When symbols are extracted from a bound representation, the brain needs a reliable method for identifying which symbol it is, allowing it to respond appropriately.

The purpose of this paper is to present an autoassociative memory constructed from spiking neurons which is appropriate for cleaning up the representations resulting from cognitive manipulations using VSAs. We first describe the characteristics of VSAs that define the statistical properties of the noise that must be removed. Next, a general method is described for encoding (and decoding) high-dimensional vectors across a population of spiking neurons. We then show that standard approaches to deriving connection weights have difficulty when scaled up to the number of symbols required for human. Our cleanup memory model is then presented, followed by an analysis of its behaviour.

Vector Symbolic Architectures

There are three core ideas for all VSAs. First, each symbol is represented by a randomly chosen vector. Second, two vectors can be combined by superposition ($+$) to produce a new vector that is *similar to* both of the original vectors. Third, two vectors can be combined by binding ($\otimes$) to produce a new vector that is *dissimilar* to both of the original vectors, and this operation can be reversed by binding with the inverse of a vector (denoted with an underline), such that $\mathbf{A}\otimes\mathbf{B}\otimes\underline{\mathbf{B}}\approx\mathbf{A}$. These operations are similar to standard addition and multiplication in terms of being associative, commutative, and distributive. With such a system we can represent a structure such as `chase(dog,cat)` by performing the following calculation:

[1] For realistic vocabulary sizes, this approach uses three orders of magnitude fewer neurons than the Neural Blackboard Architecture (van der Velde & de Kamps, 2006) and seven orders of magnitude fewer than LISA (Hummel & Holyoak, 2003).

$$\mathbf{chase}\otimes\mathbf{verb} + \mathbf{dog}\otimes\mathbf{subj} + \mathbf{cat}\otimes\mathbf{obj}$$

The result is a single vector of the same length as the vectors for the basic symbols (**chase**, **verb**, **dog**, etc.). This one vector can be interpreted as a representation of the entire structure because it is possible to extract the original components. For example, to determine the object of the above structure, we take the whole vector and bind it with the inverse of **obj**.

$$\begin{aligned}&(\mathbf{chase}\otimes\mathbf{verb} + \mathbf{dog}\otimes\mathbf{subj} + \mathbf{cat}\otimes\mathbf{obj})\otimes\underline{\mathbf{obj}}\\ &= \mathbf{chase}\otimes\mathbf{verb}\otimes\underline{\mathbf{obj}} + \mathbf{dog}\otimes\mathbf{subj}\otimes\underline{\mathbf{obj}} + \mathbf{cat}\otimes\mathbf{obj}\otimes\underline{\mathbf{obj}}\\ &\approx \mathbf{cat} + \mathbf{chase}\otimes\mathbf{verb}\otimes\underline{\mathbf{obj}} + \mathbf{dog}\otimes\mathbf{subj}\otimes\underline{\mathbf{obj}}\end{aligned}$$

The result is a vector that is similar to **cat**, but is not exactly the same since it has two additional terms superposed on it. Due to the properties of the binding operation, however, these two terms $\mathbf{chase}\otimes\mathbf{verb}\otimes\underline{\mathbf{obj}}$ and $\mathbf{dog}\otimes\mathbf{subj}\otimes\underline{\mathbf{obj}}$ will be vectors unlike any of the original symbols. They can thus be treated as randomly distributed noise. It is this noise that must be removed by the cleanup memory system.

While the above discussion applies to all VSAs, if we choose one particular type of VSA we identify the properties of the symbol and noise vectors. For this, we use Holographic Reduced Representations (HRRs; Plate, 2003). Here, each basic symbol vector is set by randomly selecting a point on the high-dimensional unit sphere (i.e. a random vector normalized to a length of one). Superposition is performed by vector addition and the binding operation is circular convolution.

The cleanup memory thus needs the following properties:

1) Recognize *M* unit vectors (one per symbol), distributed uniformly over a high-dimensional unit sphere.

2) Handle additive noise produced by adding *k* unit vectors uniformly distributed over the same sphere.

To be useful for cognitive operations, on the order of 100,000 symbols (*M*) must be able to be identified. The complexity of the structures that can be encoded is determined by *k*, indicating the number of terms that can be superposed and still lead to accurate recognition. This should be at least 7 ± 2 to conform to the standard chunk sizes used in cognitive modelling.

To determine whether recognition is accurate, we take the dot product of the correct vector and the output of the memory; if this value is above a threshold the symbol is successfully recognized. For the purposes of this paper, we arbitrarily choose a threshold of 0.7, although Plate (2003, p. 100) provides a method for determining the optimal threshold in special cases where *k* is fixed and known.

The final factor to consider when using Vector Symbolic Architectures is the number of dimensions. In an ideal case (where vectors are represented exactly, rather then via noisy spiking neurons), Plate (2003) derived the following formula for determining the minimum number of dimensions *D* required to represent combinations of k vectors out of *M* symbols and have a probability of error *q*:

$$D = 4.5(k + 0.7)\ln(M/30\mathrm{q}^4) \qquad (1)$$

From this, we note that 700 dimensions would be sufficient to represent chunks of up to 7 symbols out of a vocabulary of 100,000 with an accuracy of 95%. However, this formula assumes an ideal cleanup memory.

Distributed Representation

There are a variety of methods whereby a numerical vector can be represented by a population of spiking neurons. The most simplistic approach is to have one neuron per dimension, and the firing rate of that neuron indicates the value in that dimension. However, this approach is highly fragile to neuron death and does not correspond to known methods of spatial representation by neurons. It is well established (e.g. Georgopoulos et al., 1986) that movement directions are encoded by having a large population of neurons, each of which is sensitive to a different direction. The firing rate of each neuron is related to the angle between that neuron's preferred direction vector and the value being encoded.[2]

We take this same approach to encode high-dimensional vectors. Each neuron has a preferred direction vector $\tilde{\phi}$ and the current entering the neuron is proportional to the dot product between this and the vector ***x*** being represented. If α is the neuron gain and J^{bias} is a fixed background current, then the total current flowing into cell *i* is:

$$J_i = \alpha_i \tilde{\phi}_i \cdot \boldsymbol{x} + J_i^{bias} \qquad (2)$$

This current can be used as the input for any model of spiking neurons, such as the standard leaky integrate-and-fire (LIF) model. In general, ***x*** can vary over time as ***x(t)*** and the spikes produced will be based on this varying current. If the details of the neural model (i.e. the relation between input current and spiking behaviour) are written as *G[·]* and the neural noise of variance σ^2 is $\eta(\sigma)$, then the encoding of any given ***x(t)*** as the temporal spike pattern across the neural group is given as:

$$\sum_n \delta(t - t_{in}) = G_i[\alpha_i \tilde{\phi} \cdot \boldsymbol{x}(t) + J_i^{bias} + \eta_i(\sigma)] \qquad (3)$$

Since this spiking pattern is meant to represent the original vector ***x***, it should be possible to determine an estimate $\hat{\boldsymbol{x}}(t)$ given only this spiking pattern. This can be done by deriving linearly optimal (in terms of minimizing squared error) decoding vectors ϕ for each neuron as per Equation 4, where a_i is the average firing rate for neuron *i* (see

[2] It should be noted that the simplistic representation mentioned initially is a special case of this approach, where the preferred direction vectors are exactly aligned along the dimensions being represented, rather than being randomly distributed.

Eliasmith & Anderson, 2003 for details). This method has been shown to uniquely combine accuracy and neurobiological plausibility (e.g. Salinas and Abbot, 1994).

$$\phi = \Gamma^{-1} Y \quad \Gamma_{ij} = \int a_i a_j \, dx \quad Y_j = \int a_i x \, dx \qquad (4)$$

To derive an estimate of ***x(t)***, we weight the decoding vectors by the post-synaptic current *h(t)* induced by each spike. The shape and time-constant of this current are determined from the physiological properties of the neural group. The result is the best possible linear estimate of ***x(t)*** using only the spike timing information.

$$\hat{\mathbf{x}}(t) = \sum_{in} \delta(t - t_{in}) * h_i(t) \phi_i = \sum_{in} h(t - t_{in}) \phi_i \qquad (5)$$

The representational error between ***x(t)*** and $\hat{x}(t)$ is dependent on the particular neural parameters and encoding vectors, but in general is inversely proportional to the number of neurons in the group.

While the decoding vectors ϕ are useful for determining what a spike pattern represents, a more important feature is that they can also be used to derive optimal connection weights between neural groups. That is, consider a situation where one neural population represents ***x*** and we want a second neural population to represent ***Wx*** (where ***W*** is an arbitrary linear transformation). The optimal connection weights ω_{ij} between each neuron to achieve this are determined by Equation 6 (see Eliasmith & Anderson, 2003 for further details).

$$\omega_{ij} = \alpha_j \tilde{\phi}_j \mathbf{W} \phi_i \qquad (6)$$

These results provide a generic framework for representing vectors of any dimension using spiking neurons. These neurons can be made as realistic as possible (given computational processing constraints), including effects of adaptation, neurotransmitter re-uptake rates, refractory periods, and so on. Furthermore, we can derive the synaptic connection weights that will cause the neurons to perform the desired transformations on these represented values.

Standard Approaches

Given the above representation system, we have two groups of neurons: one representing the input (noisy) vector, and one representing the output (cleaned) vector. The goal then is to determine how to connect these neurons so as to achieve the best cleanup.

For this work, we are only considering feed-forward networks. That is, we do not consider models where activity flows backwards from the output to the input, or where the output is the same group of neurons as the input, but at a later time. These models, such as the Hopfield network, must wait for their output to "settle", requiring significantly more time than purely feed-forward models.

Linear Autoassociation

The simplest autoassociation memory merely performs a linear transformation on the input to produce the output (Hinton & Anderson, 1989). If the matrix ***X*** consists of a set of noisy vectors and the matrix ***Y*** holds the corresponding cleaned vectors, then we want to find ***W*** such that ***WX≈Y***. Given the subsequent noisy vector ***x***, it can then be multiplied by ***W*** to produce the estimated cleaned up item ***y=Wx***. Once ***W*** is found, we derive the connection weights for this linear transformation using Equation 6.

A variety of methods exist to find the ***W*** that minimizes the error between ***WX*** and ***Y***. Figure 1 shows the result of using the Penrose-Moore pseudoinverse, which was chosen since ***X*** is generally not full rank.

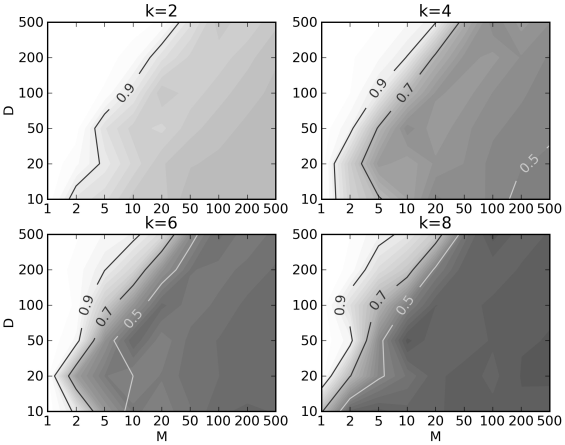


Figure 1: Accuracy of the linear autoassociation network for varying *D*, *M*, and *k*. Values above 0.7 (shown in lightest shading) indicate successful cleanup (i.e. output values sufficiently close to the original non-noisy vector).

These results show that the linear association approach does not scale up for large values of *M*. In 500 dimensions this network is unable to accurately clean up a vector where 4 symbols are combined if there are more than 50 possible symbols. This is much smaller than the desired 100,000.

Linear Neural Transformation

A second possibility is to directly determine the optimal connection weights, rather than relying on Equation 6. Here, instead of ***X*** being the noisy vectors, it is the spiking rate of the individual neurons when representing those vectors. This approach is used extensively in the Neural Engineering Framework (Eliasmith and Anderson, 2003) to derive synaptic connection weights that can perform nonlinear operations, using a slight modification of Equation 4 where ***x*** is replaced by the corresponding cleaned up vector. This allows synaptic connection weights to be derived that approximate arbitrary nonlinear functions.

While the results in Figure 2 show that this approach is a significant improvement over Figure 1 in terms of handling larger values of *k* at smaller *D*, it is still not scaling up for larger values of *M*.

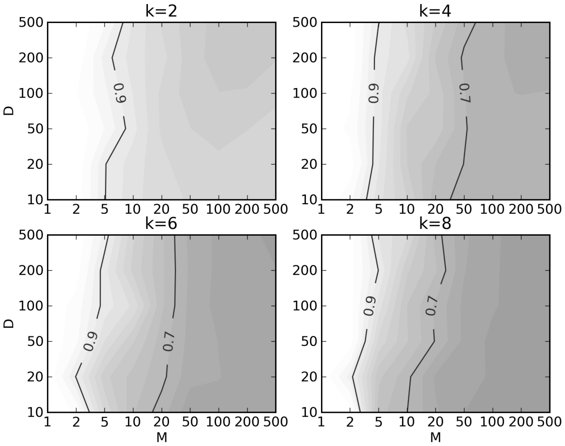


Figure 2: Accuracy of the linear autoassociation approach applied to individual neuron firing rates for varying *D*, *M*, and *k*. Values above 0.7 indicate successful cleanup.

Multilayer Perceptron

One potential reason for the failure of the linear associator discussed in the previous section is that the function being computed is highly nonlinear. To address this, we can make use of a multilayer perceptron, capable of computing much more complex functions. This involves introducing a new hidden layer of neurons between the input and output.

The multilayer perceptron is the most famous and widely used artificial neural network (Rumelhart et al., 1986). Using a two layer MLP, a mapping is learned to convert noisy input vectors into their cleaned (or prototype) vectors.

Instead of directly calculating the weights for these networks, a learning rule (such as the classic backpropagation of error rule) must be used. This allows the system to find a suitable intermediate representation in the hidden layer which makes the cleanup operation most accurate. For this task we trained the MLP using gradient descent on the sum of the squared error.

In theory, given enough time, hidden nodes, and a sufficiently powerful optimization algorithm, this approach should be able to find the optimal synaptic connection weights to perform this task. However, as the results in Figure 3 show, due to limited computational resources we were unable to successfully train this network for large M. This is in part due to the fact that the MLP requires many more hidden nodes than the vector dimension in order to generalize across the entire input domain.

More importantly, the standard strengths of a backpropagation network are not applicable to the cleanup task. Crucially, there is no inner structure in the data being modelled; each symbol is a randomly chosen unit vector. This means that the network cannot use its hidden layer to form an internal representation that simplifies the task.

Overall, it is likely possible to improve on this approach to training a network to perform cleanup. However, such a method may require significantly larger amounts of computing resources as *M* increases.

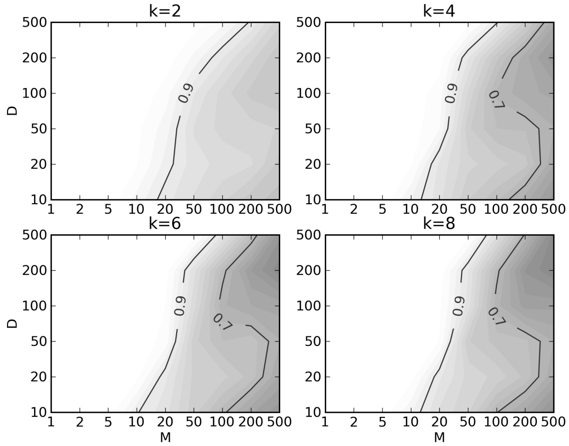


Figure 3: Accuracy of the multilayer perceptron for varying *D*, *M* and *k*. Values above 0.7 indicate successful cleanup.

A Cleanup Memory Model

From the MLP model, it is clear that while transforming the initial representation through a middle layer of neurons can provide a significant improvement, it is impractical to learn the required synaptic connection weights. Instead, for our cleanup memory model we choose to directly derive the optimal weights. To do this, we first identify how we want the middle layer of neurons to respond. This involves defining their preferred direction vectors $\tilde{\phi}$, gain α, and J^{bias} as per Equation 2. Given these, we can use Equation 6 to derive the neural connection weights that will result in this behaviour. Since no transformation of the vector itself is to be performed by the weights, ***W*** in Equation 6 is set to be the identity matrix.

For the preferred direction vectors, we choose exactly those vectors that must be cleaned up. For redundancy, we have ~10 neurons for each of the *M* vectors, meaning that there are particular neurons that fire maximally for each symbol. Furthermore, we set J^{bias} to be slightly negative for each neuron. The resulting connection weights ω_{ij} cause the middle layer neurons to only fire if the dot product of the input vector with the corresponding clean vector is greater than some small threshold (0.2).

In effect, the inherent non-linearity of the neurons (the fact that they do not fire if their input current is too low) is being used to perform cleanup. This middle layer is good at representing the cleaned vectors, but is poor at representing small vectors in any of those directions. Since the noise added to the input consists of randomly chosen vectors, these will generally have small dot products with each of the preferred direction vectors, and so will not cause sufficient activation for the neuron to fire. The presence of a slight background inhibition (the negative J^{bias}) allows the neurons to be insensitive to the noise.

The firing rates of ten sample middle layer neurons are shown in Figure 4. Their activity varies as the dot product of the input and the neurons' preferred direction vector changes.

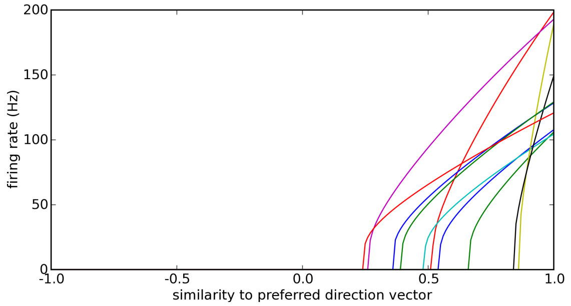


Figure 4: Middle layer neuron tuning curves. Average firing rates for ten neurons are shown as the input to the cleanup memory changes. Similarity is the dot product of the input vector with the preferred direction vector.

Given this middle layer representation we can then calculate the optimal connection weights with the output neural group. This output group can have any arbitrarily chosen preferred direction vectors $\tilde{\phi}$ and other neural properties. Equation 6 is used to calculate these weights, again setting ***W*** to be the identity matrix.

Performance

We evaluated this implementation of cleanup memory in the same manner as the previous models and the results are shown in Figure 5. It should be noted that these graphs extend to much larger *M* (10,000 symbols rather than 500) than the previous figures.

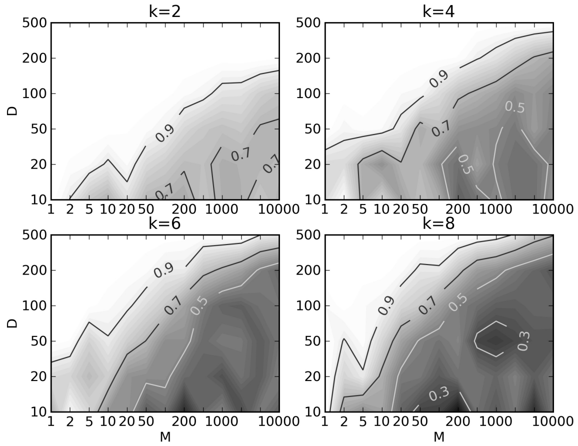


Figure 5: Accuracy of our neural cleanup memory for varying *D*, *M* and *k*. Values above 0.7 indicate successful cleanup.

Importantly, our neural cleanup memory system was able to successfully cleanup combinations of 8 symbols out of a vocabulary of 10,000 using 500 dimensional vectors. Furthermore, its capabilities increase rapidly with the number of dimensions. We have evaluated this model up to *M*=100,000 and *D*=1000, producing consistently high quality cleanup results.

We have thus demonstrated an effective implementation of a neural autoassociator as a cleanup memory for Vector Symbolic Architectures. The number of neurons required for cleanup scales linearly with *M*, while the number of neurons required for storing the resulting cleaned vector is linear in *D*.

Comparison to the Ideal

To determine how closely our model approaches ideal behaviour (even though it is implemented using realistically noisy spiking neurons), we can examine the recognition behaviour of a perfect mathematical cleanup system. This is used by Plate (2003) in his analysis of the Holographic Reduced Representation form of VSA, and merely outputs the clean vector that is closest to the input noisy vector. This ideal system can be approximated by Equation 1, and its actual behaviour is shown in Figure 6.

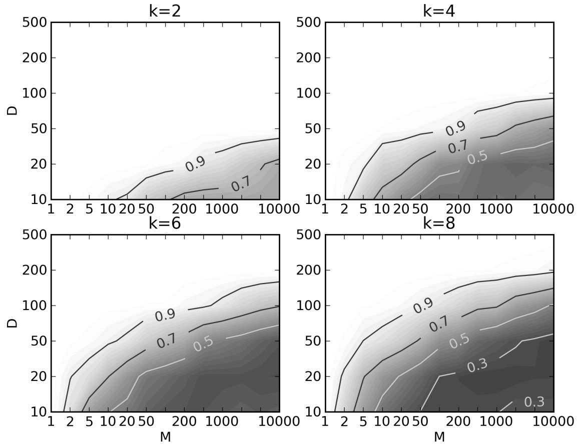


Figure 6: Accuracy of an ideal cleanup memory for varying *D*, *M* and *k*. Values above 0.7 indicate successful cleanup.

From this result, we see that our neural cleanup memory and the ideal cleanup both exhibit a similar growth in representational capacity as the dimensionality of the vectors increases. While the neural version is less accurate, it still is able to scale up to large *M*. This ability is not seen in the cleanup models examined previously.

Dynamics and Timing

Since a cleanup memory is meant to be a component to support symbolic manipulations by spiking neurons, it must not only be efficient in terms of numbers of neurons, but also in terms of the amount of time required to perform clean up. This is why we did not consider models that require a long settling time (such as a Hopfield network).

Since the dynamics of the neurons in our model (*G* in Equation 3) can be adjusted to match those of real neurons, we can generate predictions as to how the output of the cleanup memory will vary over time. Even with a constant input vector ***x***, the actual value being represented by the output of the cleanup memory will vary since it is decoded from the spike train as per Equation 5.

The precise timing characteristics of the neural model will vary based on the neural parameters. We used typical values for cortical neurons: a refractory period of 2ms, a membrane time constant of 20ms, and a maximum firing rate of 200Hz. We applied random noise in the input current to each cell of σ=10% (see Equation 3). We also assumed NMDA neurotransmitter receptors, giving a time constant of 5ms for the post-synaptic current (*h(t)* in Equation 5).

To observe the dynamics, we ran a cleanup memory with D=500, M=10,000, and k=8. Over the course of 250ms of simulated time, we input five different noisy vectors for 50ms each. The output from the system was measured at each time step. Figure 7 shows the result of comparing the output of the model (the cleaned up vector) with the corresponding five original vectors. As in the rest of this paper, comparison was done by the dot product of the output vector and the desired clean vector.

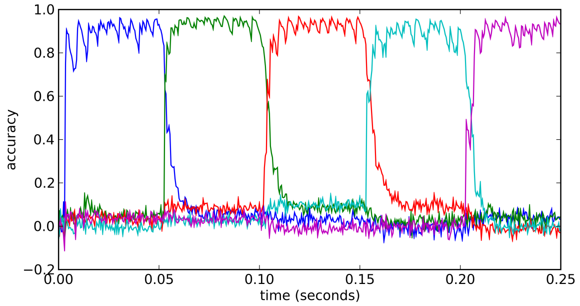


Figure 7: Temporal accuracy of the cleanup memory. Five noisy vectors are presented for 50msec each. Graphed lines show the dot product of the output of the network and the five original clean vectors.

These results indicate that the network reliably cleans the input vector and does so within 5-10 milliseconds. This makes our cleanup memory suitable for fast recognition, which is needed for symbolic manipulations at a cognitive time scale.

Conclusions

The model presented here is the first demonstration that a cleanup memory can be efficiently implemented by realistic spiking neurons. The number of neurons required to build this memory increases linearly in the number of distinct symbols that can be recognized. The accuracy approaches that of an ideal mathematical cleanup, and can perform cleanup in 5-10ms using realistically noisy spiking neurons.

Previous research (e.g. Eliasmith, 2005) has demonstrated realistic neurons performing the binding and superposition operations required for Vector Symbolic Architectures. Given the cleanup memory presented here, arbitrary symbol structures can be encoded, transformed, and recognized, all within a spiking network. As a result, we take this work to complete the currently most biologically plausible implementation of a symbolic cognitive architecture (Stewart & Eliasmith, 2009).

References

Anderson, J. R. and Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Eliasmith, C. (2005). Cognition with neurons: A large-scale, biologically realistic model of the Wason task. *Proceedings of the 27th Annual Meeting of the Cognitive Science Society.*

Eliasmith, C., & Anderson, C. (2003). *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems.* Cambridge: MIT Press.

Gayler, R. W., & Wales, R. (2000). Multiplicative binding, representation operators and analogical inference. *5th Australasian Cognitive Science Conference.*

Gayler, R. (2003). Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. *ICCS/ASCS International Conference on Cognitive Science*.

Georgopoulos, A.P., Schwartz, A.B., & Kettner, R.E. (1986). Neuronal population coding of movement direction, *Science 233*(4771), 1416-1419.

Hinton, G., & Andersen, J. (1989). *Parallel Models of Associative Memory*. Lawrence Erlbaum Associates, Inc.

Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*, 110(2), 220-264.

Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning representations by back-propagating errors. *Nature 323*, 533-536.

Kanerva, P. (1997). Fully distributed representation. *Proceedings of 1997 Real World Computing Symposium.*

Plate, T. (2003). *Holographic reduced representations.* Stanford, CA: CSLI Publication.

Pollack, J. B. (1988). Recursive auto-associative memory: devising compositional distributed representations. *Proceedings of the 10th Annual Conference of the Cognitive Science Society.*

Salinas, E., Abbott, L.F. (1994). Vector reconstruction from firing rates. *Journal of Computational Neuroscience 1*, 89-107.

Stewart, T.C. and Eliasmith, C. (2008) Building Production Systems with Realistic Spiking Neurons. *Proceedings of the 30th Annual Meeting of the Cognitive Science Society.*

Stewart, T.C. and Eliasmith, C. (2009). Compositionality and Biologically Plausible Models. In W. Hinzen, E. Machery, and M. Werning (Eds.), *Oxford Handbook of Compositionality*. Oxford University Press.

Sun, R. (2006). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In Ron Sun (ed.), *Cognition and Multi-Agent Interaction.* New York: Cambridge University Press.

van der Velde, F., & de Kamps, M. (2006). Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, 29, 37-70.

How Direct is Perception of Affordances? A Computational Investigation of Grasping Affordances

Giovanni Tessitore (tessitore@na.infn.it), Marta Borriello (aiamano82@gmail.com),
Roberto Prevete (prevete@na.infn.it), Guglielmo Tamburrini (tamburrini@na.infn.it)
University of Naples Federico II, Compl.Univ. M. S. Angelo, Naples, NA 80126 Italy

Abstract

The computational model presented here, Grasping Affordances (GA) model, provides a precise explication of the notion of affordance in the context of grasping actions carried out by monkeys. This explication is consistent with both direct perception theories and neuroscientific models of monkey brains, insofar as the identification of grasping affordances requires, according to this model, neither object recognition processes nor access to semantic memory. Nevertheless, this model posits a cascade of complicated computational processes, in the way of visuo-motor transformations, which suggest the advisability of qualifying and re-interpreting the claim that (grasping) affordances are directly available to an acting biological system. This re-interpretation undermines the alleged alternative between direct and indirect perception theories, to the extent that substantive visuo-motor transformations have to be posited in order to identify grasping affordances.

Keywords: affordances; visuo-motor transformations; direct perception; grasping

Background and Motivations

The notion of affordance was originally introduced by J. J. Gibson (Gibson, 1979) to single out perceived properties that enable one to interact with objects in the environment. Procedurally, the notion of affordance is framed in the context of *direct* perception theories, insofar as higher-level cognitive processes, such as access to semantic memory, logical inference, and object recognition processes are allegedly unnecessary to identify an affordance. Direct perception theories emerged in contrast with so-called indirect perception theories (Michaels & Carello, 1981). According to the latter, complex mental processing steps are needed to fill in the gap between impoverished descriptions of the world furnished by sensory inputs on the one hand, and the rich and accurate descriptions of the world delivered by perception on the other hand. Thus, in particular, perceiving a glass as a graspable object one can drink from is the final outcome of mental processes involving knowledge of what a glass is, what it can contain, and how one uses it.

A more precise understanding of the processes involved in identifying an affordance is crucial to isolate what is conceptually and empirically at stake in the controversy between direct and indirect perception theories. And an understanding of these processes is crucial for the modelling of specific sensory-motor control mechanisms in biological systems too. The existence of a particulary versatile sensory-motor control mechanism is witnessed by the wide range of sensory-motor associations that monkeys are able to perform. Notably, this behavioural ability persists upon presentation of many unknown/novel objects, thereby suggesting that a robust generalization process, based on perceived object properties, is at work there (Borghi, 2005).

In the context of grasping actions, neurophysiological data on the macaque's brain cortex are consistent with direct perception views of affordances. In particular, these data suggest that the anterior intraparietal area (AIP) is involved in the coding of object affordances (Rizzolatti & Sinigaglia, 2008), in the light of functional hypotheses concerning more extended brain circuits. The functional models of brain areas which have been found to deliver afferent signals to AIP include neither perceptual object recognition nor higher-level cognitive processes, such as planning and decision-making (Creem & Proffitt, 2000; Milner, 1998). Moreover, strong efferent pathways have been identified which connect AIP to pre-motor area F5 (Rizzolatti & Sinigaglia, 2008). Since F5 is prominently involved in the coding of object-oriented actions (such as grasping, holding, and manipulating), the AIP to F5 connections suggest the existence of some sort of *direct* functional link between perceptual feature detection and object-directed actions.

The computational model presented here, Grasping Affordances (GA) model, provides a precise explication of the notion of affordance in the context of grasping actions carried out by monkeys. This explication is consistent with both direct perception theories and neuroscientific models of the macaque's brain. It is consistent with direct perception theories, insofar as the identification of grasping affordances requires, according to the proposed computational model, neither object recognition processes nor access to semantic memory. It is consistent with neuroscientific models of the macaque's brain, insofar as (i) visual processes furnishing AIP inputs are modelled in accordance with the biological ”Standard Model” proposed in (Riesenhuber & Poggio, 2000), and (ii) the overall system output does not conflict with neuroscientific data and modelling constraints insofar as inputs supplied by AIP to brain motor areas are concerned. Nevertheless, this model posits a cascade of complex computational processes, in the way of visuo-motor transformations, which suggest the advisability of qualifying and re-interpreting the claim that (grasping) affordances are directly available to an acting biological system. This re-interpretation undermines the alleged alternative between direct and indirect perception theories, to the extent that substantive visuo-motor transformations have to be posited in order to identify grasping affordances.

The paper is organized as follows. First, a selective overview of neurophysiological findings about sensory-motor circuits in the macaque's brain cortex is provided, and basic features of computational models accounting for some of these data are briefly recalled. Then, an explication of the notion of affordance in the context of grasping actions is advanced. This explication sets the basic functional requirements for a computational model of grasping affordances, whose architecture and basic functionalities are described in some detail, and whose performances are evaluated on the basis of some preliminary tests. The import of this model on direct perception theories and future developments are briefly outlined in the concluding remarks.

Relevant Neurophysiological Findings and Computational Models

Brain areas in the macaque parietal and motor cortex were shown to be involved in a series of sensory-motor transformations, such as the mapping into appropriate actions of visual information about objects and their location in the visual scene (Rizzolatti & Sinigaglia, 2008). In particular, the AIP-F5 parieto-frontal circuit appears to play a crucial role in the visual guidance of hand grasping and manipulation movements, where AIP (Rizzolatti & Sinigaglia, 2008) was identified as a prominent cortical area involved in the coding of grasping affordances. One should be careful to note, moreover, that along the cerebral pathway starting from primary visual cortex (V1), and reaching F5 via AIP, visual information is transformed into motor information apparently without the intervention of cortical areas involved in higher-level perceptual and cognitive functions, such as the recognition of objects and their uses (Creem & Proffitt, 2000; Milner, 1998)

Two main computational models have been proposed in order to account for these data, by modelling AIP functionalities in the context of more comprehensive brain circuits. These are the FARS model (Fagg & Arbib, 1998) and a computational model of AIP neurons introduced in (Oztop, Imamizu, Cheng, & Kawato, 2006).

FARS is a neural model of cortical processes involved in generating and executing grasping plans. This model focuses on the interaction between AIP and premotor area F5, without providing a computational account of how inputs to area AIP are produced. In fact, affordances are ”programmed“ into this model, by hard wiring connections from units representing neurons in areas PIP and IT and units which represent neurons of area AIP. The connectivity between these units is determined by behavioural compatibilities. For example, an AIP unit which is selective for a specific grasp type and hand aperture receives inputs from units which hold input parameters of objects at which this kind of grasp and aperture are usually directed. Moreover, the model does not specify how these input parameters are computed from visual input. Their availability is taken for granted, and therefore the processing that visual information undergoes along the path from V1 to AIP is presupposed too. This comprehensive presupposition is acceptable in the FARS model, which is chiefly concerned with the generation and execution of grasping plans. It is not equally acceptable in a computational model which aims at accounting for the processes enabling one to extract affordances from visual inputs. For this reason, we have outlined here a computational account of contextually significant visuo-motor transformations occurring on the path from V1 to AIP.

The model proposed in (Oztop et al., 2006) concerns the development of AIP neuron functionalities while an infant is learning to perform grasp actions. This model focuses on an account of how units with processing properties similar to those of AIP neurons emerge by visuo-motor learning. Interestingly, the model demonstrates that units with different kinds of object selectivity emerge. In particular, units were found which encode object dimensions independently of object shape. This model exhibits limited generalization capabilities with respect to novel objects which do not belong to the initial training set. In fact, this generalization capability is restricted to transformations with respect to the size of known objects.

The model of grasping affordance extraction presented below (GA model) provides - unlike the FARS model - a detailed account of significant steps in perceptual processing along the path from V1 to AIP. In addition to this, the GA model is endowed - unlike the model proposed in (Oztop et al., 2006) - with more extended generalization abilities in the way of novel/unknown objects.

GA Model Description

Affordances for Grasping

Affordances are not intrinsic properties of an object, but rather depend on the relationship between object and agent (Chemero, 2003). For example, differences in primate and feline effectors account to a large extent for the different affordances that objects convey to humans and cats, respectively. As one moves to consider more specifically grasping affordances for monkeys and humans, one should still be careful to note that graspable objects do not merely ’afford’ our grasping them. Indeed, multiple opportunities for grasping arise in connection with many graspable objects. For example, a mug can be grasped by handle, lateral side, and top. These grasps can be distinguished from each other in terms of hand shape and wrist rotation obtaining just before grasping the object (Tucker & Ellis, 2000). Accordingly, the grasping affordances associated to a graspable object will be identified in the GA model with a collection of (codes for) appropriate hand configurations assumed by a hand just prior to grasping the object (Oztop et al., 2006; Tsiotas, Borghi, & Parisi, 2005). Since a graspable object may be grasped in several ways, this means that multiple hand configurations can be associated to any given object in the GA model.

General GA Model Description

From the above discussion, three main requirements have emerged for a computational model of grasping affordances to be empirically adequate and to move beyond previous computational models which include affordance extraction func-

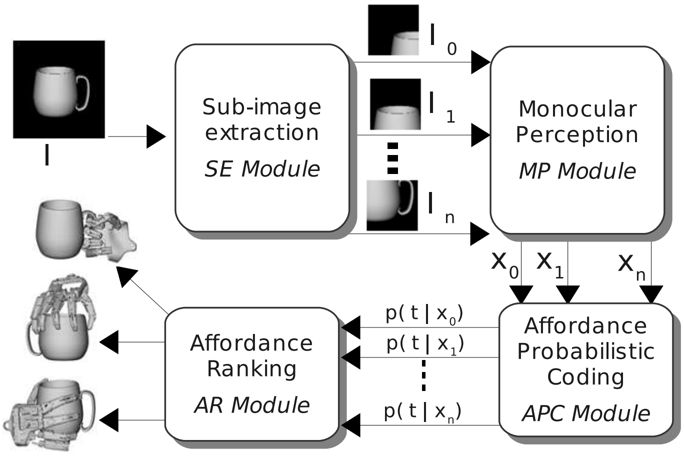


Figure 1: The GA model is formed by four modules: the SE Module, the MP Module, The APC Module, and the AR Module. This computational model receives an image depicting an object as input, and produces a list of affordances (appropriate grasps for the given object) as output.

tionalities: (a) the model must provide computational solutions for significant processing steps along the path from V1 to AIP; (b) the model must enable one to extract multiple hand-configurations from the same graspable object; (c) the model must possess generalization capabilities with respect to novel/unknown objects.

To accomplish (a), the visual pathway was modelled starting from primary visual cortex V1 and reaching, through areas V2 and V4, into the posterior infero-temporal area (PIT), which is identified as the cortical region supplying visual monocular information to AIP (Borra et al., 2007). A biologically plausible model of the ventral visual stream, named *Standard Model*, was proposed in (Riesenhuber & Poggio, 2000). A component of the Standard Model, the view-based Module, accounts for computations along the path from V1 to PIT which makes inputs available to AIP. Accordingly, the Monocular Perception (MP) Module (see Figure 1) which is an implementation of the view-based module was developed and included in the GA model.

To accomplish (b), that is, to provide a computational solution to the multiple affordance extraction problem, a probabilistic approach was pursued. In particular, this problem can be formalized as the problem of identifying and computing a multi-valued function which relates any visual input to a collection of hand-configurations. More precisely, let $X \subseteq \mathcal{R}^d$ be the d-dimensional space of visual inputs, and let $T \subseteq \mathcal{R}^c$ be the c-dimensional space of hand configurations. Then, one has to find a functional mapping f such that:

$$f : x \in X \longrightarrow \wp(T)$$

where $\wp(T)$ is the power set of T. A two-dimensional example of a multi-valued function is illustrated in Figure 2. This correspondence can be modelled by means of a probabilistic approach. More specifically, given $\bar{x}$, the output computed by the mapping f can be approximated by the unconditional probability density function $p(t)$. Thus, in general, the problem of modelling the functional mapping f can be viewed in terms of estimating the conditional distribution $p(t|x)$. A general framework for modelling conditional probability distributions makes use of mixture models whose parameters functionally depend on x (Bishop, 1995):

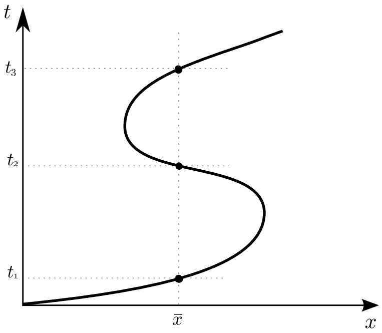


Figure 2: Two-dimensional example of a multi-valued function. Points on the x axis represent visual inputs, and points on the t axis represent hand-configurations. One may associate a x point with multiple t points.

$$p(t|x) = \sum_{k=1}^{M} \alpha_k(x)\phi_k(t|x) \tag{1}$$

The $\phi_k(x)$ are kernel functions, which are usually Gaussian functions of the form

$$\phi_k(t|x) = \frac{1}{(2\pi)^{c/2}\sigma_k^c(x)} \exp\left\{ -\frac{\|t-\mu_k(x)\|^2}{2\sigma_k^2(x)} \right\} \tag{2}$$

The parameters $\alpha_k(x)$ can be regarded as prior probabilities of t generated from the k-th component of the mixture. The *Affordance Probabilistic Coding* (APC) Module was designed so as to provide a computational solution to (b), that is, to the multiple affordance extraction problem (see Figure 1).

To accomplish (c), that is, generalization capabilities enabling one to extract affordances from novel objects, a starting point was provided by the observation that the agent usually focuses its attention on the part of the object at which the grasping action is directed (Schiegg, Deubel, & Schneider, 2003). This behaviour suggests the possibility of associating parts of a graspable object to affordances, and to store this “mereological“ information for use when novel graspable objects are presented. For example, one may learn to associate appropriate affordances to handles and cylinders, respectively, and to use this information when a cup (resulting from the “composition” of handle and cylinder) is presented. This process was actually implemented by sliding an “attention window” on the image of an object, and by extracting a collection of grasping affordances at each displacement step. This function is achieved by the Subimage Extraction (SE) Module (see Figure 1). Finally, a post-processing step was required as well, in order to select the more plausible affordances. The post-processing step is accomplished by Affordance Ranking (AF) Module (see Figure 1). APC and AR modules account for the AIP affordance computation. The online learning of sensorimotor associations might be grounded onto a basic grasping ability such as described in (Oztop, Bradley, & Arbib, 2004). Learning of sensorimotor associations may occur by collecting pairs of visually presented ”object part” and related ”hand-configuration” every time a successful grasp is made. Since the focus of this work is not on the acquisition of sensorimotor associations, however, we suppose here that a series of such pairs is already available.

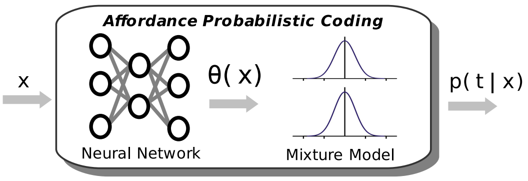


Figure 3: The APC Module is formed by a neural network and a Gaussian mixture model. Given an x vector, the neural network computes the required Gaussian parameters $\theta(x)$ to approximate $p(t|x)$ (see (Bishop, 1995) for more details).

GA Model specification and implementation

The GA model takes the image of an object as input and supplies the object’s grasping affordances as output. It is composed by four modules, as shown in Figure 1. The input image I, represented in gray scale, is processed by the SE Module, which extracts n subimages $I_j,\ j = 1,...,n$. The number of subimages depends on the dimensions of the window W sliding on the image I, the image size, and the window displacement step DS.

Each subimage is then sent as input to the MP Module. The MP Module takes a sub-image I_j as input, and gives a 256 feature vector as output x_j. The latter is presented as input to the APC Module, which computes the corresponding $p(t|x_j)$.

To estimate $p(t|x_j)$, one uses a mixture model of the form expressed in eq. 1, whose parameters $\alpha_k(x)$, $\mu_k(x)$ and $\sigma_k(x)$ (for Gaussian kernel as in eq. 2) depend on the visual input x. The relationship between visual inputs x and corresponding mixture parameters is modelled by means of a two-layer, feed-forward neural network with H hidden nodes. Therefore, the ACP Module has a combined density model and neural network structure, as shown in Figure 3.

Since the APC Module receives n feature vectors x_j in input, its overall output is formed by n density functions $p(t|x_j)$. Note, however, that the desired output is a set $T = \{t_1, t_2, \ldots, t_L\}$ corresponding to the L distinct hand-configurations that enable one to grasp the viewed object. Therefore, a non-trivial output selection problem remains to be solved at this stage: one has to isolate hand-configurations which differ from each other as much as possible, and whose probability value is sufficiently high.

This requirement corresponds, for each single feature vector x and related $p(t|x)$, to choose as member of the set T the gaussians’ centers $\mu_k(x)$ of the mixture associated to the higher values of $\alpha_k(x)$. In the case of n probability distributions $p(t|x_1), \ldots, p(t|x_n)$, in order to obtain a behaviour similar to the single distribution case, one may proceed as follow:

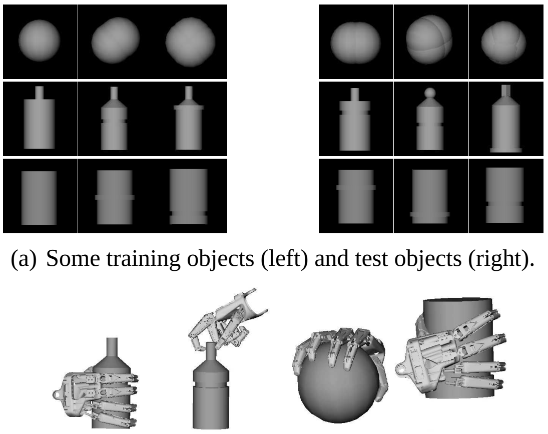
1. generate s sample points from each distribution, obtaining $n \times s$ points, each of which defines a hand configuration. Not every hand configuration thus obtained corresponds to grasps for the input object; only those gathering around the kernel’s means do, while the other points are distributed in a sparse manner;
2. a clustering over the $n \times s$ points is performed;
3. the clusters are ranked according to the order of their variance values, and the first L clusters with lower variances are selected because a lower variance implies less uncertainty about the hand configurations;
4. finally, the set T will be formed by the centers of the selected clusters.

Test and Results

The GA model was designed so as to extract multiple hand-configurations, and to generalize its affordance-extraction capability with respect to novel objects. Two experiments were performed to test the extraction and generalization abilities, respectively. The results of these tests corroborate the possession of the extraction ability, in addition to the required generalization ability as far as novel objects obtained from the composition of known object parts are concerned. Let’s see.

The first test, which is concerned with the extraction of multiple hand-configurations, makes use of three different prototypical object images: a sphere, a cylinder and a bottle. It is assumed that the first two objects can be grasped using a power grasp only, whereas the bottle can be grasped in two different ways, by precision and power grasps. For each of these prototypical object images, similar images were generated by means of small contour variations. For each prototype, the resulting training and test sets were composed by 20 and 10 images, respectively (Figure 4)

In order to generate target hand configurations, GraspIt! (Miller & Allen, 2004), a robotic grasping simulator, was used. In particular, the robotic hand called Robonaut, endowed with 14 degrees of freedom, was chosen. Consequently, in the GA model hand configurations are identified by a vector of 14 components, where each component represents just one hand joint’s angle. Spherical and cylindrical objects are associated to a single hand configuration, generated manually by changing the Robonaut’s degrees of freedom. Bottle objects are associated with two distinct hand configurations: a precision grasp, applied on the object’s top part, and a power one applied on the lateral part (see fig. 4). Training set targets are generated adding some Gaussian noise to



(a) Some training objects (left) and test objects (right).

(b) Target hand-configurations.

Figure 4: Examples of spherical, cylindrical and bottle objects used to train and test the system, and target hand-configurations.

these hand configurations. In this test, the attention window encompasses the whole object. Thus, for each object there is a single feature vector x with an associated $p(t|x)$. Hand configurations are obtained by selecting $\mu_k(x)$ associated with the higher values of $\alpha_k(x)$. The model parameters are summarized in table 2. For the i-th degree of freedom, percentage error is defined as $\frac{|t^i-y^i|}{max_i-min_i}\times 100$, where y_i is the model output, and max_i and min_i are the max and the min value, respectively, for the i-th degree of freedom. *Average error* between model output hand configuration and target hand configuration is defined as the mean of percentage error over all degrees of freedom. For all test objects in each class, mean and standard deviation of average error is computed and showed in table 1.

Table 1: For each object class, the mean and standard deviation of the average error over all objects in the test set is reported here. Moreover, for each class mean hand-configuration over all objects in the class is exhibited.

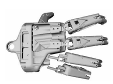
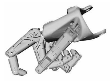
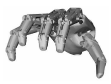
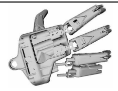
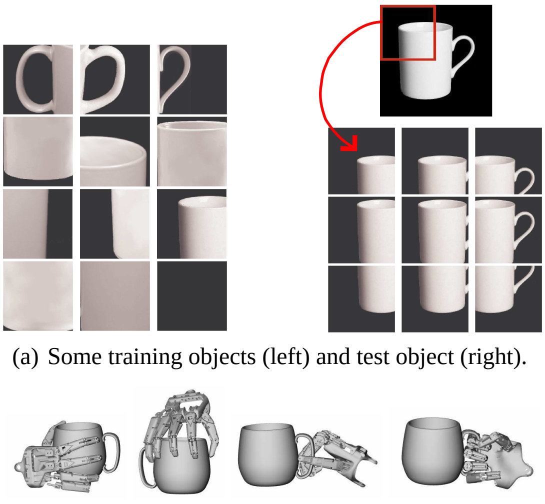
Bottle Grasp 1	Bottle Grasp 2	Spherical	Cylindrical
2% ± 0.4	1.9% ± 0.6	3.9% ± 1.4	1.3% ± 0.4
			

Table 2: Model parameters for each test. Image size, W and DS are expressed in pixels.

	H	M	Image size	W	DS	Cluster
Test 1	5	2	160 × 160	160 × 160	0	*None*
Test 2	5	5	500 × 500	160 × 160	30	5

The second experiment is meant to test generalization capabilities with respect to novel objects. To test this ability, the system was trained to associate *parts* of an object to hand-configurations. Subsequently, the system was given in input a novel object resulting from the ”composition“ of previously known parts. In this test, a cup is used, which is obtained from the composition of a cylinder and a handle. Examples of both training images and the cup used as test image are shown in figure 5. There are four kinds of training images: (a) cup handles; (b) upper and lower cup parts; (c) lateral cup parts; (d) non-graspable cup parts. Two target hand-configurations are associated with images (a); only one hand-configuration is associated to images (b) to (d). The training set targets are generated adding some Gaussian noise to hand configurations. Targets for non-graspable cup parts images are drawn from a Gaussian distribution with a large variance, so as to reflect the fact that in this case no plausible hand-configuration candidate exists. The K-Mean clustering algorithm is implemented by the AR Module, setting to 5 the number of clusters. In table 3, cluster centroids are shown together with cluster variance. The fifth cluster was discarded in view of its large variance. Note that the first four cluster centroids are very similar to target hand configurations (fig. 5) with respect to which mean percentage error was computed.



(a) Some training objects (left) and test object (right).

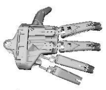
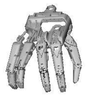
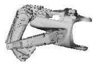
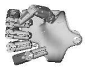
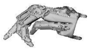
(b) Target hand-configurations.

Figure 5: (a) Examples of training and test images (see text). (b) Examples of target hand-configurations.

Concluding remarks

The architecture of the GA model is largely motivated by the goal of computationally investigating the allegedly direct link between perception and action established by the perception of affordances. One should be careful to note that the overall output of the GA model does not correspond to actions, but rather corresponds to hand configurations. Therefore, one may legitimately question the claim that the GA model computes a perception-action transformation. However, in

Table 3: The graph visualizes the obtained cluster centroids. Compare these images with target hand configurations of fig. 5. The fifth cluster was discarded in view of its large variance. The percentage error with respect to target was mediated over all degrees of freedom.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
				
$\sigma = 0.12$	$\sigma = 0.12$	$\sigma = 0.09$	$\sigma = 0.09$	$\sigma = 0.34$
Mean and standard deviation of percentage error				
1.9% ± 2	2.5% ± 2	2% ± 1.2	1.8% ± 1.5	*(discarded)*

the context of grasping actions, the model embodies the assumption that an appropriate hand configuration for grasping an object is a configuration assumed by a hand just prior to grasping that object. This configuration is closely related to the *goal* of the grasping action. Thus, the grasping action can be generated from the initial configuration, in terms of motor commands, by a forward model on the basis of such goal-related information. For this reason, one can meaningfully maintain that the computation of hand configurations from visual inputs performed by the GA model is the gist of a perception-action transformation.

As discussed in the first section, a more precise understanding of the processes involved in identifying an affordance is crucial to isolate conceptual and empirical differences between direct and indirect perception theories. The GA computational model is in agreement with the notion that the identification of affordances does not require higher cognitive processes, such as logical inference and object classification. However, the transformation performed in the GA model requires a cascade of fairly complicated processing stages, and the solution of non-trivial computational problems. Notably, in order to achieve significant generalization capabilities, the APC module was geared so as to produce in output a set of probability distributions each one of them expressed as a Gaussian mixture, coding hand configurations for just one part of the image. Here, the pertinent modelling question is: how one does choose the appropriate hand configurations for the object? In the case of just one probability distribution, a natural candidate are the centers of the Gaussians associated to the higher mixture coefficients. In the case of a set of probability distributions, various possibilities arise, only one of which was pursued in the GA model. This solution provides a significant proof-of-concept, together with a vivid illustration of the important qualifications that are needed when one makes use of the attribute direct in the expression direct perception of affordances. An alternative solution, which we are currently exploring, involves a unique probability distribution, which arises by taking as some sort of union over the set of distributions based on a similarity measure between gaussian mixture models (Hershey & Olsen, 2007).

References

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Borghi, A. M. (2005). Grounding cognition: The role of perception and action in memory, language, and thinking. In (chap. Object concepts and action). Cambrige University Press.

Borra, E., Belmalih, A., Calzavara, R., Gerbella, M., Murata, A., Rozzi, S., et al. (2007). Cortical connections of the macaque anterior intraparietal (AIP) area. *Cerebral Cortex*.

Chemero, A. (2003). An outline of a theory of affordances. *Ecological Psychology*.

Creem, S. H., & Proffitt, D. R. (2000). Defining the cortical visual systems: "what", "where", and "how". *Acta Psychologica*.

Fagg, A., & Arbib, M. (1998). Modeling parietal-premotor interactions in primate control of grasping. *Neural Networks*.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin.

Hershey, J. R., & Olsen, P. A. (2007). Approximating the kullback leibler divergence between gaussian mixture models. In *Icassp 2007. ieee international conference on acoustics, speech, and signal processing.*

Michaels, C. F., & Carello, C. (1981). *Direct perception*. Englewood Cliffs, New Jersey 07632: Prentice-Hall, Inc.

Miller, A. T., & Allen, P. K. (2004). Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*.

Milner, A. D. (1998). Neuropsychological studies of perception and visuomotor control. *Philos Trans R Soc Lond B Biol Sci*.

Oztop, E., Bradley, N. S., & Arbib, M. A. (2004). Infant grasp learning: a computational model. *Experimental Brain Research*.

Oztop, E., Imamizu, H., Cheng, G., & Kawato, M. (2006). A computational model of anterior intraparietal (AIP) neurons. *Neurocomputing*.

Riesenhuber, M., & Poggio, T. (2000). Models of object recognition. *Nature Neuroscience*, *3*, 1199–1204.

Rizzolatti, G., & Sinigaglia, C. (2008). *Mirrors in the brain. how our minds share actions and emotions.* Oxford University Press: Oxford.

Schiegg, A., Deubel, H., & Schneider, W. (2003). Attentional selection during preparation of prehension movements. *Visual Cognition*.

Tsiotas, G., Borghi, A., & Parisi, D. (2005). Objects and affordances: An artificial life simulation. In *Proceedings of the cognitive science society.*

Tucker, M., & Ellis, R. (2000). Micro-affordance: the potentiation of components of action by seen objects. *British Journal of Psychology*.

A memory for goals model of sequence errors

J. Gregory Trafton
(greg.trafton@nrl.navy.mil)
Naval Research Laboratory
Washington, DC 20375 USA

Erik M. Altmann
(ema@msu.edu)
Michigan State University
East Lansing, MI 48824 USA

Raj M. Ratwani
(raj.ratwani@nrl.navy.mil)
Naval Research Laboratory
Washington, DC 20375 USA

Abstract

We propose a model of routine sequence actions based on the Memory for Goals model. The model presents a novel process description for both perseveration and anticipation errors, as well as matching error data from a previously collected dataset. Finally, we compare the current model to previous models of routine sequential action.

Keywords: routine sequential actions; errors, cognitive modeling

Introduction

Several researchers have described classes of errors that people make as they perform routine sequential actions (Norman, 1981; Reason, 1990; Baars, 1992). Most of the categorization for these errors comes from either diary studies (Reason, 1990) or from neurologically damaged patient studies (Schwartz et al., 1998).

Sequence errors occur during routine action and consist of perseverations, omissions/anticipations, and intrusions (Reason, 1984). Perseveration errors are repeats of a previous action and come in two forms (Sandson & Albert, 1984). Continuous perseveration errors occur when an action is performed over and over. Recurrent perseveration errors occur when a previously completed subtask is performed again, usually with one or more intervening subtasks. For example, putting cream in a cup of coffee multiple times is a perseveration error. Omissions are skipped steps, while anticipation errors are skipped steps that are quickly rectified. For example, an omission error would be completely forgetting to put cream in a cup of coffee, while an anticipation error would be attempting to pour from an unopened container. It can be quite difficult to differentiate omission and anticipation errors (Cooper & Shallice, 2000). Intrusion errors (sometimes called capture errors) occur when an action comes from a different, usually related, task. For example, a capture error would occur when attempting to make coffee a person gets distracted by a tea bag and instead makes tea.

There are other types of errors that occur during routine action, but this report will focus on sequence errors.

Previous models of sequential behavior

There are two computational models of routine sequential behavior: the interactive activation network (IAN) model (Cooper & Shallice, 2000; Cooper & Shallice, 2006) and the simple recurrent network (SRN) model (Botvinick & Plaut, 2004; Botvinick & Plaut, 2006).

In the IAN model, different schemas compete for activation. Activation comes from triggers (environmental or context) and source-schemas (related schemas), but a schema will not be activated if it is not over a specific threshold. Thus, while working on a routine task, the selection of a schema is influenced by the current schema and the state of the world. The IAN model suggests that errors are caused by a lack of attentional resources or distraction in normal populations (Norman & Shallice, 1986; Cooper & Shallice, 2000). Variability in attentional resources is instantiated in IAN by noise. In the case of sequence errors, noise has two major effects. First, noise in the system can cause variability in the ordering of schemas that do not have ordering constraints. Second, noise can cause variability in the selection of which schema is selected when multiple schemas are applicable. Both these forms of variability can cause various sequence errors.

The SRN model has a set of input units that are activated by features of the environment. Activation is passed along the input units to a set of hidden units, which receive recirculated activation. The hidden units then pass activation to a set of output units that then perform an action (fixating an object, pouring an object, etc.). The connection weights encode series of sequential attractors which the trained model tends to follow (Cooper & Shallice, 2006). Errors in the SRN model are made by increasing the noise, which in turn can cause the network to drift to a related task sequence (i.e., a sequential attractor) whose internal representation resembles the next step. Thus, an error is made by the SRN model not because an attentional operation has been omitted, but because the model's internal representations have resulted in a loss of information about a previous or current state (Botvinick & Plaut, 2004; Botvinick & Bylsma, 2005).

The Memory for Goals model

A different model of routine behavior is the memory for goals model (MFG) which is an activation-based model that has been used in the study of interruptions and goal-related tasks (Altmann & Trafton, 2002; Trafton, Altmann, Brock, & Mintz, 2003; Altmann & Trafton, 2007).

The MFG is based on the hypothetical construct of activation of memory items—in particular, activation as construed in the ACT-R (Adaptive Control of Thought-Rational) cognitive theory (Anderson & Lebiere, 1998). A basic processing assumption in this theory is that when central cognition queries memory, memory returns the item that is most active at that instant. Activation thus represents relevance to the current situation. To capture the relevance of any particular item, the memory system computes that item's activation from both the item's history of use and from its associations to cues in the current mental or environmental context. In Bayesian terms, the logic is that history of use and current context together serve to predict the current relevance of that item (Anderson, 1990). In functional terms, the implication is that the cognitive system should be able to exploit the predictive computations of the memory system to overcome decay and keep certain information active for use in the future.

Two main constraints determine goal activation: strengthening and priming. The strengthening constraint suggests that the history of a goal (i.e. frequency and recency of sampling) will impact goal activation such that a subgoal that is retrieved more often or the most recently retrieved subgoal will have a higher activation value than others with less history. The priming constraint suggests that associated cues in the mental or environmental context can provide activation to a pending goal. For example, particular information in a task interface may prime a subgoal, allowing the subgoal to be retrieved over competing subgoals. In addition, each procedural step is associatively linked to the next step within the task hierarchy; thus, previously completed tasks are a source of associative activation (Altmann & Trafton, 2007).

The model incorporates the assumption that cognitive control is mediated at a fine-grained by episodic codes passed between different processes (Altmann & Gray, 2008). Applied to cognitive control here, in the context of routine sequential behavior, the assumption is that action preparation and action execution are separate processes, with the first retrieving a procedural step from semantic memory, then communicating with the second by creating an episodic code that represents the retrieved task. The communication between these two processes can be disrupted if some other cognitive operation (e.g., an interruption) occurs after the first process has executed but before the second has started.

All three models have different process explanations and capabilities for accounting for sequence errors.

Perseveration Errors

The IAN model does occasionally repeat steps, resulting in a continuous perseveration error. This occurs when, due to too much self-activation or lack of inhibition, a schema is not deselected at the appropriate time, causing a schema to be repeatedly selected. The IAN model can not, however, account for recurrent perseveration errors because once a goal is completed it is "ticked off" and not applicable for later selection (Cooper & Shallice, 2000; Botvinick & Plaut, 2004).

The SRN model does make both continuous and recurrent perseveration errors. However, one interesting aspect of the original SRN model was that virtually all errors were due to capture errors but had different manifestations. For example, with a small amount of noise, the network would occasionally drift to a similar sequential attractor (a capture process) and repeat a step (a perseveration error) (Botvinick & Plaut, 2004; Cooper & Shallice, 2006). While it is interesting that the SRN model can elicit so many error types, the capture process that causes those errors to occur has been questioned by some (Cooper & Shallice, 2006).

The MFG model can make both types of perseveration errors, though the process explanation is the same for both continuous and recurrent perseveration. The reason that MFG makes perseveration errors rests primarily on the interference level. Perseveration errors may occur when the wrong subgoal is retrieved to direct behavior. Occasionally, the difference in activation levels between previous subgoals and the target subgoal may be quite small and noise in the cognitive system may result in the retrieval of an incorrect subgoal. The constraints of the memory for goals theory suggest that when an incorrect subgoal is retrieved, it should be in close temporal proximity to the target subgoal. Recency suggests that the subgoal just completed will have a relatively high activation level and associative activation from the most recently retrieved subgoal will provide activation to neighboring subgoals. Occasionally, then, the cognitive system may retrieve the wrong subgoal to direct behavior. This will occur especially when there are relatively few environmental cues so that priming has less of an impact. Interestingly, the MFG model predicts that errors should be proximate to the next correct action. Not only should the most common error action be to retrieve the subgoal just completed, other error actions should be to subgoals that are temporally close to the next correct action. Recency suggests that the last few steps prior to the next correct action will have relatively high activation levels. The farther away the subgoal is from the correct action, the less likely this step should be retrieved. Thus, the general prediction is that when perseveration errors are made, most of the error actions should be localized to within a few steps of the correct action in a graded fashion.

Anticipation and Omission Errors

The IAN model also makes anticipation and omission errors. Omission errors could occur because a schema may not have a high enough activation due to low self-activation or poor environmental influences. Anticipation errors occur for a similar reason, but are not able to be executed because a precondition was not satisfied (e.g., a container still has its top attached).

The SRN model occasionally makes anticipation and omission errors, primarily through the capture process described before.

The MFG model also suggests that anticipation and omission errors will occur. In fact, MFG suggests that there are two possible explanations for skipping a goal. First, the primed retrieval component of the theory suggests that future steps receive activation in a decreasing graded fashion (Altmann & Trafton, 2007). Second, the model suggests that action preparation and action execution are separate processes. If communication between these two stages gets disrupted, an anticipatory error may occur. Because the primed retrieval model is not yet implemented in ACT-R, the separate-stages explanation will be focused on in the remainder of this report.

While all three models can account for the majority of error types, neither IAN nor SRN makes strong predictions about which types of errors should be more prevalent in this type of task. MFG, however, makes a strong prediction that perseveration errors should occur more often than any other type of sequence error. Additionally, MFG makes a nuanced prediction that errors should be proximate and graded from the correct step, especially with respect to perseveration errors.

Experiment

There are very few datasets that can be used to constrain or reject different models (Botvinick & Plaut, 2006). One of the issues is that the when a task is routine, people generally make very few errors, making statistical analysis difficult. Thus, different researchers have examined errors in non-routine tasks (Ruh, Cooper, & Mareschal, 2005), made the task difficult to remember (Giovannetti, Schwartz, & Buxbaum, 2007; Ruh, Cooper, & Mareschal, 2008) or interrupted participants during the routine task (Botvinick & Bylsma, 2005). We used an interruption paradigm because interruptions have been shown to increase error rates even on well-learned tasks (Li, Blandford, Cairns, & Young, 2008; Ratwani, McCurry, & Trafton, 2008). In addition, we provided no global placekeeping (Gray, 2000) such that the next step of the task could not be determined from visible cues.

Method

Participants. Fifteen George Mason University students participated for course credit.

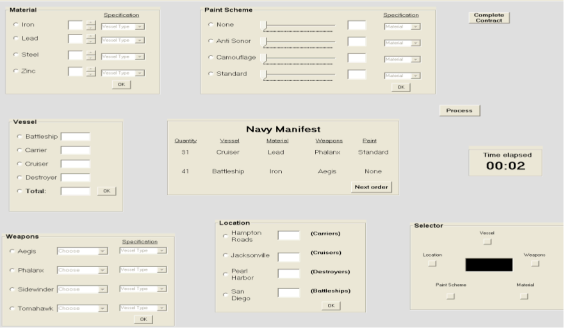
Task and Materials. The primary task was a complex production task called the sea vessel task (based on Li et al., 2008; Ratwani et al. 2008). The goal was to fill an order for two different types of sea vessels by entering in order details through various widgets on the interface (Figure 1). Order information was provided in the middle of the screen on the "Navy Manifest." A correct sequence of actions is required to complete the order: (1) Enter Vessel Information, (2) Material, (3) Paint, (4) Weapons, and (5) Location. Before entering information into each widget, the widget must be "activated" by clicking the corresponding selector button (lower right hand corner of Figure 1). The procedure was arbitrary, but participants had no trouble learning it because (1) the information that was needed to fill in the widgets was available on the Navy Manifest; and (2) the order of the widgets was straightforward to remember due to a simple spatial rule, which we provided to participants.

After completing each widget, the participant must click "ok" and the information that was entered in the fields is no longer visible. This information was cleared from the fields because it may have served as an explicit cue indicating which steps in the task hierarchy have been completed. After entering information in each of the five widgets, the order must be processed by clicking the "Process" button. Once this button is clicked, a small pop-up window appears informing the participant of the total number of sea vessels that have been created. This pop-up window served as a false completion signal (Reason, 1990). Participants must click the "ok" button to acknowledge this window. Finally the "Complete Contract" button must be clicked to finish the order. The "Next Order" button is clicked to bring up a new order. Any deviation from this procedure was recorded as an error; any time an error was made, the computer emitted a brief auditory tone to alert the participant that an error was made. When a participant committed an error the participant had to continue with the task until the correct action was made.

The interrupting task required participants to answer addition problems with four single digit addends.

Design and Procedure. Each order on the sea vessel task constituted a single trial; participants performed twelve trials. Control and interruption trials were manipulated in a within-participants design; half of the trials were control with no interruption and half were interruption trials with two interruptions each. The order of trials was randomly generated. There were six predefined interruption points in the sea vessel task. There was a potential interruption point after clicking "ok" in each of the five widgets. The sixth interruption point was after the "Process" button was clicked. During the experiment there were a total of 12 interruptions (6 interruption trials x 2 interruptions in each trial); each lasting 15 seconds. Participants were instructed to answer as many addition problems as possible in this time interval. The interruptions were equally distributed among the six interruption locations. When returning to the primary task after the interruption, there were no visual cues on the task interface indicating where to resume (i.e. no global place keeping).

Before beginning the experiment, participants were given instructions about the two tasks they were going to have to perform and completed two trials as part of training; one had no interruptions and one had two interruptions. All participants were proficient at the task before beginning the actual experiment. The experiment was self-paced. A break was offered after six trials.



Navy Manifest

Quantity	Vessel	Material	Weapons	Paint
31	Cruiser	Lead	Phalanx	Standard
41	Battleship	Iron	Aegis	None

Figure 1: Screenshot of the ship production task

Description of Errors. Perseveration errors were any actions that repeated an action that had already been accomplished for that trial. Anticipation and omission errors were any actions that skipped one or more steps.

Because it was not possible to actually omit a step, all skipped steps were categorized as anticipation errors. Errors where participants failed to activate a particular module before working on the module (e.g. device initialization errors (Cox & Young, 2000)) were not analyzed.

Measures. Error rates were calculated for control and interruption trials by calculating percentages (actual errors/total error opportunities). Multiple incorrect actions in a sequence were counted as a single error for the purposes of calculating error rates. Error actions that occurred less than 500 ms from the previous action were excluded from all analyses as they were taken to be inadvertent mouse clicks; this accounted for less than one percent of the data.

Results and Discussion

Comparing Error Rates. Of the fifteen participants, eleven participants made at least one perseveration or anticipation error. Error rates were compared between the control trials and actions immediately after the interruption using a repeated measures ANOVA. Participants made more errors following an interruption (M = 9.3%) compared to the control (M = .9%), F (1, 14) = 5.8, MSE = 91.9, p<.05. Participants rarely made errors in the control trials, suggesting the task was well learned. The non-zero error rate on control trials also matches studies showing that people do make errors on well-learned tasks (Reason, 1990).

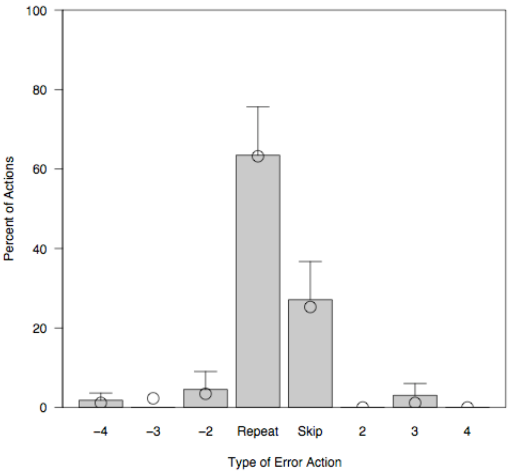


Figure 2: Distribution of errors during a sequential action task. Bars are empirical data; circles are model fits.

Pattern of Error Actions. Next, we focused on the pattern of error actions. In order to compare error actions at different points in the task hierarchy, the error actions were coded relative to the correct action at that point in the task hierarchy. Recall that the correct order of actions was Vessel, Material, Paint Scheme, Weapons, Location, Process and Complete Contract. If the next correct action is to work on the “Weapons” subtask and the participant made the error of working on the “Paint” subtask, this error action was coded as a “-1”. If instead the participant clicks the “Process” button this was coded as a “2”. Based on this coding scheme, a “-1” represents a repeat of the just completed action and a “1” represents skipping the next correct action. All errors were coded using this scheme.

The distribution of error actions is illustrated in Figure 2. A visual inspection of this graph suggests that both perseveration and anticipation errors occur relatively frequently. Additionally, the number of errors seems to be proximate to the next correct action in both directions, though this effect is not strong in this dataset. To determine whether the error action of retrieving the subgoal just completed and performing this action again was the most common error action, a repeated measures ANOVA was conducted to compare error actions at this position to all other error actions. There was a significant difference among the different error positions, F(7,70) = 12.8, MSE = 434.2, p<.0001. Tukey HSD post-hoc comparisons revealed that participants were significantly more likely to repeat the subtask just completed (M = 63.5%) than to make any other action (all p’s<.05).

Model Description

An MFG model was written in the ACT-R cognitive architecture.

High Level Description of the MFG model

There are five model components that are critical for routine sequential skill and errors that occur during execution of a routine task: the need for well-learned knowledge; the encoding of an episodic trace; the strengthening constraint, the priming constraint, and the interference level.

Well-Learned knowledge There are several ways to represent well-learned knowledge in ACT-R. We provided the model with declarative knowledge about the task such that it always knew the sequence of steps it should follow.

Encoding of an episodic memory When the model knows which step it should perform, it encodes an episodic memory. A separate ACT-R module (goal-style), called episodic was created for this purpose. An episodic memory in this task is an extremely lean memory item that contains the current goal and a unique identifier. This unique code helps differentiate an episodic memory from a semantic one. All episodic memory items are created with a slightly higher initial activation so that they can be retrieved later. This mechanism is very similar to other models (Altmann & Trafton, 2002; Altmann & Gray, 2008); we propose that people encode and retrieve episodic memories during interactive routines. This episodic trace is later retrieved to guide action; retrieval is biased by the strengthening constraint, the priming constraint, and the interference level.

Strengthening constraint Which episodic memory element is retrieved depends in part on the strengthening constraint. The strengthening constraint suggests that the most recent episode will have the highest activation.

Priming Constraint When the model attempts to retrieve an episodic memory element, activation spreads from the focus of attention to related elements, of which the relevant episodic memory element is one. Thus, the mental context provides context to facilitate the retrieval of the correct episodic trace. The environmental context could also provide priming, but that aspect is not implemented in the current model.

Interference Level When the model attempts to retrieve an episodic trace (or any other memory item, for that matter), there is interference from other similar memory items. Interference occurs because a memory request is made that does not contain a perfect cue for retrieval. Since there may be several items that match the memory request, the system retrieves the most active memory element. Transient noise (sampled from a zero-mean logistic density function) can cause older elements to be retrieved. Thus, interference can lead to retrieval of an incorrect episodic memory item.

For all models, we kept most of the ACT-R parameter defaults. Specifically, we enabled several parameters with typical ACT-R values, including the maximum associative strength parameter which is priming (from nil to a typical value of 3), activation noise (from nil to .03), and the randomize-time parameter, which allows some perceptual and motor actions to have a small amount of variability in their timings (we kept the default value of 3). The base level learning parameter was set at the default of .5.

A sample experimental model run

To provide a match to the experimental procedure, 15 models (15 participants) were run. An abstracted interface was used for model runs. The model did not perform the post-completion step (Byrne & Bovair, 1997).

Normal processing The first thing that the model does in an experimental trial is to prepare to make a step. In order to do this, it retrieves from declarative memory the first step to perform (well-learned knowledge). Next, the model encodes an episodic memory of that step (encoding of the episodic memory). This retrieval and encoding is the preparation component of the model. Next, the model must execute the action. The execution component of the model begins with an immediate attempt to retrieve that episodic memory. Because the current mental context primes the episodic memory (priming constraint) and it is the most recent (strengthening constraint), the correct episodic memory is highly likely to be retrieved. After retrieving an episodic memory, that action is executed, the next step in the procedure is retrieved (well-learned knowledge), and the whole process repeats. Note that as the model completes one action, it starts to prepare for and encode the next step. This interleaving of motor and mental actions has been shown to occur in a variety of tasks and contexts (Salvucci & Taatgen, 2008).

Interruption processing When the model notices there was a screen change, it starts working on the interruption. The interruption effectively clears out all state information from the primary task. According to the model, the two most important aspects of the interruption are that (1) state information from the primary task is cleared and (2) decay occurs during the interruption. In the current model, only cursory model processing occurs during the interruption and all state information (e.g., focus-of-attention and problem representation) is cleared.

Resumption processing After an interruption completes, the model notices the screen change and attempts to remember the last task-relevant episodic memory. If it is unable to recall an episodic item, the model executes a random action. This rarely happens in the current model, given the brief interruption duration. If the model is able to retrieve an episodic memory, it assumes that the retrieved element was the last completed action so retrieves the next step and continues in the task.

Error behavior Most of the time, due to the strengthening and priming constraints, the correct episodic memory is retrieved and the procedural task is executed flawlessly. During normal execution, however, the model will rarely (when transient noise of an older episodic trace is greater than strengthening and priming), retrieve an incorrect episodic trace (interference level). When an error is made, the model suggests that the most likely memory element to be retrieved will be the one with the next highest activation.

The model makes perseveration errors because the episode that was just completed is likely to have a relatively high activation. Thus, the model makes perseveration errors in a graded fashion away from the correct action.

The model makes anticipation errors because sometimes the model pre-encodes a particular episodic action before it gets completed (e.g., it encoded an episode but got interrupted before it could complete that action). When this pre-encoding / interruption occurs, the episodic element with the highest activation is likely to be the next (uncompleted) action upon resumption and therefore selected, leading to an anticipation error. Note that when the model makes an anticipation error, it is a simple skipped step and can not skip more than one step.

As in the empirical data the model very rarely makes an error during non-interrupted trials. These errors occur because the wrong episodic memory was retrieved: noise in the interference level overcomes the strengthening and priming constraints of the correct episode.

The role of noise Greater noise in the system increases the number of errors the system makes because there is a greater probability that a different episodic memory will have a higher activation than the correct one. Additionally, a greater noise increases the “spread” of applicable episodes. So, increasing noise increases both the number and spread of errors.

Model fit

As is evident in Figure 1, the model matches the data quite well; $R^2 = .99$ and RMSD = 1.3.

General Discussion

The current paper presents an experiment and model of sequential actions. The experiment used an interruption paradigm, increasing the rate of errors enough to see emergent patterns from the data. The model used a memory for goals model that describes the process people go through both during error-free behavior and when they make errors. In general, errors occurred because the wrong episodic memory was retrieved. Perseveration errors occurred because a recent episodic memory had a high enough activation that, with noise, it was retrieved instead of the correct memory. Anticipation errors occurred because the communication between the preparation and execution of an action gets disrupted for some reason.

The MFG model shares both similarities and differences to the other two models of sequential routine action, IAN and SRN. MFG focuses on perceptual and memorial processes rather than schemas (IAN) or distributed representations (SRN). However, it is interesting that all three models use noise as one of the primary explanatory constructs for why errors are made.

The current MFG model does have several limitations. First, it only accounts for sequence errors; it does not account for intrusions, capture errors, etc. Second, while both IAN and SRN attempt to model both normal and patient populations, the MFG model only addresses normally functioning individuals. Third, the model-task is quite simple, and a more complete task description is needed to expand the coverage of this model. Finally, the MFG model does not model the learning of the task itself.

The experiment reported here and the MFG model itself do, however, have several strengths. First, the experimental paradigm used here allows errors to be studied in the lab with normal populations. This data and other like it should be able to constrain current models of sequential actions, as Botvinick and Plaut (2006) suggest. Second, the MFG makes both qualitative and quantitative predictions about the error pattern for this task. Both the IAN and SRN models have been critiqued for the way they make perseveration errors. Finally, the model makes episodic memory an aspect of its normal processing, so errors arise out of normal processing of routine behavior.

Acknowledgments

This work was supported by ONR under funding document N0001409WX20173 to JGT and N000140310063 to EMA. The views and conclusions contained in this document should not be interpreted as necessarily representing the official policies of the U. S. Navy.

References

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science*, *26*(1), 39-83.

Altmann, E. M., & Trafton, J. G. (2007). Timecourse of Recovery from Task Interruption: Data and a Model. *Psychonomic Bulletin & Review*, *14*(6), 1079-1084.

Altmann, E. M., & Gray, W. D. (2008). An Integrated Model of Cognitive Control in Task Switching. *Psychological Review*, *115*(3), 602-639.

Anderson, J. R. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiere, C. (1998). *Atomic components of thought.* Mahwah, NJ: Erlbaum.

Baars, B. J. (1992). *Experimental slips and human error: Exploring the architecture of volition.*

Botvinick, M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to routine sequential action. *Psychological Review*.

Botvinick, M. M., & Bylsma, L. M. (2005). Distraction and action slips in an everyday task: Evidence for a dynamic representation of task. *Psychonomic Bulletin and Review*.

Botvinick, M. M., & Plaut, D. C. (2006). Such stuff as habits are made on: A reply to Cooper and Shallice (2006). *Psychological Review*.

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive science*.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*.

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological review*.

Giovannetti, T., Schwartz, M., & Buxbaum, L. (2007). The Coffee Challenge: A new method for the study of everyday action errors. *Journal of Clinical and Experimental Neuropsychology*, *29*(7).

Gray, W. D. (2000). The nature and processing of errors in interactive behavior. *Cognitive Science*, *24*(2), 205-248.

Li, S. Y. W., Blandford, A., Cairns, P., & Young, R. M. (2008). The effect of interruptions on postcompletion and other procedural errors: An account based on the activation-based goal memory model. *Journal of Experimental Psychology: Applied*, *14*(4), 314-328.

Norman, D. A. (1981). Categorization of action slips. *Psychological review*.

Norman, D. A., & Shallice, T. (1986). *Attention to action: willed and automatic control of behaviour.* Paper presented at the Consciousness and Self-Regulation: Advances in research and theory, New York.

Ratwani, R. M., McCurry, J. M., & Trafton, J. G. (2008). Predicting postcompletion errors using eye movements. *CHI*.

Reason, J. (1984). Absent-mindedness and Cognitive ControI. *Everyday Memory Actions and Absent-mindedness: Actions and.*

Reason, J. (1990). Human Error. *Cambridge University Press*.

Ruh, N., Cooper, R. P., & Mareschal, D. (2005). The time course of routine action. *Cognitive Science 2005*.

Ruh, N., Cooper, R. P., & Mareschal, D. (2008). The Hierarchies and Systems that Underlie Routine Behavior: Evidence from an Experiment in Virtual Gardening. *Cognitive Science 2008*.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*.

Sandson, J., & Albert, M. L. (1984). Varieties of perseveration. *Neuropsychologia*.

Schwartz, M. F., Montgomery, M. W., Buxbaum, L. J., Lee, S. S., TG,. (1998). Naturalistic action impairment in closed head injury. *Neuropsychology*.

Trafton, J. G., Altmann, E. M., Brock, D. P., & Mintz, F. E. (2003). Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human Computer Studies*, *58*(5), 583-603.

An embodied model of infant gaze-following

J. Gregory Trafton
(greg.trafton@nrl.navy.mil)
Naval Research Laboratory
Washington, DC 20375 USA

Anthony M. Harrison
(anthony.harrison@nrl.navy.mil)
Naval Research Laboratory
Washington, DC 20375 USA

Benjamin Fransen
(fransen@aic.nrl.navy.mil)
Naval Research Laboratory
Washington, DC 20375 USA

Magdalena Bugajska
(magda.bugajska @nrl.navy.mil)
Naval Research Laboratory
Washington, DC 20375 USA

Abstract

We present an embodied model of gaze-following. The model learns how to follow another's gaze by using cognitively plausible mechanisms. It matches a classic gaze-following experiment (Corkum & Moore, 1998) and runs on an embodied robotic system.

Keywords: infant gaze-following; embodied cognition; robotics; cognitive architectures

Introduction

Gaze-following is an important, early component of joint visual attention (Scaife & Bruner, 1975; Butterworth & Jarrett, 1991). Joint visual attention is looking at the same object as another person. Some researchers have suggested that joint visual attention is strongly related to the ability to infer others' mental states (Baron-Cohen, 1995). More recently, researchers have suggested that gaze following does not require a representational component (Woodward, 2003).

In fact, several researchers have recently built computational models to explore the emergence and learning of gaze-following.

Previous models of gaze-following

One of the challenges confronting models of gaze-following is to create an embodied model. Embodiment is important in this domain for a number of reasons. First, there has recently been a movement for embodied models of cognition (e.g., Wilson, 2000). Second, spatial and developmental models seem to be particularly amenable to embodied cognition. Third, embodied cognition forces an integrative approach across models, theories, and empirical results. Finally, the complexity of the physical world provides strong tests for the theory under question. Each of the models of gaze following (including ours) claims they have embodied characteristics. There are three existing models of the acquisition of gaze-following.

Nagai, Hosoda, Morita, & Asada (2003) used a neural network approach to learn that shifts in the caregiver's head pose pointed to a salient and interesting object. Over time, the model (which also runs on a robot) learned to follow the gaze of the caregiver to an interesting object.

Doniec, Sun, & Scassellati (2006) greatly sped up the algorithm by using pointing gestures to acquire joint attention. Their algorithm (which also ran on a robot) had the robot actively point to the object it thought the caregiver was gazing at. This pointing greatly increased learning rate through positive examples. The fact that infants start to make deictic gestures around 10 months of age (Bates, Benigni, Bretherton, Camaioni, & Volterra, 1979), which is about the same age that gaze-following is acquired (Corkum & Moore, 1995; Corkum & Moore, 1998) provides empirical evidence that infant gesture may be a component of gaze-following. Beyond this interesting suggestion, however, Doniec et al.'s primary contribution is that it is able to learn at a much faster rate than previous models.

Triesch, Teuscher, Deak, & Carlson (2006) also developed a model of gaze-following. Triesch et al.'s model monitors the caregiver's direction of gaze and gradually learns that the caregiver looks at objects in the environment that are interesting or novel to the infant, which is rewarding. Triesch et al. modeled the learning process through Temporal-Difference (TD) learning, a biologically plausible reinforcement learning algorithm. Triesch et al.'s model used a model of habituation to determine when to shift attention and learned to follow gaze to determine where optimal (most interesting) objects were in the environment. Their model used a simple grid world where objects could only exist in a limited number of locations.

It is a mantra in the modeling community that no model is perfect; future models attempt to improve upon past models. All three of these models made strong progress toward the understanding of gaze-following. Their biggest weakness, however, is that they had significant issues with cognitive plausibility. In order to show cognitive plausibility, we (1) use and integrate a variety of cognitively plausible mechanisms (e.g., models of human memory, attention, etc.), (2) run models using a similar experimental paradigm, and (3) match experimental data using those mechanisms within the constraints of the experimental paradigm.

Several criticisms have been leveled against the Nagai et al. model. First, that model required an extremely large amount of training data; probably too much to be cognitively plausible (Doniec et al., 2006). Second, their model does not seem to be able to scale up to the more representational stage of gaze-following (Butterworth & Jarrett, 1991). Third, their model seems to work for only a single caregiver (Doniec et al., 2006).

Doniec et al.'s model was built in a manner that did not emphasize cognitive plausibility; their focus was on achieving fast and efficient learning for gaze-following in a realistic embodied context. One aspect of their model that limits its plausibility as a cognitive developmental account is the fact that they used six objects (toys) for joint gaze-following. If we assume that their model is approximately a 10 m. old infant, it is well known that infants at that age can not reliably identify objects a caregiver is gazing at if there are other objects in the line of sight (Butterworth & Jarrett, 1991).

While we agree with many aspects of Triesch et al.'s model, several criticisms have also been leveled at it. Some researchers have explicitly questioned the psychological plausibility (Moore, 2006). Specifically, Moore suggested that accurately modeling the attentional processes of infants during gaze following is a critical component to psychological plausibility in gaze-following. Additionally, because Triesch et al. used a grid system to simplify the training, the need for spatial cognition was greatly reduced. Thus, according to critics, a more robust and/or psychological representation of space was needed (Doniec et al., 2006; Moore, 2006).

The goal of this project is to show how an embodied model of gaze-following can not only perform gaze-following but also have a higher degree of cognitive plausibility by having cognitive attentional mechanisms (Doniec et al., 2006; Moore, 2006), a spatial representation (Doniec et al., 2006; Moore, 2006), and a match to data. While a match to data is not a perfect measure of cognitive plausibility (Cassimatis, Bello, & Langley, 2008), it can be used to differentiate models. At the least, if a model can show performance and competence as well as a reasonable data fit, it is more plausible (and, to us, preferred), than a model that does not.

The data we attempt to match is an experiment by Corkum and Moore (1998).

Method (Corkum & Moore, 1998)

A complete description of the experiment can be found in Corkum & Moore (1998).

Participants

63 participants completed the study, 21 participants in each of three age groups (6—7, 8—9, and 10—11 month olds).

Setup and Procedure

The experiment took place in a cubicle where two toys had been placed. Each toy rested on a turntable on either side of the room. When activated, the toy lit up and the turntable rotated. Both toys were visible to the infant at all times.

At the beginning of the experiment, each child entered into the cubicle and sat on their parent's lap directly across from the experimenter. The experimenter sat .6 m away. The experimenter called the child's name or tickled the child's tummy to get the infant to look at the experimenter. After the child looked at the experimenter, the trial began.

Each trial consisted of the experimenter looking 90° left or right at one of the two toys. The experimenter gazed at the toy for 7 s. During the trial, the experimenter did not vocalize or touch the infant, nor did the experimenter call the infant's name.

The experiment consisted of three consecutive phases. In the baseline phase, there were four trials where the experimenter looked at a toy (two trials to each side). During the baseline phase the toy remained inactive (i.e., did not light up or turn) in order to assess spontaneous gaze-following.

During the shaping phase, there were four trials (two to each side), but this time, regardless of the infant's gaze, the toy that was gazed at by the experimenter lit up and rotated.

During the final testing phase, a maximum of 20 trials (10 to each side) occurred where the toy was activated only if the infant and the experimenter looked at the same toy. If the child successfully followed the experimenter's gaze 5 times in a row, the experiment terminated.

Scoring

Each head turn was coded as either a target (joint-gaze with the experimenter) or a non-target (the wrong toy was gazed at) response. Infant head turns that did not look at a toy (e.g., naval-gazing) were not scored.

Random gaze-following would correspond to approximately 50% accuracy. Accurate gaze-following would correspond to an accuracy rate significantly greater than 50%, while anti-gaze-following would correspond to an accuracy rate significantly less than 50%.

Results and Discussion

To maintain clarity and connection with other researchers who report accuracy, percentage scores will be reported here for both the baseline and the last four test trials instead of the reported difference scores.

As Figure 1 suggests, only 10—11 m infants could reliably follow gaze at baseline. After training, however, both 8—9 m and 10—11 m infants could reliably follow gaze (there was a slight, non-significant increase in gaze-following for the 6—7 m infants).

These results are consistent with other researchers (Corkum & Moore, 1995) who have shown that gaze-following reliably occurs during the end of the first year: only 10—11 m infants could reliably follow gaze at baseline. Interestingly, however, 8—9 m infants learned to follow gaze in the experimental setting with a modest amount of training.

Corkum and Moore (1998) interpret these data as showing that there are several precursors to gaze-following. First, infants must be mature enough to respond to different spatial locations; they must have some rudimentary spatial ability. Second, infants must be able to learn that an interesting event will occur where the person looks. They further suggest that the adult's head turn cues the infant's attention in the direction of the turn.

We next describe the architecture and the task model.

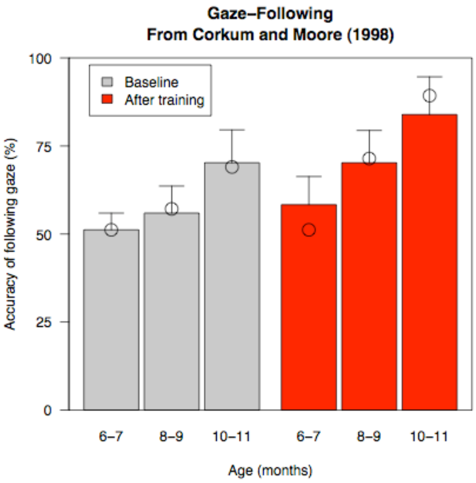


Figure 1: Experimental data from Corkum and Moore (1998). Bars are experimentall data and circles are model data. Error bars are 95% confidence intervals.

Architecture Description

ACT-R is a hybrid symbolic/sub-symbolic production-based system (Anderson, 2007). ACT-R consists of a number of modules, buffers, and a central pattern matcher. Modules contain a relatively specific cognitive faculty associated with a specific region of the brain. For each module, there are one or more buffers that communicate directly with that module as an interface to the rest of ACT-R. At any point in time, there may be at most one item in any individual buffer; thus, the module's job is to decide what and when to put a symbolic object into a buffer. The pattern matcher uses the contents of the buffer to match specific productions.

ACT-R supports the concept of purely bottom-up processing. Bottom-up or reactive processing occurs when there is no goal-directed processing that occurs. In contrast, top-down or goal-directed processing occurs when the goal buffer (intentional module) is part of the processing.

ACT-R interfaces with the outside world through the visual module, the aural module, the motor module, and the vocal module. Other current modules include the intentional, imaginal, temporal and declarative modules.

We have modified ACT-R by allowing it to perceive the physical world by attaching robotic sensors and effectors to it; we call our system ACT-R/E (the "E" is for Embodied). For ACT-R/E, we have added a new module (spatial) and modified the visual, aural and motor modules to work with our robot and to use real-world sensor modalities. We did not modify other parts of the architecture itself. Below we discuss the modifications to visual and motor (aural is not used in this project) and a brief description of the spatial module. Figure 2 shows a schematic of ACT-R/E.

Visual

The Visual Module is used to provide a model with information about what can be seen in the current environment. ACT-R normally sees information presented on a computer monitor. We modified the original visual module to accept input from a video camera. The visual module allows access to both the location of an object (the "where'" system) and a more detailed representation (the "what" system). Obtaining additional information about an object or person requires declarative retrieval(s). We used a 3D optical flow model to capture a person's 3D head pose in space and a fiducial tracker for object identification and localization. These systems are described more fully elsewhere (Kato, Billinghurst, Poupyrev, Imamoto, & Tachibana, 2000; Trafton, Bugajska, Fransen, & Ratwani, 2008; Fransen, Hebst, Harrison, & Trafton, under review).

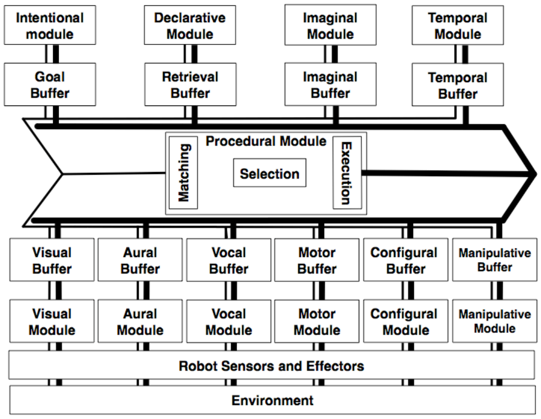


Figure 2: Schematic of ACT-R/E

Motor

Traditional ACT-R has a virtual motor system that allows virtual hand movements (e.g., typing, mouse movements). ACT-R/E's motor module allows commands to be issued for navigation and mobility, as well as providing self-localization knowledge. In this project, motor is used to control the robot's head, including the eyes and head pose.

Spatial

To facilitate acting in space, ACT-R/E utilizes a spatial theory called Specialized Egocentrically Coordinated Spaces (SECS, pronounced seeks) (Harrison & Schunn, 2003). SECS is neurologically inspired and based on 3D space (Previc, 1998). SECS provides two egocentric spatial modules, which are responsible for the encoding and transformation of representations in service of navigation (configural) and manipulation (manipulative).

The configural module provides high fidelity location information for attended representations that is automatically updated as the model moves through or looks around the environment. The configural module represents the world as spatial blobs that need to be navigated around, above, or below. These spatial blobs do not have a high degree of precision. The manipulative module uses a metric, geon-based 3D representation for objects. The manipulative module provides encodings of object geometry and orientation, a critical component to the gaze-following discussed below.

Simulator and Robot Description

Currently, the open-source Stage robot simulator (Collett, MacDonald, & Gerkey, 2005) is used to enable data collection and to speed-up the model development cycle.

Our current robot platform is the MDS (Mobile-Dexterous-Social) Robot (Breazeal, 2009). The MDS robot neck has 18 DoF for the neck and head including eye pitch and pan which allows the robot to look at various locations in 3D space. Perceptual inputs include a color video camera and a SR3000 camera to provide depth information. For the current project, the MDS head can move its eyes and head to look at various locations in 3D space.

Model Description

An ACT-R/E model was developed that simulates the development of gaze-following.

High Level Description of the gaze-following model

There are five model components that enable gaze-following: the reactive nature of the model; using ACT-R's memory system as a model of habituation; a more detailed description of the spatial components; the gaze-following itself; and the utility learning mechanism.

The reactive nature of the model The model itself is completely bottom-up; there is no goal-directed or top-down action in this model. The model was written in this manner because early gaze-following seems to be emergent rather than goal-directed (Triesch et al., 2006). Later models in the developmental process will need to have a goal-directed component.

Habituation in ACT-R When the model gazes at any object (person, toy, etc.), it looks at that object until it can recall the object before it attempts to look at a different object. This is an approximation of habituation (Sirois & Mareschal, 2002); several other researchers (Triesch et al., 2006) use an exponential function that is remarkably similar and formally equivalent to ACT-R's model of memory retrieval (Anderson, Bothell, Lebiere, & Matessa, 1998).

After the model gazes at and habituates to an object, it starts to look for a new object.

Spatial Module As mentioned earlier, standard ACT-R has only a rudimentary spatial ability. This ability is part of the visual module. In the visual module, a visual description of the object (a "what" component) and where that object is located in screen coordinates (a "where" component) is available (Byrne & Anderson, 1998). ACT-R's what and where system are used any time visual objects in the world need to be attended to. Many successful models of attention have been built using these mechanisms.

Unfortunately, the what and where components of ACT-R are not sufficient to follow gaze, much less provide even rudimentary spatial competency. As previously mentioned, two spatial modules were added to ACT-R, the configural module and the manipulative module.

The configural module is focused on the configuration of objects in the world relative to self. Specifically, it allows the model to determine how far away from self another object is and what angle that object is from self. Configural information changes dynamically as objects in the world change or move (including the self-model). This information is critical for navigation in general and spatial cognition in an embodied context.

For gaze-following, the manipulative buffer provides the orientation that a particular object is facing. Specifically, the manipulative buffer provides information about what direction a person is facing (body) or gazing (head).

The visual, configural, and manipulative modules are linked symbolically so that different types of spatial information about an object can be easily kept track of.

Gaze Following Gaze-following was implemented by adding constraints to the visual search mechanism. As implemented, gaze-following is a directed visual search along a retinotopic vector. Given a starting point and either an angle or an end point, the visual search will return the location on an object somewhere along that line within some tolerance. Note that this mechanism works in 3D space.

This simple mechanism allows the visual system to find candidate objects along a gaze, or any potential obstructions. These skills align nicely with Butterworth's developmental stages of gaze (Butterworth & Jarrett, 1991).

Utility Learning ACT-R is able to not only learn new facts and rules, but also to learn which rule should fire (called utility learning in ACT-R). It accomplishes this by learning which rule or set of rules lead to the highest reward. ACT-R uses an elaboration of the Rescorla-Wagner learning rule and the temporal-difference (TD) algorithm. The TD algorithm has been shown to be related to animal and human learning theory. The elaboration in ACT-R is more applicable for human learning and allows it to be more easily incorporated into a production-system framework (Fu & Anderson, 2006).

Briefly, any time a reward is given (e.g., for infants, a smile from a caregiver), a reward is propagated back in time through the rules that had an impact on the model getting that reward. Punishments may also be given with a similar time-course, but no punishments were given in this model.

For all models, we kept most of the ACT-R parameter defaults. The parameters that were changed include the base level learning (a decay value of .2 instead of the typical default of .5), which allowed for a reasonable habituation timecourse; utility noise (set at a reasonable .5) to allow low-use productions to occasionally fire; and the utility learning rate (set at .2) which allowed the productions to converge to a stable expected utility within a reasonable period of time (minutes instead of months).

A sample experimental model run

The first thing that the model does in an experimental trial is to find a person (called a caregiver in this example). This corresponds to the experimental procedure where the experimenter got the infant's attention (Corkum & Moore, 1998). The model looks at the caregiver until it has habituated to that person, as described above. The caregiver looks at an object in the environment for 7 s or until the model makes a decision about where to look.

When the model is "young" it has a favored rule set, which is to locate, attend-to, and gaze at an object. The object can be anything in the model's field of view and it is chosen randomly.

If the caregiver is looking at the same object that the model decides to look at, the model is given a small reward. If the caregiver is looking at a different object than the model, no reward is given but the trial is completed and the reward process begins anew.

Even though there is a favored rule to find an object and gaze at it, the gaze-following rule competes with it. The gaze-following rule has a much lower utility when the model is young so it does not get an opportunity to fire very often. However, because of the relatively high noise value for utility (called expected-utility-noise in ACT-R), the gaze-following rule does occasionally get a chance to fire. If the gaze-following rule has a high enough utility to fire, it attempts to follow the gaze of the caregiver to an object.

The gaze-following production uses configural knowledge to determine the caregiver's distance and orientation from itself. As long as the model attends to the caregiver, the current information is available to the model.

The gaze-following production also uses manipulative knowledge of the head of the caregiver to determine what direction the caregiver's head is facing. This information is clearly important because without it the gaze of the caregiver could not be determined. Note also that the model assumes that the eyes are facing the same direction as the head. For the experimental procedure discussed here, this assumption is appropriate, but as children develop (by 1 year) they do differentiate between head pose and where the eyes themselves are gazing (Brooks & Meltzoff, 2002).

With this information, the infant model looks from the caregiver in the direction the head is facing. The model then finds the first available object in that direction, which is consistent with previous research (Butterworth & Jarrett, 1991). The model is again given a small reward. After habituation to that object, the trial ends and the model looks for another object to attend to.

Because the gaze-following production is correct more often than the random production (which is accurate on average 1/(number-of-objects), the gaze-following production slowly gains utility. However, it takes a period of time before the combination of noise and utility allow the gaze-following production to overtake and eventually become dominant over the random-object production.

Modeling developmental progress

When the model is young, it has a handful of productions that look around the world. Experience is simulated by concentrating gaze-following learning such that a few minutes is equal to 2 months. For the 6-7 m model, it was given 80 seconds of experience with looking around a simple world at objects and receiving feedback as described in the experimental run. For the 8-9 m model, three minutes of experience were given, and for the 10-11 six model, six minutes of experience were given. Because the rate of learning is dependent entirely on the utility learning rate parameter, learning occurred quite quickly in this model. Utility learning rate could be scaled down substantially to match actual infant learning time. In order to do this correctly, however, it would be important to know approximately how many times an infant attempts to follow a gaze or how often an infant receives feedback or the infant found something especially interesting to look at as well as knowledge about the environment (e.g., the number of objects). Other researchers have come to a similar conclusion concerning the importance of learning in gaze-following (Corkum & Moore, 1998; Triesch et al., 2006).

At each age (6-7, 8-9, and 10-11 m), the model was put through the exact same experimental procedure as Corkum & Moore (1998). Note that the lighting up and rotating of the toy provided a strong reward to the child, which is modeled by joint attention during the training phase of the procedure; no reward was given during the baseline phase, so this was a relatively pure measure of age-related ability.

To provide some match to the experimental procedure, 21 models (corresponding to the 21 participants) were run at each age group. However, to achieve stable results, the model was run 10 times with no utility learning for the baseline and after training conditions. This allowed the model to be tested after different age or experimental related amounts of practice yet maintain stable results.

Model fit

As is evident in Figure 1, the model matches the data quite well; $R^2 = .95$ and RMSD = .3. Critically, all model points are within 95% confidence intervals of the data. The model suggests that there is not a qualitative change in any child, but that as children gain more experience they get better at it. Interestingly, with a modest amount of experimental training, the 8-9 m model also showed improvement (though not, of course, as much as the 10-11 m model). Again the model suggests that the reason for this is that 8-9 m children were at the "right" developmental age to take advantage of the concentrated training. This training allowed productions that occasionally fired during "real life" to be focused and rewarded, which brought their utility to surpass the random behavior they had before the experiment started. Note again that the 6-7 m children did not statistically improve. The model explanation for this is that they simply had not had enough experience yet.

Embodied gaze following

The infant model at each stage of development was trained using Player and then run on an embodied platform (our robot). Movies are available at http://www.nrl.navy.mil/aic/iss/aas/CognitiveRobotsVideos.php.

General Discussion

We described an embodied model of gaze-following that is not only functional but matches data from a classic gaze-following paradigm and experiment. The primary advantage of this model over previous models is that it has a very high degree of cognitive plausibility. First, as Moore (2006) suggested, it has an accepted model of visual attention. Second, it has a psychologically plausible representation of space that is critical to the success of the model. Third, this model is embodied and runs on a physical robot, allowing additional tests of the theory as well as added complexity.

Of the model's 5 components (reactivity, habituation, the spatial module, gaze-following, and utility learning), three of them are absolutely critical to the success of the model. The reactivity nature of the module is a theoretical commitment to modeling young children, though the model could be written using a top-down model. Likewise, habituation is something that has been theoretically proposed and empirically observed, though it is not a critical component to the success of the model. The other three components, however, are needed. The spatial component integrates the spatial aspects of the task while the entire system could not function without the ability to perceive which direction a person is gazing. Because the developmental progress is accounted for by utility learning, it also is a necessary part of the model.

The model does make an interesting prediction: that 6 m infants (and even younger) could learn to follow gaze with enough practice. A core component to this prediction is that the infant have enough patience to go through enough training and the ability of young children to extract 3D information from the world. It is believed that 6 m olds do have this capability, but very young children do develop it.

This model also has several similarities to other infant data. The model does not understand obstructions and follows gaze to the first object along a path (Butterworth & Jarrett, 1991). The architecture does have the capability, however, to perform relatively precise gaze-following, ignoring highly salient objects in the path (the 'geometric' stage; Butterworth & Jarrett, 1991). The current model can not, however, follow gaze to a position outside its current field of view (the 'representational' stage). The current model has no true perspective-taking ability at all.

In order to provide the model with perspective taking abilities, it would presumably need more goal-directed cognition as well as more developed spatial capabilities.

Acknowledgments

This work was supported by the Office of Naval Research under funding document N0001409WX20173 to JGT.

References

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*, 341-380.

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press, USA.

Baron-Cohen, S. (1995). The eye direction detector (EDD) and the shared attention mechanism (SAM): Two cases for. *Joint attention: Its origins and role in development*.

Bates, E., Benigni, L., Bretherton, I., Camaioni, L., & Volterra, V. (1979). *The emergence of symbols: Cognition and communication in infancy*. New York, NY: Academic Press.

Breazeal, C. (2009). MDS Robot. http://robotic.media.mit.edu/projects/robots/mds/overview/overview.html.

Brooks, R., & Meltzoff, A. N. (2002). The Importance of Eyes: How Infants Interpret Adult Looking Behavior. *Developmental psychology*.

Butterworth, G., & Jarrett, N. (1991). What minds have in common is space: Spatial mechanisms serving joint visual attention in infancy. *British Journal of Developmental Psychology*.

Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson, & C. Lebiere (Eds.), *Atomic Components of thought* (pp. 167-200). Mahwah, NJ:: Lawrence Erlbaum.

Cassimatis, N. L., Bello, P., & Langley, P. (2008). Ability, Breadth and Parsmiony in Computational Models of Higher-Order Cognition. *Cognitive Science*, *32*(8), 1304-1322.

Collett, T. H., MacDonald, B. A., & Gerkey, B. P. (2005). Player 2.0: Toward a Practical Robot Programming Framework. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005)*.

Corkum, V., & Moore, C. (1998). The origins of joint visual attention in infants. *Developmental Psychology*, *34*(1), 28-38.

Corkum, V., & Moore, C. (1995). Development of joint visual attention in infants. In V. Corkum, & C. Moore (Eds.), *Joint attention: Its origins and role in development* (pp. 61–83).

Doniec, M. W., Sun, G., & Scassellati, B. (2006). Active Learning of Joint Attention. *IEEE-RAS International Conference on Humanoid Robotics*

Fransen, B., Hebst, E., Harrison, A., & Trafton, J. G. (under review). 3D position and pose tracking. In.

Fu, W., & Anderson, J. (2006). From recurrent choice to skill learning: A model of reinforcement learning. *Journal of Experimental Psychology: General*.

Harrison, A. M., & Schunn, C. D. (2003). ACT-R/S: Look Ma, No "Cognitive-map"! In *Int'nl Conference on Cognitive Modeling*.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., & Tachibana, K. (2000). Virtual object manipulation on a table-top AR environment. In *IEEE and ACM International Symposium on Augmented Reality* (pp. 111-119).

Moore, C. (2006). Modeling the development of gaze following needs attention to space. *Developmental Science*.

Nagai, Y., Hosoda, K., Morita, A., & Asada, M. (2003). A constructive model for the development of joint attention. *Connection Science*.

Previc, F. H. (1998). The neuropsychology of 3-D space. *Psychological Bulletin*, *124*(2), 123-164.

Scaife, M., & Bruner, J. S. (1975). The capacity for joint visual attention in the infant. *Nature*.

Sirois, S., & Mareschal, D. (2002). Models of habituation in infancy. *Trends in Cognitive Sciences*, *6*(7), 293-298.

Trafton, J. G., Bugajska, M. D., Fransen, B. R., & Ratwani, R. M. (2008). Integrating vision and audition within a cognitive architecture to track conversations. *Proceedings of the Human Robot Interaction 2008 (HRI 2008)*.

Triesch, J., Teuscher, C., Deak, G. O., & Carlson, E. (2006). Gaze following: why (not) learn it? *Develop. Science*, *9*, 125-147.

Woodward, A. L. (2003). Infants'developing understanding of the link between looker and object. *Developmental Science*.

Human and Optimal Exploration and Exploitation in Bandit Problems

Shunan Zhang (szhang@uci.edu)
Michael D. Lee (mdlee@uci.edu)
Miles Munro (mmunro@uci.edu)
Department of Cognitive Sciences, 3151 Social Sciences Plaza A
University of California, Irvine, CA 92697-5100 USA

Abstract

We consider a class of bandit problems in which a decision-maker must choose between a set of alternatives—each of which has a fixed but unknown rate of reward—to maximize their total number of rewards over a short sequence of trials. Solving these problems requires balancing the need to search for highly-rewarding alternatives with the need to capitalize on those alternatives already known to be reasonably good. Consistent with this motivation, we develop a new model that relies on switching between latent *searching* and *standing* states. We test the model over a range of two-alternative bandit problems, varying the number of trials, and the distribution of reward rates. By making inferences about the latent states from optimal decision-making behavior, we characterize how people should switch between searching and standing. By making inferences from human data, we attempt to characterize how people actually do switch. We discuss the implications of our findings for understanding and measuring the competing demands of exploration and exploitation in decision-making.

Keywords: Bandit problems, exploration versus exploitation, reinforcement learning, Bayesian graphical models, human decision-making, optimal decision-making

Bandit Problems

Bandit problems, originally described by Robbins (1952), present a simple challenge to a decision-maker. They must choose between a known set of alternatives on each of a series of trials. They are told each of the alternatives has a fixed reward rate, but are not told what the rates are. Their goal is just to maximize the total reward they receive over the series of trials. In this paper, we focus on short finite-horizon versions of the bandit problem, involving just a small number of trials.

As an example of the challenge posed by these sorts of bandit problems, consider the situation shown in Figure 1. Here there are two alternatives, and 16 total trials available to attain rewards. After 10 trials, one alternative has been chosen 8 times, and returned 3 successes and 5 failures, while the other alternative has been tried just 2 times, for 1 success and 1 failure. Which alternative should be chosen on the 11th trial? Choosing the first alternative exploits the knowledge that it quite likely returns rewards at a moderate rate. Choosing the second alternative explores the possibility that this alternative may be the more rewarding one, even though much less is known about it.

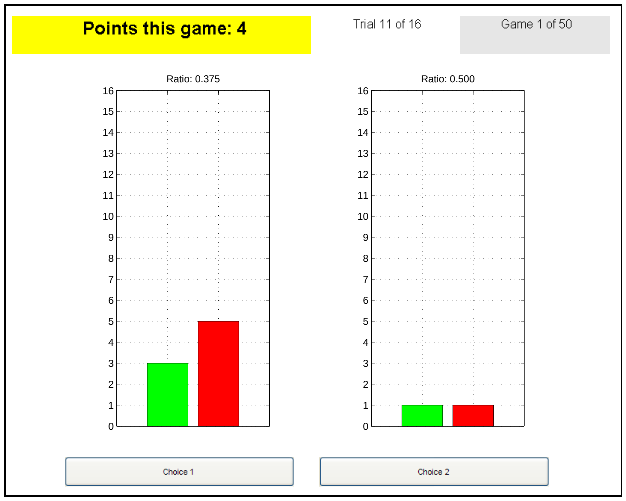


Figure 1: An example bandit problem, with two alternatives and 16 total trials. After 10 trials, the first alternative on the left has 2 successes (lighter, green bar) and 5 failures (darker, red bar), while the alternative on the right has 1 success and 1 failure.

As this example makes clear, finite-horizon bandit problems are psychologically interesting because they capture the tension between exploration and exploitation evident in many real-world decision-making situations. Decision-makers must try to learn about the alternatives, which requires exploration, while simultaneously satisfying their goal of attaining rewards, which requires exploitation. In this way, studying human performance on bandit problems addresses basic questions, including how people search for information, how they adapt to the information they find, and how they optimize their behavior to achieve their goals.

Human performance on bandit problems has been studied from a variety of psychological perspectives. Early studies used models and experimental manipulations motivated by theories of operant conditioning (e.g., Brand, Wood, & Sakoda, 1956); later studies were informed by economic theories with a focus on deviations from rationality in human decision-making (e.g., Banks, Olson, & Porter, 1997; Meyer & Shi, 1995); most recently human performance on the bandit problem has been a topic of interest in cognitive neuroscience (e.g., Cohen, McClure, & Yu, 2007; Daw, O'Doherty, Dayan, Seymour, & Dolan, 2006) and probabilistic models of human cognition (e.g., Steyvers, Lee, & Wagenmakers, in press).

One common finding is that people often switch flexibly between exploration and exploitation, often choosing alternatives in proportion to their reward rate, unless they are given strong incentives to maximize their reward by repeatedly choosing the most-rewarding alternative (e.g., Shanks, Tunney, & McCarthy, 2002). Typically, these experiments involve a large number of trials, and so one plausible explanation for sub-optimal probability matching is that people are allowing for the possibility that rewards rates might change over time. This seems less likely to be a confounding consideration in short-horizon bandit problems, and so we are especially interested to know if people switch between exploration and exploitation for these problems.

Accordingly, in this paper we develop and evaluate a probabilistic model that assumes different latent states guide decision-making for short-horizon bandit problems. These latent states give emphasis either to searching the environment, or to choosing the same alternative repeatedly, and so dictate how a decision-maker solves the dilemma in our introductory example, where a well-understood but only moderately-rewarding alternative must be compared to a less well-understood but possibly better-rewarding alternative. Using the optimal decision process, and human data, for a range of bandit problems we apply our model to understand the best way to switch between searching and standing, and how people actually do switch, for short horizon two-alternative bandit problems.

The outline of the paper is as follows. In the next section, we present our model, including its implementation as a probabilistic graphical model. We then report an experiment collecting human and optimal decisions for a range of bandit problems. Next, we use the behavioral data and our model to make inferences about the optimal way to switch between searching and standing, and how people actually do switch. Finally, we draw some conclusions relating to simpler latent state models suggested by our analysis.

A Latent State Model

Bandit problems have been widely studied in the fields of game theory and reinforcement learning (e.g., Berry, 1972; Berry & Fristedt, 1985; Gittins, 1979; Kaebling, Littman, & Moore, 1996; Macready & Wolpert, 1998; Sutton & Barto, 1988). One interesting idea coming from established reinforcement learning models is that of a latent state to control exploration versus exploitation behavior.

In particular, the 'ε-first' heuristic (Sutton & Barto, 1988) assumes two distinct stages in bandit problem decision-making. In trials in the first 'exploration' stage, alternatives are chosen at random. In the second 'exploitation' stage, the alternative with the best observed ratio of successes to failures from the first stage is chosen. The demarcation between these stages is determined by a free parameter, which corresponds to the trial at which exploration stops and exploitation starts.

Our Model

Our model preserves the basic idea of a latent exploration or exploitation state guiding decision-making, but makes two substantial changes. First, we allow each trial to have a latent state, introducing the possibility of switching flexibly between exploration and exploitation to solve bandit problems. In our model, for example, it is possible to begin by exploring, then exploit, and then return for an additional period of exploration before finishing by exploiting. Indeed, any pattern of exploration and exploitation, changing trial-by-trial if appropriate, is possible.

Second, we implement exploration and exploitation behavior using a more subtle mechanism than just random search followed by deterministic responding. In particular, for the two-alternative bandit problems we consider, our model distinguishes between three different situations,

- The *Same* situation, where both alternatives have the same number of observed successes and failures.
- The *Better-Worse* situation, where one alternative has more successes and fewer failures than the other alternative (or more successes and equal failures, or equal successes and fewer failures). In this situation, one alternative is clearly better than the other.
- The *Search-Stand* situation, where one alternative has been chosen much more often, and has more successes but also more failures than the other alternative. In this situation, neither alternative is clearly better, and the decision-maker faces a dilemma. Choosing the better-understood alternative corresponds to standing; choosing the less well-understood alternative corresponds to searching.[1]

Within our model, which alternative is chosen depends on the situation, as well as the latent search or stand state. For the *same* situation, both alternatives have an equal probability of being chosen. For the *better-worse* situation, the better alternative has a high probability, given by a parameter γ, of being chosen. The probability the worse alternative is chosen is $1-\gamma$.

[1] Intuitively, our notion of searching is a form of exploration, and our notion of standing is a form of exploitation. We use the new terms, however, to emphasize that our search and stand decisions have formal characterizations that are different definitions of exploration and exploitation in reinforcement learning algorithms. For example, ε-first uses simple random choices as a model of exploration, whereas our approach is based on choosing specifically the alternative that is less well known in a search-stand situation.

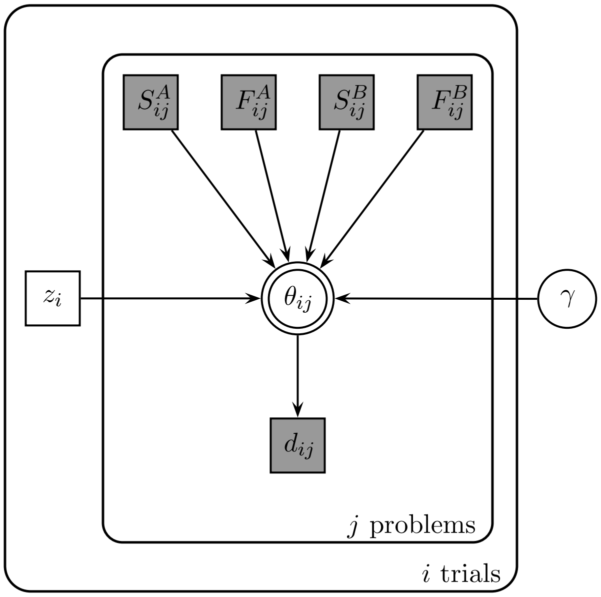


Figure 2: Graphical representation of the latent state model.

For the *search-stand* situation, the exploration alternative will be chosen with the high probability γ if the decision-maker is in a latent search state, but the exploitation alternative will be chosen with probability γ if the decision-maker is in the latent stand state. In this way, the latent state for a trial controls how decisions are made each time the decision-maker encounters a search-stand situation.

Graphical Model Implementation

We implemented our model as a probabilistic graphical model in WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000), which makes it easy to do fully Bayesian inference using computational methods based on posterior sampling. The graphical model is shown in Figure 2, using the same notation as Lee (2008).

The encompassing plates show the repetitions for the trials within each problem, and the multiple problems completed by a decision-maker. The square shaded nodes S_{ij}^A, S_{ij}^B , F_{ij}^A and F_{ij}^B are the observed counts of successes and failures for alternatives A and B on the ith trial of the jth problem. The unshaded node γ is the 'accuracy of execution' parameter, controlling the (high) probability that the deterministic heuristic described by our model is followed. The unshaded z_i nodes are the discrete latent indicator variables, with $z_i = 0$ meaning the ith trial is in the explore state, and $z_i = 1$ meaning it is in the exploit state. We assumed uninformative priors $\gamma \sim \text{Uniform}(0,1)$ and $z_i \sim \text{Bernoulli}(1/2)$.

The double-bordered θ_{ij} node is a deterministic function of the S_{ij}^A, S_{ij}^B , F_{ij}^A, F_{ij}^B, γ and z_i variables. It gives the probability that alternative A will be chosen on the ith trial of the jth problem. According to our model, this is

$$\theta_{ij} = \begin{cases} 1/2 & \text{if A is same} \\ \gamma & \text{if A is better} \\ 1-\gamma & \text{if A is worse} \\ \gamma & \text{if A is search and } z_i = 0 \\ 1-\gamma & \text{if A is search and } z_i = 1 \\ \gamma & \text{if A is stand and } z_i = 1 \\ 1-\gamma & \text{if A is stand and } z_i = 0. \end{cases}$$

The shaded d_{ij} node is the observed decision made, $d_{ij} = 1$ if alternative A is chosen and $d_{ij} = 0$ if alternative B is chosen, so that $d_{ij} \sim \text{Bernoulli}(\theta_{ij})$.

In this way, the graphical model in Figure 2 provides a probabilistic generative account of observed decision behavior. It is, therefore, easy to use the model to make inferences about latent search and stand states from decision data. In particular, the posterior distribution of the z_i variable represents the probability that a decision-maker has a latent search versus stand state on the ith trial. In the next section, we describe an experiment that provides both human and optimal data suitable for this type of analysis.

Experiment

Participants

We collected data from 10 naive participants (6 males, 4 females).

Stimuli

We considered six different types of bandit problems, all involving just two alternatives. The six bandit problem types varied in terms of two trial sizes (8 trials and 16 trials) and three different environmental distributions ('plentiful', 'neutral' and 'scarce') from which reward rates for the two alternatives were drawn.

Following Steyvers et al. (in press), we defined these environments in terms of Beta (α, β) distributions, where α corresponds to a count of 'prior successes' and β to a count of 'prior failures'. The three environmental distributions are shown in Figure 3, and use values $\alpha = 4$, $\beta = 2$, $\alpha = \beta = 1$, and $\alpha = 2$, $\beta = 4$, respectively.

Procedure

We collected within-participant data on 50 problems for all six bandit problem conditions, using a slight variant of the experimental interface shown in Figure 1. The order of the conditions, and of the problems within the conditions, was randomized for each participant. All $6 \times 50 = 300$ problems (plus 5 practice problems per condition) were completed in a single experimental session, with breaks taken between conditions.

Optimal Performance

Given the α and β parameters of the environmental distribution, and the trial size, it is possible to find the optimal decision-making process for a bandit problem. This is achieved via dynamic programming, using a recursive approach well understood in the reinforcement learning

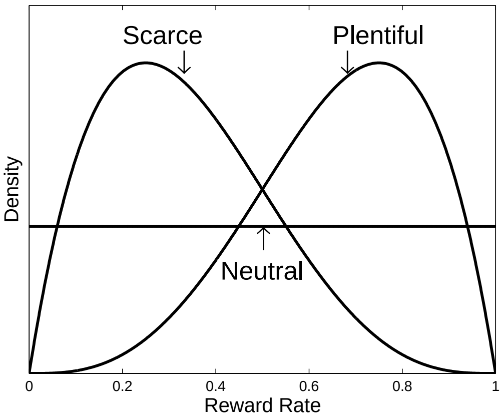


Figure 3: The plentiful, neutral and scarce environmental distributions of reward rates.

Table 1: Posterior predictive agreement between the latent state model, and the optimal and human decision-makers (DMs), for the three environments and two problem sizes.

	Plentiful		Neutral		Scarce	
DM	8	16	8	16	8	16
Optimal	.95	.93	.95	.94	.92	.90
PH	.96	.94	.92	.92	.84	.90
ST	.99	.87	.94	.84	.93	.80
AH	.89	.89	.76	.75	.71	.73
MM	.92	.88	.92	.93	.90	.94
SZ	.92	.94	.95	.92	.88	.91
MY	.94	.95	.92	.93	.89	.88
EG	.94	.91	.90	.90	.85	.89
MZ	.97	.91	.92	.88	.93	.86
RW	.89	.90	.86	.80	.84	.80
BM	.93	.88	.92	.87	.89	.90

literature (e.g., Kaebling et al., 1996). Using this approach, we calculated optimal decision-making behavior for all of the problems completed by our participants.

Modeling Analysis

We applied the graphical model in Figure 2 to the optimal and human decision data, for all six bandit problem conditions. For each data set, we recorded 1,000 posterior samples from the joint distribution of the unobserved variables. We used a burn-in also of 1,000 samples, and multiple independent chains, to assess convergence.

Basic Results

Descriptive Adequacy A basic requirement of any cognitive model is that it can fit the observed data reasonably well. To test the descriptive adequacy of the latent state model, we used a standard Bayesian approach and evaluated its posterior predictive fit to the to all of the human and optimal decision-making data (i.e., the agreement between the model and data averaged over the posterior distribution of the parameters). The levels of agreement are shown in Table 1. It is clear that the latent state model is generally able to fit both human and optimal behavior very well. There are some small suggestive differences—scarce environments seem, for example, to be a little less well described, as does one participant (AH)—that are worthy of future investigation, but do not affect our broad analyses in this paper.

Latent States Having checked the descriptive adequacy of the latent state model, our main interest is in the change between latent search and stand states, as shown by the inferred model parameters.[2] The basic results needed to address this question are summarized by the posterior mean of the z_i indicator variables, which approximates the posterior probability that the ith trial uses the stand state.

Figure 4 shows the posterior means of the z_i variables for the optimal decision process, and all 10 participants, in all six experimental conditions. The experimental conditions are organized into the panels, with rows corresponding the plentiful, neutral and scarce environments, and the columns corresponding to the 8- and 16-trial problems. Each bar graph shows the probability of an stand state for each trial, beginning at the third trial (since it is not possible to encounter the search-stand situation until at least two choices have been made). The larger bar graph, with black bars, in each panel is for the optimal decision-making data. The 10 smaller bar graphs, with gray bars, corresponds to the 10 participants within that condition.

Analysis

The most striking feature of the pattern of results in Figure 4 is that, to a good approximation, once the optimal or human decision-maker first switches from searching to standing, they do not switch back. This is remarkable, given the completely unconstrained nature of the model in terms of search and stand states. All possible sequences of these states over trials are given equal prior probability, and all could be inferred if the decision data warranted.

The fact that both optimal and human data lead to a highly constrained pattern of searching and standing states across trials reveals an important regularity in bandit problem decision-making. We consider this finding first in terms of optimal decision-making, and then in terms of human decision-making.

Optimal Decision-Making The optimal decision process results in Figure 4 show that it is optimal to be-

[2] We observed that the inferred γ parameter values were all close to 1, as expected, and do not report them in detail.

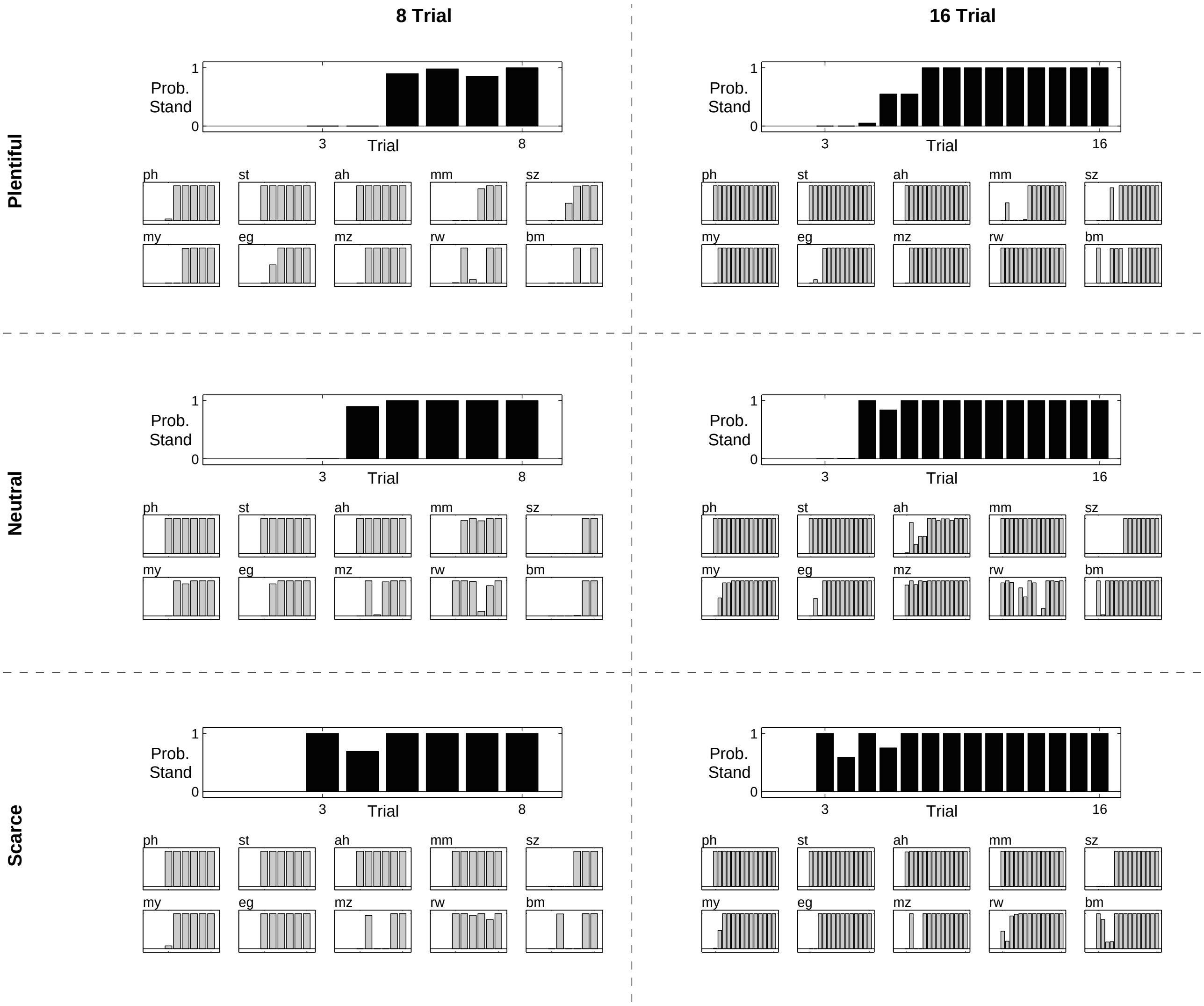


Figure 4: Each bar graph shows the inferred probabilities of the stand state over the trials in a bandit problem. Each of the six panels corresponds to an experimental condition, varying in terms of the plentiful, neutral or scarce environment, or the use of 8 or 16 trials. Within each panel, the large black bar graph shows the stand probability for the optimal decision-process, while the 10 smaller gray bar graphs correspond to the 10 participants.

gin with searching, then transition (generally) abruptly to standing at some trial that depends on the nature of the environment, and remain in the stand state for all of the remaining trials. The plentiful and scarce environments for 16-trial problems show a few trials where there is uncertainty as to whether searching or standing is optimal but, otherwise, it seems clear that optimal decision-making can be characterized by a single transition from searching to standing.

It is also clear from Figure 4 that the optimal decision-making must be sensitive to the environment in switching from searching to standing. In particular, as environments have lower expected reward rates, the switch away from searching begins earlier in the trial sequence. For example, the optimal decision process for 8-trial problems essentially switches from searching to standing at the 5th trial in the plentiful environment, but at the 4th trial in the neutral environment, and the 3rd trial in the scarce environment.

Human Decision-Making While the regularity in switching might not be surprising for optimal decision-making, it is more remarkable that human participants show the same pattern. There are some exceptions—both participants RW and BM, for example, sometimes switch from standing back to searching briefly, before returning to standing—but, overall, there is remarkable consistency. Most participants, in most conditions, begin by searching, and transition at a single trial to standing, which they maintain for all of the subsequent trials.

However, while there is consistency over the participants in switching just once from searching to standing, there are clear differences between individuals in when that switch happens. For example, the participant SZ, in all of the conditions, switches at a much later trial than most of the other participants.

There also seem to be individual differences in terms of sensitivity to the environment. Some participants switch at different trials for different environments, while others—such as participant ST—switch at essentially the same trial in all experimental conditions.

Discussion

Our basic findings involve both a regularity and a flexibility in the way people (and optimal) decision-makers switch between exploration and exploitation in bandit problems. The regularity is that a beginning period of searching gives way to a sustained period of standing. The flexibility is that when this switch occurs depends on the individual decision-maker, the statistical properties of the reward environment, and perhaps the interaction between these two factors.

The obvious cognitive model suggested by our findings combines the regularity with the flexibility. We propose that decision-making on finite-horizon bandit problem can be modeled in terms of a single parameter, controlling when searching switches to standing. That is, rather than needing a latent state parameter for each trial, only a single switch-point parameter is needed, with all earlier trials following the searching state, and all later trials following the standing state. Such a model would be similar in spirit—but formally different in an important way—to the standard ε-first heuristic from reinforcement learning. It would combine the single switch-point with an analysis of bandit game situations (‘same’, ‘better-worse’, ‘search-stand’) that produces more focused and principled operational definitions of what it means for decision-maker to explore and exploit.

A priority for future research is to apply this new single-switch model to human and optimal behavior on bandit problems. Being able to make inferences about when people and optimal decision-makers switch from exploration to exploitation promises a direct way to assess individual differences in how people search their environment for information, and react to different distributions of reward in those environments.

Acknowledgments

This work is was supported by the Air Force Office of Scientific Research (FA9550-07-1-0082) to Michael Lee and Mark Steyvers. We thank Jun Zhang for suggesting we evaluate the type of model considered in this paper.

References

Banks, J., Olson, M., & Porter, D. (1997). An experimental analysis of the bandit problem. *Economic Theory*, *10*, 55–77.

Berry, D. A. (1972). A Bernoulli two-armed bandit. *The Annals of Mathematical Statistics* , *43*(3), 871–897.

Berry, D. A., & Fristedt, B. (1985). *Bandit problems: Sequential allocation of experiments* . London: Chapman & Hall.

Brand, H., Wood, P. J., & Sakoda, J. M. (1956). Anticipation of reward as a function of partial reinforcement. *Journal of Experimental Psychology* , *52*(1), 18–22.

Cohen, J. D., McClure, S. M., & Yu, A. J. (2007). Should I stay or should I go? exploration versus exploitation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *362*, 933–942.

Daw, N. D., O’Doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, *441*, 876–879.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, *41*, 148–177.

Kaebling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* , *4*, 237–285.

Lee, M. D. (2008). Three case studies in the Bayesian analysis of cognitive models. *Psychonomic Bulletin & Review*, *15*(1), 1–15.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS: A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325–337.

Macready, W. G., & Wolpert, D. H. (1998). Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on evolutionary computation*, *2*(1), 2-22.

Meyer, R. J., & Shi, Y. (1995). Sequential choice under ambuigity: Intuitive solutions to the armed-bandit problem. *Management Science*, *41*(5), 817–834.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, *55*, 527–535.

Shanks, D. R., Tunney, R. J., & McCarthy, J. D. (2002). A re-examination of probability matching and rational choice. *Journal of Behavioral Decision Making*, *15*, 233–250.

Steyvers, M., Lee, M. D., & Wagenmakers, E.-J. (in press). A Bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*.

Sutton, R. S., & Barto, A. G. (1988). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Extending the Contention Scheduling Model of Routine Action Selection: The Wisconsin Card Sorting Task and Frontal Dysfunction

Richard P. Cooper (R.Cooper@bbk.ac.uk)
Department of Psychological Science, Birkbeck, University of London
Malet Street, London WC1E 7HX, UK

Abstract

We extend a previously developed model of routine action selection by incorporating functional components to support behaviour in a simple non-routine task – sorting cards according to a rule that must be discovered by the subject. A minimal extension to the previous model, consisting of an activation-based working memory/inference system in which evidence is incorporated by simply exciting or inhibiting relevant rule nodes, is demonstrated to be capable of capturing basic performance on the task. The task is commonly used in assessing frontal brain injury, and the extended model is further shown to be capable of capturing the gross behavioural characteristics of frontal patients. However, it is argued that a purely activation-based working memory cannot capture the requirements of more complex tasks. The paper thereby demonstrates 1) how the basic routine action model might be extended to more complex behaviours, but 2) that such behaviours require more than simple activation-based memory processes to structure non-routine behaviour over time.

Keywords: Cognitive architecture; contention scheduling; supervisory system; Wisconsin card sorting task; Frontal dysfunction.

Introduction

Norman and Shallice (1986) argued, on the basis of evidence from slips and lapses in naturalistic everyday action and the more severe errors of patients with frontal lesions, that action is controlled by two systems: a low-level routine system (*contention scheduling*) which is responsible for behaviour in routine or mundane situations when our attention is not focused on action, and a higher-level non-routine system (the *supervisory system*) which works by biasing contention scheduling when acting in novel situations or when it is necessary to avoid temptation. (See Shallice (2006) for an updated overview of the account.) In previous work we have developed a model of the contention scheduling component of the theory, and shown how everyday slips and lapses (Cooper & Shallice, 2000), as well as the more flagrant errors of action that occur following frontal (Cooper et al., 2005) and left parietal (Cooper, 2007) brain injury, may be accounted for in terms of damage to different parts of the contention scheduling system. Previous computational work has not, however, considered in any detail how the supervisory system might act to bias contention scheduling in non-routine situations. This paper begins to redress this omission by considering how the contention scheduling model might be extended to capture behaviour on a simple neuropsychological task that requires both inhibition of a prepotent response and generation of novel (or at least novel with respect to the task) behaviours.

The task we consider is the Wisconsin Card Sorting Test (WCST; Grant & Berg, 1948). Subjects in the task are required to sort a series of cards, presented one at a time, into four piles. Drawn on each card is a set of shapes (e.g., two red circles or four blue squares). The piles to which the cards must be sorted are indicated by "target" cards. Each target card differs with respect to the number, colour and shape of items it depicts (see Figure 1). Thus subjects may sort cards to match the targets on any of the three dimensions. During the task, subjects are given feedback after sorting each card, and are required on the basis of this feedback to infer the correct sorting rule and use it for sorting subsequent cards. The trick is that once the subject correctly sorts 10 cards in sequence, the experimenter changes the sorting rule without warning. The subject must then use feedback to adjust his/her sorting rule. This is more difficult than it might at first seem, as some cards match the targets on multiple dimensions, so feedback can be ambiguous. Even so, neurologically healthy subjects have little difficulty on the task. For example, in a sample of 48 subjects tested at Birkbeck, mean sorting accuracy was over 40 correct out of 64 cards. Patients with frontal lesions, however, are known to perform poorly (see, e.g., Stuss et al., 2000), frequently successfully determining the first sorting rule but failing to change rules following negative feedback, i.e., they make perseverative errors.

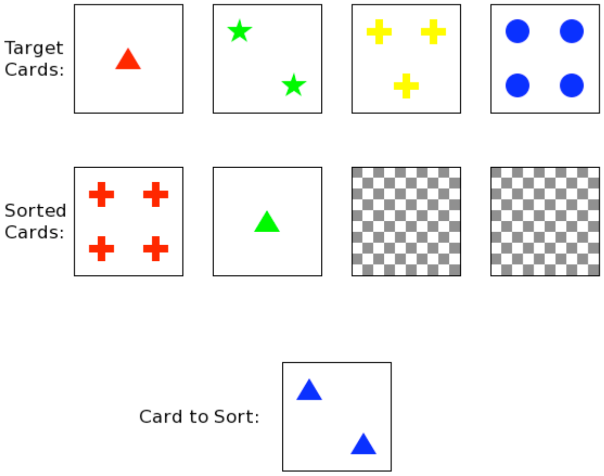


Figure 1: The Wisconsin Card Sorting Test, after two cards have been sorted according to the colour of their symbols and as preparing to sort the third card.

Extending the CS Model: A Naïve Model of WCST

We consider first a naïve and somewhat minimal extension of the contention scheduling model that is capable of completing the WCST at levels comparable to neurologically healthy adults.

Model Assumptions and Description

As discussed above, we assume that behaviour is the product of a simple scheduling system capable of effecting routine sequential behaviour (*contention scheduling*) regulated or biased by a more complex system capable of planning, reasoning and structuring behaviour in the pursuit of intentions (the *supervisory system*). The contention scheduling system has been described in detail elsewhere (e.g., Cooper & Shallice, 2000; Cooper et al., 2005; Cooper, 2007). At its heart is a hierarchically structured interactive activation network in which schemas that encode familiar goal-directed action sequences compete for the control of behaviour, with competition implemented through lateral inhibition between sets of schemas that correspond to alternate ways of achieving a desired goal or sets of schemas that share cognitive or effective resource requirements. The schema network is complemented by further interactive activation networks in which nodes represent objects (with separate object representation networks for different abstract object functional roles). The networks interact, such that schema nodes may excite object representation nodes and vice versa. These interactions encode actions that may be facilitated or afforded by the state of the environment (e.g., that a card on the table might be picked up, or that a card in hand might be placed on the table).

The naïve model of WCST assumes that the contention scheduling system includes schemas for sorting cards according to the different criteria (i.e., *sort by colour*, *sort by number* and *sort by form*), and supplements it with a minimal supervisory (or control) system capable of biasing a specific sorting schema on the basis of positive or negative feedback obtained during the task. The key component of the minimal supervisory system is an activation-based working memory system that contains nodes corresponding to the different schemas that might be used for sorting the cards. It is assumed that when a card is presented for sorting, the most active working memory element biases the corresponding schema within the contention scheduling system, resulting in the card being sorted according to the corresponding criterion (assuming that the scheduling system is functioning correctly). Positive feedback from the experimenter (indicating that the card was sorted correctly) results in excitation of all working memory nodes consistent with the attempt, while negative feedback (if the card was sorted incorrectly) results in inhibition of all working memory nodes consistent with the attempt. Thus, if the card to be sorted depicts one green triangle, and the card is placed under the left-most target card (which in the standard test shows one red triangle), positive feedback will result in excitation of both the sort-to-number and the sort-to-form working memory nodes, while negative feedback will result in inhibition of both of these nodes.

In order to give behaviour coherence over time, we assume that the activation of nodes within working memory persists over time, but that this persistence is imperfect (i.e., activation decays). We also assume that the activation of nodes is subject to noise. For simplicity we adopt for the working memory component the same activation-update equations used in the interactive activation networks, namely:

$$A_{t+1} = \overline{\sigma}\left(\sum_{i=0}^{t} P^i . I_{t-i}\right)$$

where A_t is the activation of a node on card sorting step t, I_t is the net input (excitation or inhibition plus noise) to the node on card sorting step t, P is a persistence parameter (see below) and $\overline{\sigma}$ is a sigmoidal squashing function bounded between 0 and 1 whose output, with zero net input, is 0.1.

With this activation-update equation, activation of working memory nodes tends to 0.1 in the absence of any net excitation or inhibition. Net excitation pushes the activation of a node towards 1, while net inhibition suppresses the activation of node towards 0. Given this formulation, the behaviour of the supervisory aspects of the model is determined by four parameters:[1]

- P: The persistence of working memory representations across card sorting steps.
- N: Standard deviation of noise added to the input of working memory representations on each card sorting step.
- F_e: Excitatory activation of matching working memory representations following positive feedback – a non-negative real number.
- F_i: Inhibitory activation of matching working memory representations following negative feedback – a non-negative real number.

Behaviour of the Model

As anticipated, with appropriate parameter settings the model is capable of performing the WCST with relatively few errors. Thus, in a typical run with $P = 0.85$, $N = 0.05$, $F_e = 0.25$ and $F_i = 0.75$, the model succeeds in correctly sorting approximately 55 cards out of 64, with all errors occurring following a change in sorting category. This corresponds to the upper limit of normal performance.

A full explanation of the model's behaviour requires explanations at the level of both working memory and contention scheduling. We begin with working memory. Figure 2 shows the activation profiles of working memory elements over the complete duration of one administration of the WCST (64 cards) with the above parameter settings.

[1] Additional parameters govern the behaviour of the contention scheduling component of the model. For all simulations reported in this paper we fix those parameters to the values used in other recent work (e.g., Cooper et al., 2005; Cooper, 2007).

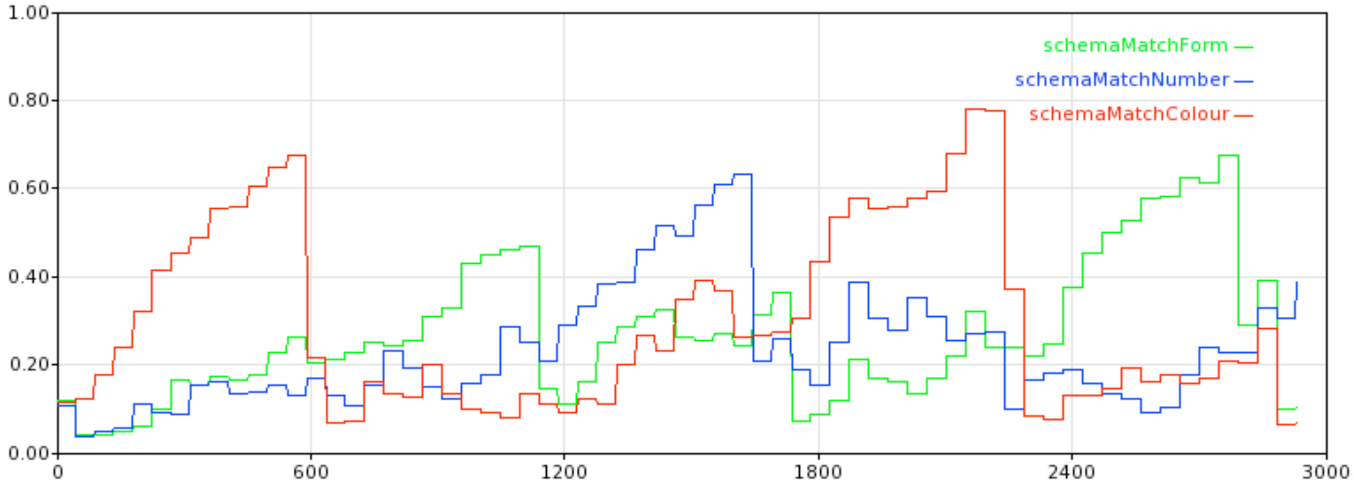


Figure 2: Activation profiles of working memory elements over the duration of the WCST. Activation is plotted on the vertical axis with processing cycles plotted on the horizontal axis.

Each step in the graphs (corresponding to approximately 40 processing cycles, see below) corresponds to the sorting of one card. On the first step working memory elements corresponding to all three sorting schemas have activations close to resting levels, with only noise differentiating them. In this example, the most active element is that which corresponds to matching to *form*. This is therefore selected as the initial rule. The corresponding schema within contention scheduling then receives top-down excitation from the supervisory system, resulting (as discussed below) in the first card being placed under the target card that shares the *form* feature. The first card depicts one green triangle, so matching to form involves matching this card with the left-most target card, which depicts one red triangle. This is incorrect – colour is initially the correct sorting criterion – so negative feedback is provided. This results in inhibition of the working memory representations of all schemas that are consistent with the current sorting attempt. Note though that this attempt matched against two criteria, sorting by form and sorting by number. Hence, the working memory representations of both receive inhibition. The working memory representation corresponding to sorting by colour is the only one not to receive inhibition, and hence is the representation that is most active when the second card is presented. The second card is therefore sorted by colour. Positive feedback results in excitation of this working memory representation, ensuring that it remains the most active, while the activations of the other nodes begin to return to their resting levels.

The model continues sorting by colour, with feedback occasionally providing support for multiple working memory representations (when a card matches against more than one criterion). Only when the criterion changes (after ten successful sorts to the colour criterion) does sorting to colour result in negative feedback. The representation of sorting to colour in working memory is rapidly inhibited, while the representation of sorting to form is excited (through positive feedback when a card matches against the form criterion). Once the activation of the representation of sorting to form exceeds that of sorting to colour (and sorting to number) the model switches to sorting to form (i.e., on presentation of a card, top-down excitation is passed to the schema that corresponds to sort-by-form within the contention scheduling system).

We turn now to the contention scheduling component. Figure 3 shows the activation profile of schema nodes within this component of the model over the first two sorting events. On presentation of the first card, top-down excitation is passed to the *sort-by-form* schema as described above. This results in that schema's activation rising to its maximum level during the first few processing cycles. The *sort-by-form* schema activates in turn the subschemas corresponding to *pick-up card* and *put-down card*. It also activates representations of cards in the object representation networks (which are not shown in the figure). Thus, the presented card (rather than, e.g., the target card) is activated as the card to be picked-up and, once the presented card is held, the target key card which matches this on the form feature is activated as the destination for the *put-down card* schema. The first card is therefore placed under the left-most key card.

Processing is similar during sorting of the second card

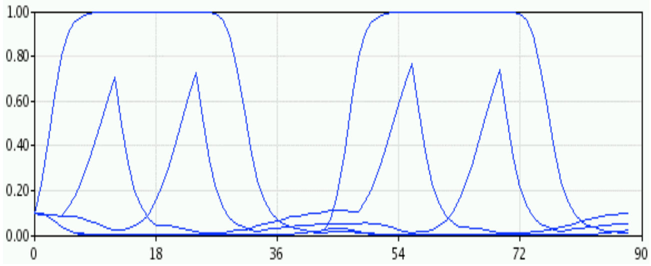


Figure 3: Activation profiles of schema nodes within contention scheduling during two consecutive sorting events. The vertical axis shows activation while the horizontal axis shows processing cycles. The first peak within each sorting event (cycles 12 and 56) corresponds to picking up a card while the second corresponds to placing it in the appropriate target pile (cycles 24 and 69).

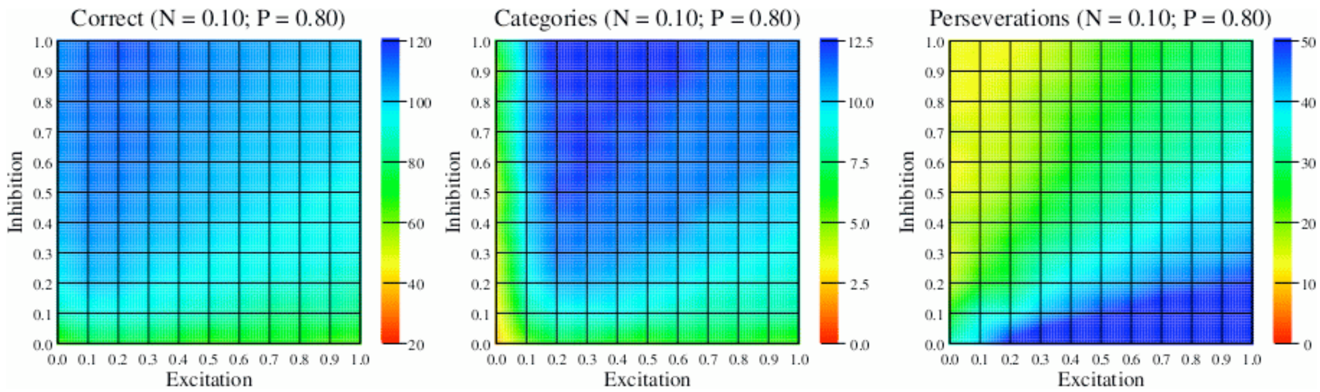


Figure 4: Contour maps showing the number of cards correctly sorted (out of 128), number of categories achieved and classical perseverations when *N* is 0.1, *P* is 0.8, and F_i and F_e vary from 0.0 to 1.0. Data are averaged over 10 attempts at each parameter combination.

(cycles 42 to 85), except that it is the *sort-by-colour* schema that is most active, and hence the card that is being sorted is placed under the target key card that matches it on the colour feature.

Parameter Study 1: "Normal" Behaviour

The behaviour of this naïve model depends upon the values of the model's four parameters. Thus, good performance requires that inhibition following negative feedback (F_i) is substantially greater than excitation following positive feedback (F_e). If not, the model will perseverate following negative feedback, as positive feedback during a run of correct responses will result in the working memory representation of the correct sorting criterion becoming highly active, and it will take several consecutive errors following a change in criterion for this activation to subside and be exceeded by that of a competing sorting criterion. At the same time, persistence must be relatively high. If it is too low, behaviour on each card sort will be based primarily on feedback from the previous trial – feedback that can be ambiguous if a card matches against multiple criteria.

Given the potential complexity of interactions between parameter values, two systematic surveys of the parameter space were conducted. In parameter study 1, the model's susceptibility to standard perseverative errors was investigated by varying F_e, F_i and *P* from 0.0 to 1.0 in steps of 0.1 with *N* at 0.1, 0.2 and 0.3. The model was run 10 times at each point in the parameter space, and three dependent variables – the number of correct sorts, categories achieved and classical perseverative errors – were recorded for each run of the model. In each case the model was required to sort 128 cards, with the simulated experimenter changing the sorting criterion whenever 10 consecutive cards were sorted correctly. Thus, following Stuss et al. (2000) but unlike most behavioural studies, the test was not terminated after 6 categories had been achieved. Scoring was automated by a separate program that implemented the scoring algorithm described by Heaton (1981).

These simulations demonstrated that, for each value of N, there are values for the other parameters that result in accurate sorting with few errors (e.g., *N* = 0.1, *P* = 0.9, F_i = 0.1, F_e = 0.8) that is similar to the behaviour of normal participants. They also demonstrated, however, that the model generates high numbers of perseverative errors (i.e., more than 1/3rd of responses) and achieves relatively few categories when *P* is high and F_i is low relative to F_e. Thus, Figure 4 shows contour maps for the number of cards correctly sorted, number of categories achieved, and number of perseverative errors when *N* is 0.1, *P* is 0.8, and F_i and F_e vary from 0.0 to 1.0. From the figure, it can be seen that *N* is 0.1, *P* is 0.8, F_i is 0.1 and F_e is 0.7, the model correctly sorts 60 to 80 cards (out of 128), obtains 5.0 to 7.5 categories, but produces 40 to 50 perseverative responses.

Parameter Study 2: "Frontal" Behaviour

It is clear from parameter study 1 that the naïve model is susceptible to perseverative behaviour, at least when persistence is high and feedback inhibition is low relative to feedback excitation. While this echoes the behaviour of certain frontal patients, the number or proportion of perseverative errors alone is a coarse measure of behaviour. Parameter study 2 therefore sought to evaluate the model's performance against a published dataset with a more fine-grained scoring system, namely the dataset and scoring system of Stuss et al. (2000).

Stuss et al. (2000) tested six groups of patients (four groups with frontal lesions centred in different areas and two non-frontal patient groups) and control participants on three versions of the WCST, with increasing instructional support on successive versions. In scoring participant behaviour, errors were subdivided into four categories: perseveration of preceding category (PPC: a response that matches the previous sorting criterion but not the current one), perseveration of preceding response (PPR: a response that matches exactly the features matched on the immediately preceding incorrect trial), set loss (an error following attainment of the current sorting category, as demonstrated by three consecutive correct responses, at least one of which was non-ambiguous) and other errors. Subtle differences between the various frontal groups were observed. For example, when participants were told the possible sorting criteria prior to the test (Stuss et al.'s 64A

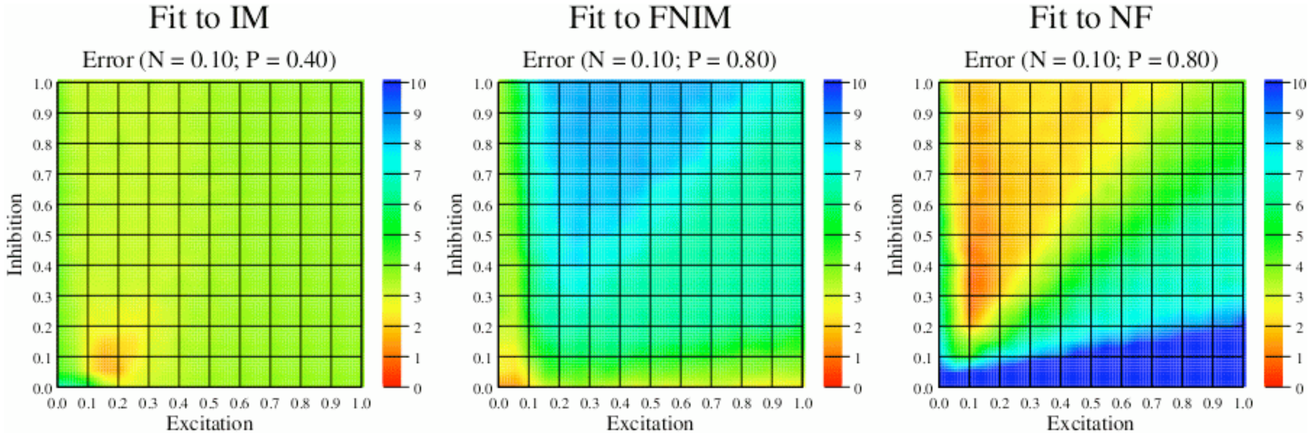


Figure 5: Goodness of fit plots for best fitting planes in $F_e \times F_i$ space for each of the three groups. The best fit to the inferior medial group (IM: left panel) occurs with $P = 0.40$. The best fits for the frontal non-inferior medial (FNIM: centre panel) and the non-frontal (NF: right panel) groups occur with P = 0.80.

condition), inferior medial patients achieved significantly fewer sorting categories and produced significantly more set loss errors than control and non-frontal patients, but they did not make significantly more PPC or PPR errors. Other frontal groups achieved even fewer categories and made fewer set loss errors than the inferior medial patients, but made significantly more PPC and PPR errors than the inferior medial, non-frontal and control groups.

Parameter study 2 therefore explored the behaviour of the model following variation of F_e, F_i and P using the scoring system of Stuss et al. (2000). The aim was to replicate the behaviour of each of Stuss et al.'s participant groups and thereby further understand the possible nature of the deficit in each case. Note, however, that Stuss et al. found no significant differences in the pattern of behaviour between their right dorsolateral, left dorsolateral and superior medial groups – all three groups produced qualitatively similar behaviour across the four dependent variables. These frontal groups did differ, however, from the inferior medial group. Our analysis therefore merges these groups. Similarly, Stuss et al. found no significant differences between their left non-frontal, right non-frontal and control groups. Our analysis also merges these groups. This results in three groups: inferior medial (IM), frontal non-inferior medial (FNIM) and non-frontal (NF). Descriptive statistics for each group based on the 64A version of the task are shown in Table 1.

To explore the parameter space F_e and F_i were varied from 0.00 to 1.00 at intervals of 0.05 and P was varied from 0.10 to 0.90 at intervals of 0.10. N was fixed at 0.10. The model was run 50 times with 64 cards to sort at each combination of parameter values (totalling 21 × 21 × 9 × 50 = 198450 runs). Four dependent measures were collected for each run (categories achieved, PPC errors, PPR errors and set loss errors, all following definitions given in Stuss et al., 2000). For each of the three groups and for each point in parameter space, a goodness of fit measure was then calculated as the maximum of the fits to the four dependent measures, where the fit to each of the dependent measures was calculated as the difference between the simulated mean value of that dependent measure at the point in parameter space and the observed mean value of that dependent measure for the specific group divided by the observed standard deviation of that dependent measure for the group. Thus, a fit of less than one to any group would correspond to a case where each of the four dependent measures was within one standard deviation of the observed group means. Figure 5 shows plots of this goodness of fit measure for the best fits for each group in $F_e \times F_i$ space.

	Categor-ies	PPC Errors	PPR Errors	Set Loss Errors
NF	4.01 (0.44)	7.15 (1.09)	0.94 (0.68)	0.93 (0.48)
FNIM	1.08 (0.46)	24.27 (6.04)	11.68 (3.18)	1.14 (0.63)
IM	2.60 (0.60)	10.60 (1.70)	2.90 (0.9)	2.60 (0.70)

Table 1: Means (standard deviations) for WCST behaviour of three patient groups (derived from Stuss et al., 2000)

From Figure 5 it can be seen that the best fit to the IM group is obtained when P is 0.40, F_i is between 0.05 and 0.10 and F_e is between 0.15 and 0.20. This fit is approximately 1.5. A slightly better fit is obtained for the FNIM group, of 1.0, when P is 0.80, F_i is 0.00 and F_e is 0.05. Only for the NF group is a fit of less than one obtained, and when P is 0.80 this level of goodness of fit is obtained for a wide region of $F_e \times F_i$ space (and this result holds for other values of $P \geq 0.70$).

Discussion

The naïve model has been shown to be capable of both normal and frontal-like behaviour on the WCST (parameter study 1), but the scan of the parameter space in parameter study 2 found only modest fits for the two subgroups of frontal patients, with the best fits in each case failing to be simultaneously within one standard deviation for all dependent measures. There may be good reason for this – none of the subject groups is completely homogenous, and even if all patients in a group can be argued to have a qualitatively similar deficit, that deficit is likely to vary in degree. Nevertheless the naïve model does provide some insight into the deficits. Inferior medial frontal patients are particularly prone to PPR errors and set loss errors. These errors occur when excitation, inhibition and persistence are all low. The latter provides a clear intuitive account of set loss errors: if persistence is low it is likely that the model will frequently fail to maintain a sorting rule, even after successfully discovering the rule. If both excitation and inhibition are low the model effectively makes little use of either positive or negative feedback. This explains to some extent the existence of perseverative errors. However, the type of perseverative error depends upon maintaining some record of a sorting rule. For PPR errors, this cannot be the most recent successful sorting rule – that would result in PPC errors. Rather, it is the rule apparently used *unsuccessfully* on the previous trial. PPR errors are therefore a more accurate reflection of failure to respond to negative feedback than are PPC or classical perseverative errors.

General Discussion

In comparison with previous work, the model shares a family resemblance with models inspired by the operation of prefrontal cortex (e.g., Dehaene & Changeux, 1991; Amos, 2000; Rougier et al., 2005). Like these models, behaviour in the extended contention scheduling model is a function of bias operating on a routine system that, in the case of card sorting, embodies simple stimulus-response links. The work presented here differs from the above, however, in considering the behaviour of different frontal subgroups as revealed by Stuss et al (2000).

The extended contention scheduling model does moderately well at accounting for both normal and impaired performance, but there are severe limitations to the working memory module. Both basic assumptions – that working memory comprises nodes corresponding to atomic symbols and that evidence accrues only through processes of excitation and inhibition – are problematic. Thus, the approach does not generalize well to other non-routine behaviours such as solving Tower of Hanoi problems, which appear to require both the storage and manipulation of structured information within working memory and the manipulation of that information according to operations more complex than simple excitation or inhibition.

Indeed, in an alternative extension of the contention scheduling model to be reported elsewhere working memory has been modelled as a collection of feature-value pairs (similar to production system approaches). Space limitations prevent a full description of the model. However, as with the naïve model presented here the alternative model was able to capture normal and impaired performance on the WCST. More critically, the working memory structures of the alternative model allow it to be applied to other non-routine tasks, including solving Tower of London problems and generating random sequences of numbers – both non-routine tasks that have frequently been discussed in the literature on cognitive control. In these tasks, autonomous functioning of the lower-level system supports the solution of one-move tower problems and the generation of sequences of associated numbers (e.g., digits increasing by two). Again, the role of the supervisory system is to modulate behaviour. The system allows, in the first case, the solution of tower problems where intermediate states are required, and in the second, detection and inhibition of stereotyped responses before they are produced. This is achieved through operations on the content of working memory which depend on relations between working memory elements. It is unclear how the working memory mechanisms of the naïve model (or of other models such as those mentioned above, and also the recent influential working memory model of O'Reilly and Frank (2006)) might meet such a challenge.

References

Amos, A. (2000). A computational model of information processing in the frontal cortex and basal ganglia. *Journal of Cognitive Neuroscience*, *12*, 505–519.

Cooper, R. P. (2007): Tool Use and Related Errors in Ideational Apraxia: The Quantitative Simulation of Patient Error Profiles. *Cortex*, *43*, 319–337.

Cooper, R.P., Schwartz, M.F., Yule, P., & Shallice, T. (2005). The simulation of action disorganisation in complex activities of daily living. *Cognitive Neuropsychology*, *22*, 959–1004.

Cooper, R.P., & Shallice, T. (2000). Contention Scheduling and the control of routine activities. *Cognitive Neuropsychology, 17*, 297–338.

Dehaene, S. & Changeux, J-P. (1991). The Wisconsin Card Sorting Test: Theoretical analysis and modelling in a neuronal network. *Cerebral Cortex*, *1*, 62–79.

Grant, D.A. & Berg, E.A. (1948). A behavioural analysis of degree of reinforcement and ease of shifting to new responses in a Weigl-type card-sorting problem. *Journal of Experimental Psychology*, *38*, 404–411.

Heaton R. (1981). *A manual for the Wisconsin Card Sorting Test*. Psychological Assessment Resources. FL: Odessa.

Norman, D.A. & Shallice, T. (1986). Attention to action: willed and automatic control of behaviour. In R. Davidson, G. Schwartz & D. Shapiro (eds.) *Consciousness and Self Regulation, Volume 4*. NY: Plenum.

O'Reilly, R.C. & Frank, M.J. (2006). Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation*, *18*, 283–328.

Rougier, N.P., Noelle, D.C., Braver, T.S., Cohen, J.D. & O'Reilly, R.C. (2005). Prefrontal cortex and flexible cognitive control: Rules without symbols. *Proceedings of the National Academy of Sciences of the USA*, *102*, 7338–7343.

Stuss, D.T., Levine, B., Alexander, M.P., Hong, J., Palumbo, C., Hamer, L., Murphy, K.J. & Izukawa, D. (2000). Wisconsin Card Sorting Test performance in patients with focal frontal and posterior brain damage: Effects of lesion location and test structure on separable cognitive processes. *Neuropsychologia*, *38*, 388–402.

Is the Linear Ballistic Accumulator Model Really the Simplest Model of Choice Response Times: A Bayesian Model Complexity Analysis

Chris Donkin (Chris.Donkin@newcastle.edu.au)

Andrew Heathcote (Andrew.Heathcote@newcastle.edu.au)

Scott Brown (Scott.Brown@newcastle.edu.au)
School of Psychology, The University of Newcastle
Callaghan, NSW 2303 Australia

Abstract

Brown and Heathcote (2008) proposed the LBA as the simplest model of choice and response time data. This claim was, in part, based on the LBA requiring fewer parameters to fit most data sets than the leading alternative, the Ratcliff diffusion model (Ratcliff & Tuerlinckx, 2002). However, parameter counts fail to take into account functional form complexity, or how the parameters interact in the model when being estimated from data. We used p_D, or the "effective number of parameters", calculated from Markov Chain Monte Carlo samples, to take these factors into account. We found that in a relatively simple, simulated, data set and on average in a complex, real, data set that the diffusion had fewer effective parameters than the LBA.

Keywords: decision models; response time; Bayesian statistics; model complexity.

A wide range of experimental tasks involve a decision between at least two alternatives. Some believe that the process behind making simple decisions is the same regardless of what the decision is about. The most successful class of theories about simple decision processes are evidence accumulator models. There are many types of evidence accumulator model that differ slightly from one another. However, the central assumption common to all is that, when making a decision about a stimulus, evidence is gradually accumulated for each alternative response. Once there is enough evidence for one particular response that response is made, and the time taken to accumulate that evidence is the decision time. The most frequently applied evidence accumulator model for decisions between two alternatives is the Ratcliff diffusion model (Ratcliff, 1978; Ratcliff & Rouder, 1998; Ratcliff & Tuerlinckx, 2002). For example, Ratcliff and colleagues have used the diffusion model to account for the decision process in lexical decision tasks (Ratcliff, Gomez, & McKoon, 2004), recognition memory tasks (Ratcliff, 1978), to investigate the effects of aging on cognitive performance (e.g. Ratcliff, Thapar, & McKoon, 2004). Ratcliff, Segraves, and Cherian (2003) also present neural evidence consistent with the diffusion model.

Brown and Heathcote (2008) recently proposed an alternative evidence accumulator model of the decision process: the Linear Ballistic Accumulator (LBA) model. The LBA was proposed as a simpler model of decision than the diffusion model. The claim of simplicity was based in part on the fact that the LBA assumes one less source of noise in the decision process. That is, in constrast to the diffusion model, evidence accumulation in the LBA is ballistic (i.e. without moment-to-moment variability). This simplification, enables the derivation of full analytic expressions for the model's full probability density function. Despite this simiplificaiton, Brown and Heathcote (2008) show that the LBA is able to account for benchmark data from two-choice tasks (Ratcliff & Rouder, 1998; Ratcliff, Gomez, & McKoon, 2004) [1]. LBA parameters have also been shown to have neural correlates (Forstmann et al., 2008; Ho, Brown, & Serences, submitted).

Brown and Heathcote (2008) also claimed the LBA is simpler because, when fiting standard two-choice data, it required one less parameter than the most recent version of the diffusion model (Ratcliff & Tuerlinckx, 2002). Myung and Pitt (1997), however, explain that the number of free parameters, k, does not necessarily provide a full indication of model complexity. Specifically, k fails to take into account functional form complexity (i.e., differences in flexability between different mathematical functions), or how the parameters interact when parameters from the model are estimated from data. Spiegelhalter, Best, Carlin, and van der Linde (2002) proposed a method to address these aspects of model complexity using the deviance information criterion (DIC) and an associated estimate, p_D, of the effective number of model parameters. These quantities are estimated using posterior samples obtained by Bayesian Markov Chain Monte Carlo (MCMC) methods. We use these methods to investigate the claim that the LBA is a "simpler" model of the decision process. To begin we provide a brief overview of the diffusion and LBA models.

Overview of Models

Consider the following example – participants are shown a patch of 64x64 pixels, each of which are either white or black, and the asked whether the stimulus is mostly bright or mostly dark. The Ratcliff diffusion model begins by assuming that participants sample information continuously from the stimulus. Each sample of information counts as evidence for one of the two responses and is used to update an evidence total, say x, shown by the irregular line in the left panel of Figure 1.

[1] Brown and Heathcote (2008) also show that the LBA is able to account for decisions between more than two alternatives becasue it allows one accumulator for each choie. As the Ratcliff diffusion model has not been extended to the multiple choice case we will focus on the two choice case.

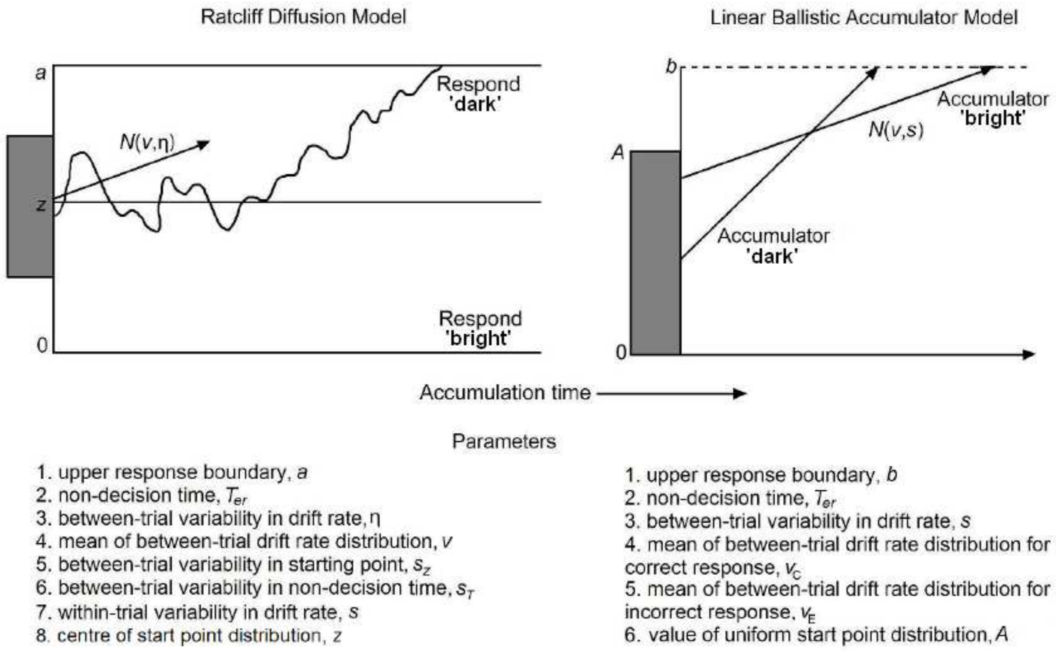

Ratcliff Diffusion Model
a
z
0
N(v,η)
Respond 'dark'
Respond 'bright'
Linear Ballistic Accumulator Model
b
A
0
N(v,s)
Accumulator 'bright'
Accumulator 'dark'
Accumulation time
Parameters
1. upper response boundary, a
2. non-decision time, T_er
3. between-trial variability in drift rate, η
4. mean of between-trial drift rate distribution, v
5. between-trial variability in starting point, s_z
6. between-trial variability in non-decision time, s_T
7. within-trial variability in drift rate, s
8. centre of start point distribution, z
1. upper response boundary, b
2. non-decision time, T_er
3. between-trial variability in drift rate, s
4. mean of between-trial drift rate distribution for correct response, v_C
5. mean of between-trial drift rate distribution for incorrect response, v_E
6. value of uniform start point distribution, A

Figure 1: Overview of the diffusion and LBA models (left and right panel, respectively)

Total evidence begins at some starting point, $x = z$, and evidence that favours a “bright” response decreases the value of x and evidence for a “dark” response increases the value of x. Evidence accummulation continues until x reaches one of the response boundaries, the horizontal lines at 0 or a in Figure 1. The choice made depends upon which boundary was reached, a for “dark” and 0 for “bright” response. The time taken to make the choice is the accumulation time plus a non-decision time component, T_{er}, composed of things such as encoding time and the time taken to make a motor response.

Consider a stimulus composed of almost 100% white pixels. When a participant samples from this stimulus almost all of the evidence will favour a “bright” response, and so the accumulation total will quickly increase towards a. The average rate of this accumulation is called the *drift rate*, v, and variability in moment-to-moment accumulation is assumed to take the value s. Ratcliff (1978) added the additional assumption that drift rate also varies from trial-to-trial according to a normal distribution with mean v and standard deviation η. Ratcliff and Rouder (1998) incorporated between-trial variability in the start point of acccumulation, assuming that z follows a uniform distribution on $[z - \frac{s_z}{2}, z + \frac{s_z}{2}]$. Finally, Ratcliff and Tuerlinckx (2002) included between-trial variability in non-decision time T_{er} in the form of a uniform distribution on $[T_{er} - \frac{s_t}{2}, T_{er} + \frac{s_t}{2}]$.

In the LBA there are separate accumulators gathering evidence for each of the “bright” and “dark” responses. These accumulators are assumed to be linear, ballistic and independent. That means evidence accumulation has a linear increase with no within-trial variability (i.e,. is ballistic rather than stochastic as in the diffusion model), and accumulation in one accumulator has no effect on the other accumulator(s). The amount of evidence an accumulator begins with on each trial is sampled (separately for each accumulator) from the interval $[0, B]$. The evidence in each accumulator increases at a linear rate determined by the drift rate parameters, v_b and v_d, for bright and dark responses, respectively. Accumulation continues until evidence in one accumulator reaches a response boundary, a^2 which is usually assumed to be the same for all accumulators. The accumulator which reaches the boundary first selects its associated response and accumulation time plus non-decision time, T_{er}, gives the reaction time. As in the Ratcliff diffusion model, the drift rate is assumed to vary between-trials according to a normal distribution with mean v and standard deviation η.

To sum up, the diffusion model has the parameters $(a, z, s_z, T_{er}, s_t, v, s, \eta)$ and the LBA has the parameters $(a, B, T_{er}, v_1, v_2, \eta)$, where v_i refers to the mean drift rate in the accumulator for the i_{th} response. The parameterisation for each model, however, differs depending on the design of the data from which the data were obtained. There is, therefore, no fixed difference in the number of parameters between the models. There are, however, parameterisations of these models which are commonly applied. For example, when there is no bias for one response over the other then the z parameter of the diffusion model can be fixed at $\frac{a}{2}$, reducing the number of free parameters in the diffusion model by one. Also, in order to solve a scaling property common to all evidence accumulator models, the s parameter is generally fixed at 0.1. Sim-

[2] In previous applications of the LBA a and B have been labelled b and A, respectively. We adopt this alternative labelling here to facilitate equality in parameter names across models.

ilarly, when fitting the LBA, drift rates for correct and error responses tend to be assumed equal for both choices unless the choice corresponds to an experimental manipulation (e.g., word vs. non-word in a lexical decision task or studied vs. unstudied in a recogniton memory task). Drift rates for error responses are also typically assumed to be fixed at one minus the drift rate for correct response, solving the scaling property for the LBA. This means when the LBA has been applied then usually only one drift rate parameter is estimated– the drift rate for correct responses. Based on these standard parameterisations, Brown and Heathcote (2008) concluded that the LBA uses one less parameter than the diffusion model to account for data typical of two-choice tasks. This finding, combined with some apparently simpler structural assumptions, led Brown and Heathcote (2008) to conclude that LBA was simpler than the diffusion model. We now explore whether the p_D measure of model complexity agrees with the author's conclusions.

Model Complexity

An overly complex model can provide an excellent fit to a given set of data, yet still not be considered to give a satisfying account of the underlying process. In particular, a more complex model can "overfit" the data by fitting the random error specific to a particular sample as well as the structure due to the underlying processes. Becasue only the structre re-occurs in new data, overfitting limits the model's ability in terms of prediction. Myung (2000) suggests that at least two factors contribute to model complexity – the number of parameters in the model and the functional form of the model, which determines how the parameters interact. Functional form complexity can differ between models with the same number of parameters when one model is able to produce a wider range of predictions than the other. In any particular experimental design, the degree to which the effects of functional form complexity are observed depends on the interaction between model and data.

A number of model selection methods take into account functional form complexity. We will focus on one such measure: the Deviance Information Criterion (DIC) (Spiegelhalter et al., 2002). DIC has been applied across a wide range of fields including psychology (e.g., Myung, Karabatsos, & Iverson, 2005). Vandekerckhove, Tuerlinckx, and Lee (2008) used DIC to compare various instantiations of the diffusion model. The DIC can be considered the Bayesian version of the Akaike Information Criterion (AIC; Akaike, 1973), but with a complexity penalty term which takes into account functional form complexity, rather than simply counting the number of free parameters, as in AIC.

DIC can be computed from MCMC samples of a model's posterior parameter distributions. Let θ represent such a sample. Deviance can be written as $D(\theta) = -2\log L(y|\theta)$, where $L(y|\theta)$ represents the likelihood of data vector y given parameters θ. Then $D(\bar{\theta})$ is the deviance of the estimated posterior mean parameters and $\overline{D}(\theta)$ is the mean of the distribution of posterior samples. DIC can be expressed in two parts as $DIC = D(\bar{\theta}) + 2p_D$, where $p_D = \overline{D}(\theta) - D(\bar{\theta})$, where $D(\bar{\theta})$ is a measure of misfit between data and model predictions, and $2p_D$ is a penalty for the "effective" number of parameters in the model (Spiegelhalter et al., 2002). The p_D measure adjusts the number of parameters in the model to take account of functional form complexity. Larger values of p_D indicate a more complex model able to potentially predict a greater the range of patterns of data. A better model, which achieves a balance between fit and complexity, has a smaller DIC.

Posterior sampling for both the Ratcliff diffusion and LBA models have been implemented using the Bayesian MCMC program WinBUGS (diffusion: Vandekerckhove et al., 2008; LBA: Donkin, Averell, Brown, & Heathcote, 2009). We use these implementations to calculate DIC and p_D, allowing us to compare the functional form complexity between the models. Because DIC and p_D are dependent on the data to which the models are applied we will present the results of fits to two different sets of data: simulated data generated by the diffusion model, and a benchmark data set from Ratcliff and Rouder (1998).

Estimating p_D and DIC for the LBA and Diffusion Models

Simulated Data

The first set of data were generated from a diffusion process with parameters given in Table 1. Our simulated data set was intended to mimick data from a two-choice task with a single experimental factor where stimuli were varied so as to only affect the difficulty of the task. This meant that only the drift rate parameter, v, was allowed to vary across the three conditions. All other parameters $(a, sz, T_{er}, s_t, s, \eta)$ were assumed to be constant across all conditions. We also fixed z to be $\frac{a}{2}$, representing unbiased responding. This parameterisation is standard for fitting data from experiments which have a single within-subjects condition which varies from trial-to-trial (e.g. Ratcliff, Gomez, & McKoon, 2004). The simulated data can be thought of as coming from a single participant who completed 1000 trials in each of the three difficulty conditions.

When fitting both the diffusion model and the LBA model, parameters were fixed to match the assumptions made when generating the data; so only drift rate was allowed to vary between the three difficulty conditions. This means that for the diffusion model we have eight free parameters $(a, s_z, T_{er}, st, \eta, v_1, v_2, v_3)$, and for the LBA seven free parameters $(a, B, T_{er}, \eta, v_1, v_2, v_3)$. Unbiased responding in the LBA corresponds to having the same values of a and B for each response. Posterior samples were obtained for both models using their WinBUGS implementations. For each model three chains each containing 10,000 MCMC samples were collected, with the first 3,000 samples for each chain were discarded as burn-in. Visual inspection of the chains suggested that after burn-in samples collected from each chain were from the same stationary distribution, which we now assume to be the true posterior distribution.

Table 1: Mean of posterior samples for parameters from the diffusion and LBA models for fits to data generated from diffusion model. DIC and p_D are also reported for each model.

Parameter	Data	Diffusion	LBA
a	.125	.128	.252
s_z / B	.044	.034	.432
η	.133	.123	.237
T_{er}	.435	.432	.237
s_t	.196	.196	-
v_1	.1	.103	.609
v_2	.23	.226	.74
v_3	.363	.369	.882
DIC	-	-183.76	-47.55
p_D	-	5.97	6.81

Table 1 contains mean posterior samples for each parameter for both the diffusion and LBA models. The average posterior diffusion model parameter samples are close to the parameters used to generate the data, as expected. The average posterior LBA parameters are close to parameters estimated using non-Bayesian methods of fitting (e.g. maximum likelihood estimation) to the same data set.

DIC and p_D values are also given in Table 1. As one might expect, the DIC for the diffusion model is smaller than the DIC value for the LBA model (-183.76 and -47.55, for diffusion and LBA respectively), suggesting that the diffusion model provides a better account than the LBA of data simulated from a diffusion process. Quite unexpectedly, however, the p_D value for the diffusion model is also smaller than that for the LBA model, p_D equal to 5.97 and 6.81 respectively. This suggests that – despite the diffusion model having more free parameters than the LBA model – when functional form complexity is taken into account, the number of “effective” parameters is actually smaller than that of the LBA model.

At least for these simulated data, from a very simple experimental design, the results seem clear – the diffusion model is less complex than the LBA. As previously stated, however, functional form complexity depends upon the data being modelled. We turn now to actual data, to a data set which has become a benchmark data set for models of choice and response time (Brown & Heathcote, 2008; Vandekerckhove et al., 2008).

Ratcliff and Rouder’s (1998) Data

Ratcliff and Rouder (1998) performed a simple brightness discrimination task with two within-subject factors: brightness and instructions. There were 33 levels of brighness used, determined by the proportion of white vs. black pixels in a 64x64 display (brightness was varied randomly from trial-to-trial). Between blocks of trials, participants were given instructions on whether to respond with an emphasis on speed or an emphasis on accuracy.

We fit diffusion and LBA models seperately to data from three individual participants, each of whom completed almost 8000 trials. Both models have previously been fit to the Ratcliff and Rouder (1998) data sets using non-Bayesian estimation techniques (diffusion: Ratcliff & Rouder, 1998; LBA: Brown & Heathcote, 2008). We used very similar parameterisations to that used in the original fits with three exceptions. First, for the diffusion model we included between-trial variability in non-decision time. This variability was included in the diffusion model as it has been standard practice since Ratcliff and Tuerlinckx (2002). Second, for the LBA both the upper bound of the uniform distribution of starting point of accumulation, B, and response threshold, a, were allowed to vary between speed and accuracy conditions. Brown and Heathcote (2008) assumed $B = a$ in the speed-emphasis condition, but we found that fit was greatly improved by removing this constraint. For the diffusion model we followed Ratcliff and Rouder (1998) and assumed that only boundary separation, a was allowed to vary between speed and accuracy conditions. Third, we found that the diffusion gave much better fits to data by estimating between-trial variability in start point of accumulation for speed and accuracy conditions separately. This contrasts with Ratcliff and Rouder (1998) approch, where s_z was fixed at $a/20$ for both speed and accuracy conditions.

For both models, only drift rate was allowed to vary between brightness conditions. Although there were 33 brightness conditions in the original data, the conditions were collapsed to seven since visual inspection suggested that the majority of brightness levels which were either very difficult or very easy were homogenous in RT and accuracy. This meant that for the diffusion model $(a_{acc}, a_{spd}, s_{z_{acc}}, s_{z_{spd}}, T_{er}, s_t, \eta)$ were free parameters, and for the LBA $(a_{acc}, a_{spd}, B_{acc}, B_{spd}, T_{er}, \eta)$ were free parameters. When combined with the seven drift rate parameters common to both models, there were 14 free parameters for the diffusion model, and 13 free parameters for the LBA model.

A single chain of 10,000 samples was collected for each of the LBA and diffusion models, with the first 3,000 samples discarded from analysis as burn-in. Again, visual inspection of the chain confirmed that stationarity after burn-in. Table 2 contains mean posterior parameter values for each model and each participant. Though, for brevity we do not present them here, plots of model predictions and data confirm that the average parameter values provide a good fit to the data. The quality of fit between model and data was greater for the diffusion model than the LBA. This is reflected in DIC and p_D values reported in Table 2: for all participants the diffusion model had a smaller DIC value than the LBA model[3]. As

[3] Donkin, Brown, and Heathcote (2009) have shown that an LBA model where the sum of correct and error drift rates are not over-constrained to be one can provide a large improvement in quality of fit. This comes, however, at the expense of an increase in the number of free parameters. Since we wish the present discussion to be a retrospective look at the claims of Brown and Heathcote (2008) we discuss this no further here.

Table 2: Mean of posterior samples for parameters from the diffusion and LBA models for fits to individual participants from Ratcliff and Rouder (1998). DIC and p_D are also reported for each model.

Participant	Model	a_{acc}	a_{spd}	B_{acc} / $s_{z_{acc}}$	B_{spd} / $s_{z_{spd}}$	η	T_{er}	s_t	DIC	p_D
JF	Diffusion	.256	.061	.066	.006	.155	.245	.181	**-3478**	11.93
	LBA	.603	.215	.373	.116	.263	.107	-	-2293	**11.59**
KR	Diffusion	.249	.065	.023	.015	.153	.227	.152	**-3793**	**10.05**
	LBA	.615	.223	.383	.143	.341	.123	-	-1327	12.79
NH	Diffusion	.246	.086	.078	.003	.213	.259	.172	**-5938**	11.85
	LBA	.479	.251	.27	.121	.307	.129	-	-4870	**11.15**

was the case in the simulated example the decrease between the nominal and effective number of model parameters due to functional form complexity was greater for the diffusion (-2.7 on average) than the LBA (-1.2 on average). Overall, when applied to real data coming from a more complicated design, the diffusion model tended to require fewer “effective parameters” (11.3 on average) than the LBA model (11.8 on average). At the level of individual participants, however, we see that p_D was smaller for the LBA than the diffusion model for two out of three participants.

Discussion

DIC is a model selection criterion which attempts to select the model which is best able to predict new data. DIC, and p_D, a measure of model complexity, can be calculated from MCMC samples from the deviance of posterior parameter distributions. The p_D measure takes into account functional form complexity, and can be thought of as the effective number of parameters used to fit the data. When using data simulated from the diffusion model with a simple experimental design, the diffusion model, perhaps surprisingly, had a smaller p_D value than the LBA model. In other words, for our simulated data set the diffusion model was simpler than the LBA in terms of functional form complexity. When the models were fit to benchmark data from Ratcliff and Rouder (1998) which model was simpler differed between participants. For two out of three participants the LBA required fewer effective parameters. Averaging over participants, however, suggested the diffusion model was simpler.

There are a number of technical details associated with DIC and p_D should be addressed. Spiegelhalter et al. (2002) state that DIC and p_D are appropriate when: the distribution of posterior samples are approximately normal, and the model provides a reasonable account of the data. We have already addressed the second point, i.e. the posterior parameters were providing good predictions of data. In the models presented here the posterior distributions for each parameter closely approximate normal distributions, making it more likely that the joint distribution of these parameters are also approximately normally distributed. DIC and p_D are also dependent on the prior distribution used and the “focus” of our analysis. We have made an attempt to make these factors equivalent across models.

First, we used numerical integration of the Winbugs results for the diffusion model in order to equate the focus of inference for each model. The WinBUGS code given by Vandekerckhove et al. (2008) for the diffusion model implements start point variability and non-decision time variabilty hierachically –that is, by drawing a sample for each of these parameters for each trial performed by a participant on each MCMC iteration. This approch was necessitated because the Ratliff diffuison does not have an analytic likelihood when these sources of between-trial variability are included. In contrast, the WinBUGS code takes advantage of the LBA’s mathematical simplicity by using an analytic expression for the likelihood of the LBA model which integrates out all forms of between-trial variability. This difference makes the deviances for each model produced by WinBUGS incommensurate; for the diffusion model this deviance focuses on the particular set of trials observed, whereas for the LBA the deviance is appropriate for the population of possible trials, and hence prediciton of performance by each subject performing new trials. As the latter focus is clearly more appropriate for our purposes we numerically integrated the deviance for each diffusion model posterior sample and used these integrated deviances to calcualte DIC and p_D.

Second, the prior distributions for diffusion model parameters are based on the range of parameter values estimated from all of the published diffusion fits found by Matze and Wagenmakers (submitted). Priors for LBA parameters were obtained from simulations which took the range of diffusion model parameters from Matze and Wagenmakers (submitted) and mapped them onto changes in LBA parameters. This gave a range of LBA parameters to be used as priors which may account for approximately the same range of patterns of data. In both cases the prior distribution of parameters was assumed uniform within these ranges. These prirors are informative not only in excluding parameters outside the allowed range, but also because the width of the range of allowed parameters determines the contribution made by the prior to the posterior deviance. A narrower range reduces posterior deviance and hence improves DIC. The large sample sizes that we examined means that the contribution of the prior is dominated by the likelihood of the data when determining parameter estimates within a model. However, this does not necessarily mean that differences in the prior for each model are not influential on the *difference* in posteior deviance between models, and hence DIC. In ongoing work we are implementing “vague” priors (i.e., priors with approximately equal probability across a very broad range of parameters for both models) in order to test the sensitivity of our results to the prior specification.

In summary, we have provided a relatively preliminary investigation into the complexity of models of choice and response time using a Bayesian model selection criterion. The criterion, DIC, and an associated measure of model complexity that takes into account differences in funcitonal form, p_D, are relatively easy to apply becasue it can be directly calculated based on MCMC samples from posterior model parameter distributions. If we consider simplicity as the range of potential data patterns which a model can predict, our results suggest that it may have been premature to claim that the LBA is the simplest model of choice and response time. Our results suggest that for these models a simple count of parameters will not suffice, and that more investigation is required. Functional form complexity based on prediction, however, is not the only aspect which might define a model’s simplicity. For example, the mathematical tractability of the LBA, which enables analytic likelihoods to be derived, make it possible to more estimate parameters from data using even quite basic software, such as Microsoft Excel (Donkin, Averell, et al., 2009).

Although DIC has been found to be reliable (e.g. Myung et al., 2005), there are alternative approches to defining functional form complexity. For example, both DIC and Bayes factors adjust for complexity, but DIC emphasizes posterior prediciton whereas Bayes factors emphasize the selection of a true model. Different approches have different strengths and weaknesses. For example, DIC, like AIC, is inconsistent, so that as sample size increases it tends to select overly complex models. Bayes factors are less attractive in terms of prediciton becasue they asses the degree to which the prior rather than posterior predicts new data (Liu & Aitkin, 2008). As part of a larger project we are investigating the degree to which conclusions about complexity are robust over a range of such model selection measures (Myung & Pitt, 1997; Myung, 2000).

Acknowledgments

We acknowledge support from the Keats Endowment Fund.

References

Akaike, H. (1973). Information theory as an extension of the maximum likelihood principle. In B. N. Petrov & F. Csaki (Eds.), *Second international symposium on information theory* (pp. 267–281). Budapest: Akademiai Kiado.

Brown, S., & Heathcote, A. J. (2008). The simplest complete model of choice reaction time: Linear ballistic accumulation. *Cognitive Psychology, 57*, 153-178.

Donkin, C., Averell, L., Brown, S., & Heathcote, A. (2009). Getting more from accuracy and response time data: Methods for fitting the linear ballistic accumulator. *Manuscript Submitted for Publication*.

Donkin, C., Brown, S., & Heathcote, A. (2009). The overconstraint of response time models. *Manuscript Submitted for Publication*.

Forstmann, B. U., Dutilh, G., Brown, S., Neumann, J., von Cramon, D. Y., Ridderinkhof, K. R., et al. (2008). The striatum facilitates decision-making under time pressure. *Proceedings of the National Academy of Science*, *105*, 17538–17542.

Ho, T., Brown, S., & Serences, J. (submitted). Domain general mechanisms of perceptual decision making in human cortex. *Journal of Neuroscience*.

Liu, C. C., & Aitkin, M. (2008). Bayes factors: prior sensitivity and model generalizability. *Journal of Mathematical Psychology*, *52*, 362–375.

Matze, D., & Wagenmakers, E. J. (submitted). Psychological interpretation of exgaussian and shifted wald parameters: A diffusion model analysis. *Manuscript submitted for publications*.

Myung, I. J. (2000). The importance of complexity in model selection. *Journal of Mathematical Psychology*, *44*, 190–204.

Myung, I. J., Karabatsos, G., & Iverson, G. J. (2005). A bayesian approach to testing decision making axioms. *Journal of Mathematical Psychology*, *49*, 205–225.

Myung, I. J., & Pitt, M. A. (1997). Applying Occam’s razor in modeling cognition: A Bayesian approach. *Psychonomic Bulletin & Review*, *4*, 79–95.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*, 59–108.

Ratcliff, R., Gomez, P., & McKoon, G. (2004). Diffusion model account of lexical decision. *Psychological Review*, *111*, 159–182.

Ratcliff, R., & Rouder, J. N. (1998). Modeling response times for two–choice decisions. *Psychological Science*, *9*, 347–356.

Ratcliff, R., Segraves, M., & Cherian, A. (2003). A comparison of macaque behavior and superior colliculus neuronal activity to predictions from models of simple two–choice decisions. *Journal of Neurophysiology*, *90*, 1392–1407.

Ratcliff, R., Thapar, A., & McKoon, G. (2004). A diffusion model analysis of the effects of aging on recognition memory. *Journal of Memory and Language*, *50*, 408–424.

Ratcliff, R., & Tuerlinckx, F. (2002). Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review*, *9*, 438–481.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society B*, *64*, 583–639.

Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2008). Hierarchical Bayesian diffusion models for two–choice response times. *Manuscript submitted for publication*.

Modeling the confidence of predictions: A Time Based Approach

Uwe Drewitz (uwe.drewitz@tu-berlin.de)
Department of Cognitive Psychology and Cognitive Ergonomics, Berlin University of Technology,
Franklinstr. 5-7, 10587 Berlin, Germany

Manfred Thüring (manfred.thuering@tu-berlin.de)
Department of Cognitive Psychology and Cognitive Ergonomics, Berlin University of Technology,
Franklinstr. 5-7, 10587 Berlin, Germany

Abstract

Everyday life demands explanations and predictions from everybody all the time. Using experience based knowledge, the human mind is well suited to draw the required causal inferences. However, due to failures in the past, such inferences are usually drawn under uncertainty and come along with different degrees of confidence. We present an ACT-R model describing the cognitive processes of induction and deduction for a prediction task in a simple, simulated technical environment. While ACT-R provides excellent mechanisms to capture causal learning and causal inferences, no process has been defined yet to account for the trust humans put in their predictions. Based on the availability heuristic by Tversky and Kahneman (1973), we propose an approach for modeling different levels of trust by using a temporal module from Taatgen, van Rijn and Anderson (2007), thus relating availability to retrieval time and confidence judgments. The forecasts of our model are compared with the results of an empirical study and nicely fit the experimental data.

Keywords: causal models; uncertainty; inductive learning; availability heuristic, temporal module; time estimation.

Introduction

The explanation of a current state of the world by events in the past and the prediction of future events from a present situation are fundamental qualities of human cognition. We follow the assumption proposed by many others that such reasoning processes are based on causal models (e.g., Waldmann, 1996) and proceeded under uncertainty (e.g. Einhorn & Hogarth, 1982). Two factors determine how much trust we put in an explanation or a prediction.

The first factor is the perceived amount of missing information in a given situation. This case applies when a causal model demands more data than currently available. Experiments by Thüring and Jungermann (1992) as well as Jungermann and Thüring (1993) demonstrated that such situations appear as ambiguous and lead to a reduction of confidence people have in their causal inferences.

The second factor is not an attribute of the situation, but of the causal model itself. Causal models – as any other kind of mental model – may be incomplete or even incorrect (Norman, 1983), hence leading to faulty conclusions.

Obviously, deficient models are not trustworthy. Confidence requires success, i.e., "…it's the *model's ability to make accurate predictions that is the ultimate measure of the model's value"* (Chown 2006, p. 69). This value can be characterized as the reliability of the model. To summarize, the ambiguity of the situation at hand and the reliability of the causal model currently employed determine the strength of confidence we have in the conclusions we draw. If we want to predict this confidence, we require a formal basis for modeling the influence of both factors. In the former studies by Thüring and Jungermann, rule-based systems served as such a basis and were used to describe the structure of a causal model. This approach was well suited to characterize ambiguous situations by the degree of matching between data and the conditional parts of the rules and to predict the content and confidence of causal inferences drawn from them. The reliability of a model, on the other hand, proved as more complicated to handle. Especially when we tried to describe how rules are formed in the course of inductive learning and which psychological mechanisms influence the confidence of causal judgments based on such rules "under construction", it became apparent that a comprehensive cognitive framework is needed to cope with the complexity of the matter.

The cognitive architecture ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere & Qin, 2004) provides such framework. We will use it to demonstrate how simple rule-based causal models can be built from induction and how predictions can be derived from such models. Special emphasis will be placed on the issue of how the success (respectively failure) of predictions in the course of learning influence the reliability of the rules and the confidence people place in their inferences.

Modeling Objectives

To model induction, predictions and confidence, three basic objectives must be achieved.

(i) To ensure inductive learning, not only the current situation must be represented in the ACT-R model, but preceding situations must be accounted for as well. In addition, the success or failure in coping with these situations must be captured. (ii) The ACT-R model must be able to make predictions. A prediction can be characterized as a statement about a future state of the world in terms of specific propositions. Since predictions are made under uncertainty, the ACT-R model must be able to combine a propositional content with a degree of confidence. To achieve this, reliability as well as ambiguity must be considered by the ACT-R model (although the latter is not emphasized here). (iii) In case of incorrect predictions, the ACT-R model must provide mechanisms to modify the causal knowledge structure if new evidence is available. To put the objectives into practice and to implement an ACT-R model with the ability to generate predictions with different degrees of confidence, we have to refer to experimental data.

The Experiment

The empirical basis of our approach are data obtained in an experiment by Thüring, Drewitz and Urbas (2006) that tested the following assumption: When a causal model is induced from observations, inferences *deduced* from that model are usually probabilistic and their uncertainty is influenced by the observer's former experience with the model. The results of this experiment were extensively discussed in Thüring et al. (2006) to clarify the interplay of induction, deduction and confidence judgments.

In the experimental task, the participants had to acquire the causal model of a technical system, i.e., the cooling system of a power plant. The system could run properly (state OK) or not (state MALFUNCTION) and consisted of four pumping devices (subsystems A, B, C and D). Information about the subsystems was displayed on four dials (Fig. 1) which could be turned on (A, D) or off (B, C). Each dial represented the state of a subsystem that was either 'up' (A), 'down' (D), or 'unknown' because its dial was switched off (B, C). While each of the factors A, B and C was causally relevant at some point of the experiment, factor D was a random variable serving as a distractor, which was introduced to obtain a sufficient level of task complexity. In each trial, participants were shown a combination of dials as in the left part of figure 1. Based on this information, they first predicted the state of the overall system by pressing one of two buttons 'OK' or 'MALFUNCTION', and then rated their confidence by adjusting a slider. After submitting their confidence rating, a status message informed them about the correct system state as shown in the right part of figure 1.

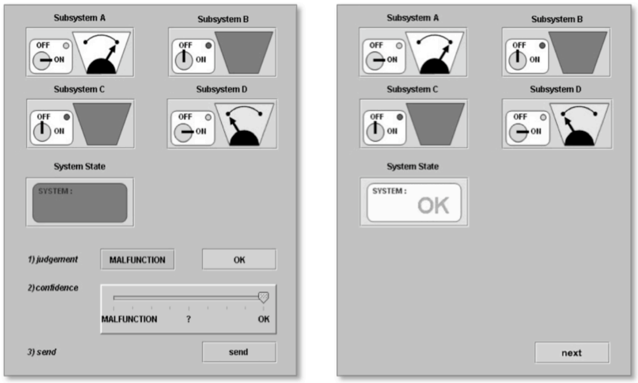


Figure 1: Screen layout of the experiment.

Using the feedback they received in each trial, participants could gradually develop a causal model representing the relation between the state of the subsystems (A, B, C, D) and the state of the entire system (OK or MALFUNCTION). In the first phase of our study, a simple model was induced in which just the proper functioning of one subsystem (e.g., A) was required for the faultless running of the cooling. Our participants learned this model fairly quickly from the data. In figure 8, the curve labeled "human" shows their mean confidence ratings (transformed into percentage values). Data points in the upper half of the figure represent ratings for the prediction "OK", those in the lower part for the prediction "MALFUNCTION". Note that the ratings start well above zero, because three trials in which A was coupled to OK were used in advance to acquaint the participants with the experimental setting. Starting from there participants soon reached a high and stable level of confidence (i.e., mean values between 70% and 80% with some exceptions due to the random condition D). At the end of this learning phase, information was provided which reduced the reliability of the model, i.e., in the trials 26-31 the feedback was contrary to the initial system behavior. Consequently, our participants' confidence in their predictions dramatically decreased and some of them even predicted a state contradictory to the rule they had learned before.

In the second phase of the experiment, information was provided that allowed for expanding the simple *'mono causal'* model into a more extensive one. This was either an '*or*-model' capturing multiple alternative causes each of them being sufficient for the effect, or an '*and*-model' representing a conjunction of several causal conditions each of them being necessary for the effect. When the new model was reinforced over several trials, confidence ratings raised to a level similar to the one of the mono causal model at the beginning (see fig. 8 and 9). When the reliability of these models was reduced (trials 31-35 and 45-49), the same effects occurred as in phase one, i.e., confidence ratings dropped again.

According to our first objective, the ACT-R model must be able to capture the cognitive processes of knowledge acquisition in this experiment, which are distinguished by the fact that people revise and expand their causal model when new facts become available.

Knowledge Acquisition

We propose three mechanisms of knowledge acquisition complementing each other, with each of them being necessary to form and diversify a causal model.

Inductive Learning

The first mechanism can be characterized as inductive learning. Within their natural environments, people make observations and store them in memory. Observing the same constellation of events repeatedly strengthens their associative relation in the memory trace. Thus, rudimentary causal models are constituted that guide further observations. In our experiment, these models could be described in terms of simple rules such as "if A is up then the system is OK" or "if A is down then a MALFUNCTION occurs".

Deductive Reasoning

Inductive learning is closely related to deductive reasoning. When a rule has been formed via induction, its reliability is tested via deduction, thus creating a circle in which these two mechanisms take turns in forming a causal model. In each deduction, available data are matched with the rules and a conclusion is drawn. Those rules, which have been reliable in the past, are chosen over less reliable ones. In the first phase of our experiment, the rule "if A is up then the system is OK" produced a correct prediction whenever A was up, while the rule "if D is up then the system is OK" did not, because the relation between D and the system state was random.

Though reliable rules should be chosen most frequently, less reliable ones can get a chance when their conditions are matched by the current data. When this happens, the confidence that is placed in the prediction should be less compared to the confidence in a prediction derived from a reliable rule. For example, the confidence in predicting a well functioning system when "D is up" should be lower compared to a situation when "A is up".

To summarize, reliability serves two purposes. It determines which rules are chosen over others and it tunes the confidence people place in their predictions. Both these functions must be implemented in ACT-R to explain the data of our experiment and to achieve the second objective stated above, i.e., the derivation of the propositional content of a prediction in combination with a specific degree of confidence based on the experienced reliability of the model.

Rule Revision

While the reduction of confidence placed in a prediction is one consequence of the failure of a rule, the revision of the rule itself is another one. Changing the content or the structure of a rule is the third mechanism required to describe the forming of a causal model. Revisions only make sense in the light of new evidence, i.e., when the failure of a rule coincides with the observation of new conditions that must be satisfied in addition to (or instead of) the conditions that have been accounted for so far. In this case, the rule in question is altered. In our experiment, this happened in the second phase where simple mono causal models where expanded to an "or-model" or an "and-model". To attain our third objective, such changes must be accounted for when causal models are developed in ACT-R.

Overview of the ACT-R Model

The three mechanisms were implemented in the framework provided by ACT-R 6.0. Figure 2 displays the cyclic concept we used to establish the cognitive flow of control for performing the successive trials in our experiment. The nodes represent different control states, whereas the directed links indicate possible transitions between them.

At the START of each experimental trial, the current situation is stored in an ACT-R buffer. This situation consists of the states ("up" or "down") of the four components (A to D) of the cooling system. The task is to predict if these states will entail a proper functioning or a malfunction of the system. The next step is to SEARCH for instances in declarative memory matching the situation at hand. In our model, each search in memory relies on the spreading activation mechanism and is affected by noise resulting from the according parameter in ACT-R. Those instances, that have been frequently used in former cycles, have a higher activation and hence a higher probability to be found. Two outcomes are possible at this stage. (i) If a match is made, the according instance is retrieved. Now, a first propositional content for the required prediction has been found, since the instance contains the effect (OK or MALFUNCTION) that this specific constellation of A to D has produced in the past. To account for previous experiences with the prediction, its content is linked to an appraisal value. The appraisal is "good" when former predictions were correct, but "bad" when mistakes were made in the past. (ii) If no match is made, the model switches to "GUESS". This is the case either when the current situation is new, or if the activation of no instance in declarative memory is high enough for a successful retrieval. Guessing means that one of the two outcomes "OK" or "MALFUNCTION" is chosen at random from declarative memory. Therefore, in case (i) as well as in case (ii), the result at this stage is a first propositional content for the required prediction enhanced by an appraisal value.

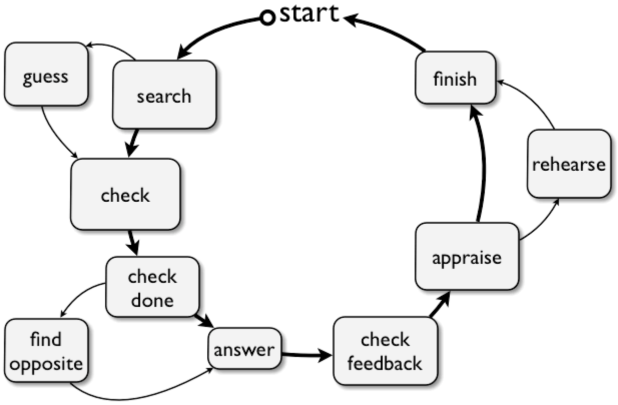


Figure 2: Cognitive flow of control of the ACT-R model.

In the next step, the content is CHECKED against different experiences made in the past. Three alternatives are possible: (i) If an instance with "good appraisal" was found during SEARCH, the check looks for an instance with the same effect but a "bad appraisal". (ii) If an instance with "bad appraisal" was found during SEARCH, the check looks for an instance with the same effect but a "good appraisal". (iii) If the result of GUESSING was "OK", then the alternative effect "MALFUNCTION" is produced during memory search, and vice versa for the result of "MALFUNCTION". The idea underlying this stage is twofold. First, it mimics reasoning under uncertainty where inferences are compared to other possibilities. Second, the cognitive processes involved here produce different retrieval times that are used to model different degrees of confidence. How this is achieved will be described later.

When the CHECK has been accomplished, the model switches to CHECK DONE. Now, if the appraisal of the instance retrieved during SEARCH is "bad", the preliminary propositional content is not reliable. In this case, the process FIND OPPOSITE generates the opposite effect as alternative prediction and uses it as ANSWER. Otherwise, no alternative prediction is required and the result of the former SEARCH is delivered as ANSWER. In our experiment, this is the point where subjects make their prediction and then receive a feedback.

In the ACT-R model, the FEEDBACK is CHECKED by comparing it with the prediction. If the prediction is correct, APPRAISE generates the appraisal value "good" and links it to the according instance. In case of a wrong prediction, however, the appraisal turns to "bad" and hence the instance represents an incorrect prediction.

If a successful prediction is made based on GUESSING or on TAKING THE OPPOSITE, this new information is RE-HEARSED to strengthen the activation of this valuable new insight. For the same reason, REHEARSAL occurs when a formerly reliable instance produces a wrong prediction.

When the state FINISH is reached, all buffers are cleared and the results are transferred to declarative knowledge. A result consists of a new instance, whenever an unprecedented constellation was encountered in that cycle and used for a prediction. In this way, declarative knowledge is extended and revised.

So far, we have described a circular process of knowledge acquisition consisting of inductive learning, deduction and rule revision. Figure 3 shows the predictions made by the ACT-R model (over 21 runs) compared to the predictions made by the participants in the experiment by Thüring et al. (2006). As indicated in the chart, there is a very good fit between both types of predictions.

To fulfill our second objective, these results must be related to the generation of confidence judgments. Reliable causal rules are represented by instances with a positive appraisal. Among these instances, those with a high activation constitute a person's actual causal model. The amount of activation not only determines which rules are used for prediction, but should also influence the confidence people have in their predictions.

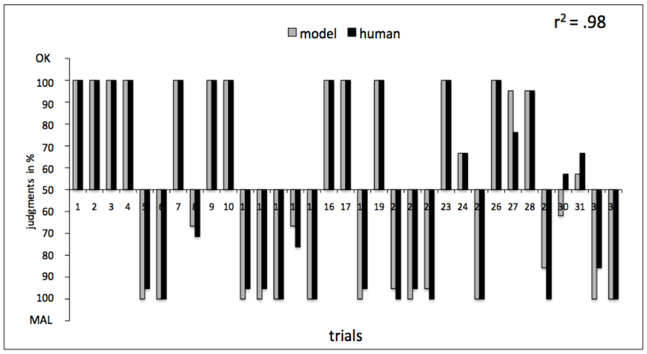


Figure 3: Mean propositional judgments (n=21).

However, since activation is a subsymbolic parameter, it cannot be directly used to produce confidence judgments. To solve this problem, we adopt a heuristic proposed by Tversky and Kahneman (1973) to our ACT-R model.

The Availability Heuristic: Degree of Confidence and Retrieval Time

When people have to evaluate the frequency or likelihood of an event, they often use heuristics to do so. In case of applying the availability heuristic, the subjective probability of an event depends on how fast the representation of a former occurrence of the event can be retrieved from memory, i.e., the faster the retrieval of the event, the higher its estimated probability. Tversky and Kahnman (1973) assumed that the ease of retrieval is equivalent to the perceived time of retrieval. This offers an interesting solution for the problem of modeling the confidence of predictions. The retrieval of an instance raises its overall activation, which in turn lowers its retrieval time and hence should increase the confidence in its propositional content. Within ACT-R, the perception of time can be captured by a temporal module that was developed by Taatgen, van Rijn and Anderson (2007), especially for estimating short times.

Estimating Time with the Temporal Module

The temporal module consists of a pacemaker and its relations to a temporal buffer (see fig. 4). Three different parameters can be set to influence time estimation within this framework (Taatgen et al., 2007). One of them is the *time-master-start-increment*. This parameter has to be set at a low level to make the module sensitive enough for estimating short durations, such as retrieval times.

When time measuring begins, a start signal is created which causes the pacemaker to generate time pulses, so-called ticks. These ticks are collected in the accumulator of the temporal buffer. When a time estimation is needed, the number of ticks that have been accumulated between the temporal request and the retrieval represents the elapsed time.

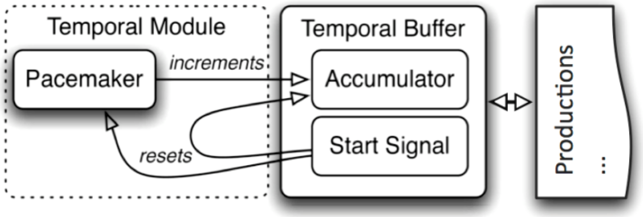
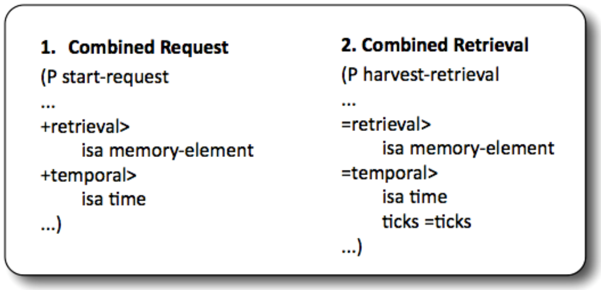


Figure 4: The temporal module (taken and adapted from Taatgen et. al., 2007).

In our approach, time estimation is always related to the retrieval of a specific memory element, such as an instance. Therefore, any temporal request is combined with the request for a memory element, and the analog holds for the retrieval. The ACT-R syntax implementing the combined request and retrieval is shown in figure 5.

The result of a temporal retrieval is a symbolic value characterizing the perceived time for finding the memory element. This value can be processed further to generate different degrees of confidence.



```
1.  Combined Request          2. Combined Retrieval
(P start-request              (P harvest-retrieval
...                           ...
+retrieval>                   =retrieval>
      isa memory-element            isa memory-element
+temporal>                    =temporal>
      isa time                      isa time
...)                                ticks =ticks
                              ...)
```

Figure 5: Combined declarative and temporal request.

From Time to Confidence

We propose two different methods to transform perceived retrieval times into confidence judgments. Both are mathematical functions, which (at least for the time being) are not implemented within ACT-R itself.

Transforming retrieval time. The first method can be characterized as a direct implementation of the availability heuristic. It is expressed by the formula in figure 6. Two properties of this function are immediately salient: (i) Short retrieval times lead to high confidence values while long retrieval times cause low confidence judgments. (ii) Since the function is logarithmic, the decrease of confidence decelerates with the number of ticks increasing. This accounts for the observation that differences between longer retrieval times result in rather small differences for related confidence ratings and vice versa.

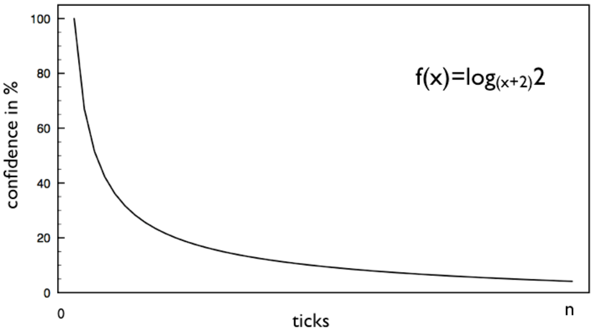


Figure 6: Transformation function ($f(x)=\log_{(x+2)}2$) for the transformation of *retrieval time* (schematically).

Figure 8 displays the confidence judgments for predicting the system states “OK” and “MALFUNCTION” that are generated by our model when this function is used. Although the match between the model and human data is good, a more sophisticated approach can be taken to model the confidence of predictions.

Transforming retrieval time differences. The idea underlying our second method is to check the retrieval time for an original prediction against the retrieval time for an alternative prediction. The alternative is an instance of the same content, but with an appraisal indicating that (at least once) the instance has failed to be successful. Due to its success in the past, the original predication is highly activated and can be retrieved fast. If the same holds for the alternative, the difference between the retrieval times of both predictions is small and the confidence in the original should be low. On the other hand, if the alternative prediction has been less successful than the original, its lower activation entails a longer retrieval time. In this case, the difference between the retrieval times of both predictions is large and the confidence in the original prediction should remain high. These relations between retrieval time and degree of confidence are captured by our second function. It accounts for the fact that we may find conflicting information of different value when we search our memory to make a prediction.

The difference of both retrieval times is calculated (as an absolute integer) and taken as input for the transformation process. Figure 7 shows the formula and form of the function used for this transformation.

Figure 9 presents the generated data by the model using the method of transforming time differences into confidence ratings.

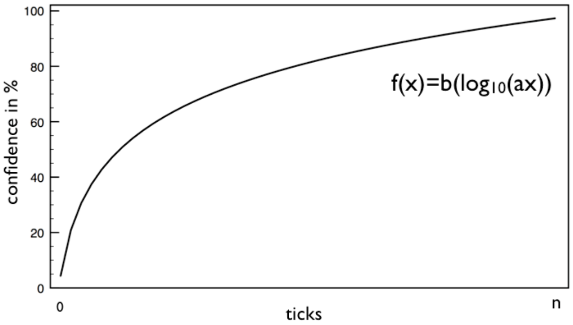


Figure 7: Transformation function ($f(x)=b*\log_{10}a*x$) for the transformation of *time differences* (schematically).

Discussion

The ACT-R model and the two functions described above were developed to account for the data of the first experimental block where a *‘mono causal model’* had to be learned. A comparison of the charts in figure 8 and 9 indicates that both functions are well suited to model confidence ratings based on time measures.

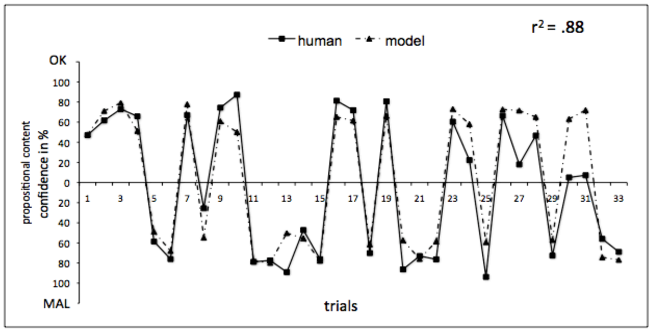


Figure 8: Combined ratings for the *‘mono causal’* block calculated with method I: transformation of retrieval time (n=21, RMSSD=4.3).

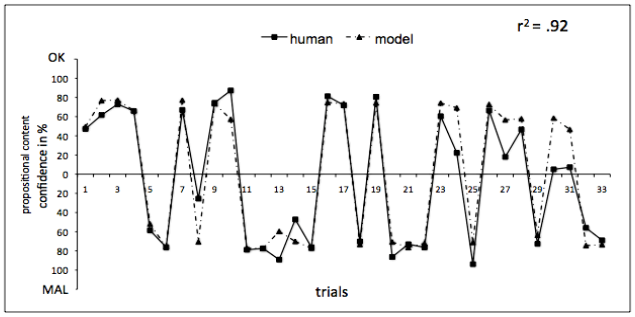


Figure 9: Combined ratings for the *'mono causal'* block calculated with method II: transformation of time differences (n=21, RMSSD=3.1).

Nevertheless, there is an advantage for the second function. The trend measure (r^2) and the goodness-of-fit measure (RMSSD) show a better fit with the empirical data for that method. Therefore, the second function was chosen to predict the confidence ratings in the second experimental block, where the *'and-model'* as well as the *'or-model'* were induced. Again, the model proved to be well applicable, matching the empirical data with a high fit (see fig. 10 and 11).

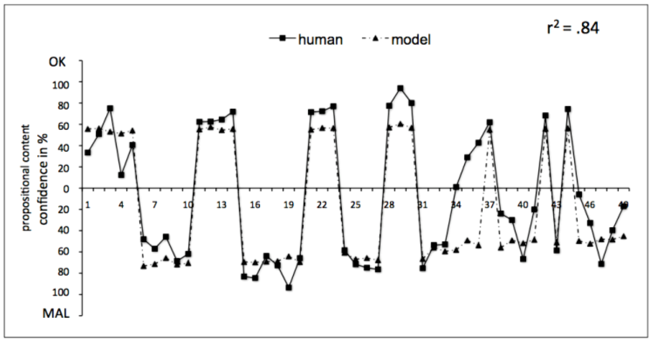


Figure 10: Combined ratings of propositional content and related confidence for the *'and'* block (n=21, RMSSD=4.3).

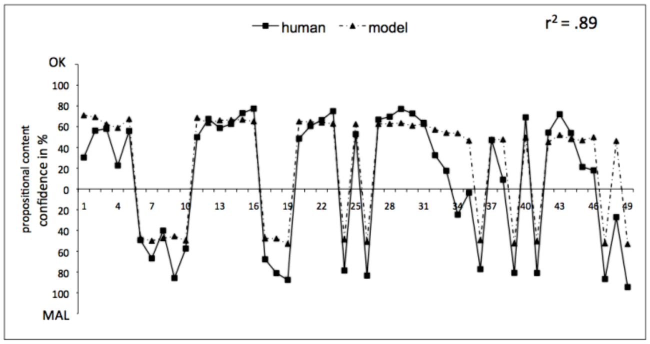


Figure 11: Combined ratings of propositional content and related confidence for the *'or'* block (n=21, RMSSD=3.8).

To summarize, we have proposed an ACT-R model, which combines inductive learning, deductive reasoning and mechanisms for revising knowledge structures to describe the acquisition of causal models. Predictions derived from a causal model are made under uncertainty, i.e., the propositional content of an inference is combined with a particular confidence. In order to describe different degrees of confidence, the availability heuristic proposed by Tversky and Kahneman (1973) was adopted to our ACT-R model. This was accomplished by using estimated retrieval times of memory elements to operationalize availability. The operationalization was achieved by two mathematical functions, which transform retrieval times into confidence judgments. The data generated by our ACT-R model in combination with these functions where compared to data generated by humans in an experiment reported by Thüring et al. (2006). As a result, the second function proved as slightly superior to the first one.

Future research will adress the problem of how this function can be implemented directly within the ACT-R framework. Moreover, our approach must be tested in further experiments adressing different situations of inductive learning as well as different domains of reasoning.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111 (4), 1036-1060.

Chown, E. (2004). Cognitive Modeling. In A. Tucker (Ed.), *Computer Science Handbook*. Chapman & Hall.

Einhorn, H. J. & Hogarth, R. M. (1982). Prediction, diagnosis, and causal thinking in forecasting. *Journal of Forecasting*, 1, 23-36.

Jungermann, H. & Thüring, M. (1993). Causal knowledge and the expression of uncertainty. In G. Strube & K. F. Wender (Eds.), *The cognitive psychology of knowledge*. Amsterdam: Elsevier.

Norman, D. A., (1983). Some Observations on Mental Models. In D. Gentner & A.L. Stevens (Eds), *Mental Models.* NJ: Hillsdale.

Taatgen, N. A., Rijn, H. v., & Anderson, J. R. (2007). An Integrated Theory of Prospective Time Interval Estimation: The Role of Cognition, Attention and Learning. *Psychological Review*, 114(3), 577-598.

Thüring, M., Drewitz, U., & Urbas, L. (2006). Inductive Learning, Uncertainty and the Acquisition of Causal Models. In R. Sun & N. Miyake (Eds.), *Proceedings of the 28th Annual Cognitive Science Society*. NJ: LEA.

Thüring, M. & Jungermann, H. (1992). Who will catch the Nagami Fever? Causal inferences and probability judgment in mental models of diseases. In D. A. Evans & V. L. Patel (Eds.), *Advanced models of cognition for medical training and practice* (pp. 307-325). Berlin: Springer.

Tversky, A. and D. Kahneman, 1973. Availability: A heuristic for judging frequency and probability. *Cognitive Psychology* (5), 207-32.

Waldmann, M., (1996). Knowledge-based causal induction. In D. R. Shanks, D. L. Medin & K. J. Holyoak (Eds), *The Psychology of Learning and Motivation (Vol. 34): Causal Learning.* San Diego, CA: Academic Press.

Caffeine's Effect on Appraisal and Mental Arithmetic Performance: A Cognitive Modeling Approach Tells Us More

Sue E. Kase (skase@ist.psu.edu), Frank E. Ritter (frank.ritter@psu.edu)
College of Information Sciences and Technology, Pennsylvania State University
University Park, PA 16802 USA

Michael Schoelles (schoem@rpi.edu)
Cognitive Science Department, Rensselaer Polytechnic Institute
Troy, NY 12180 USA

Abstract

A human subject experiment was conducted to investigate caffeine's effect on appraisal and performance of a mental serial subtraction task. Serial subtraction performance data was collected from three treatment groups: placebo, 200 mg caffeine, and 400 mg caffeine. Data were analyzed by average across treatment group and by challenge and threat task appraisal conditions. A cognitive model of the serial subtraction task was developed and fit to the human performance data. How the model's parameters change to fit the data suggest how cognition changes across treatments and due to appraisal. Overall, the cognitive modeling and optimization results suggest that the speed of vocalization is changed the most along with some changes to declarative memory. This approach promises to offer fine-grained knowledge about the effects of moderators on task performance.

Keywords: Caffeine, stress, task appraisal, cognitive arithmetic

Introduction

Caffeine is widely consumed throughout the world in beverages, foods, and as a drug for a variety of reasons, including its stimulant-like effects on mood and cognitive performance (for review see Fredholm et al., 1999). Its positive effects on performance, notably sustained vigilance and related cognitive functions, are well documented when administered to rested volunteers in doses equivalent to single servings of beverages (Amendola et al., 1998; Smith et al., 1999). Additionally, its consumption in moderate doses is associated with few, if any, adverse effects (Nawrot et al., 2003). Therefore, caffeine has been a strategy examined for its usefulness to military personnel (Lieberman & Tharion, 2002; McLellan et al., 2007).

The majority of caffeine research is conducted through human experimentation with analysis of the collected performance data. Few studies have attempted to *model* the effects of caffeine. One such study by Benitez et al. (2009) presented a biomathematical model for describing performance during extended wakefulness with the effect of caffeine as a stimulant.

Likewise, this study takes a modeling approach employing cognitive modeling and optimization techniques to investigate the effects of caffeine on cognitive performance. In particular, we examined the effects of caffeine and task appraisal during the arithmetic portion of the Trier Social Stress Test (TSST), a mental serial subtraction task. Based on human subject observations, self-reported appraisal, and performance data, we then developed a cognitive model in the ACT-R cognitive architecture of the serial subtraction task. Parametric solution sets resulting from optimizing the serial subtraction cognitive model to data from three treatment groups (placebo, 200 mg, 400 mg) and two task appraisal conditions (challenge and threat) provided the first cognitive modeling-derived insights on the cognitive effects of caffeine.

Method

This section begins with an overview of the human subject experiment where performance and task appraisal data were collected and later utilized in the development and optimization of a cognitive model. A detailed description of the cognitive task follows, as well as, the formulation of the self-reported appraisal conditions. Lastly, results and interpretations of the human performance data are suggested.

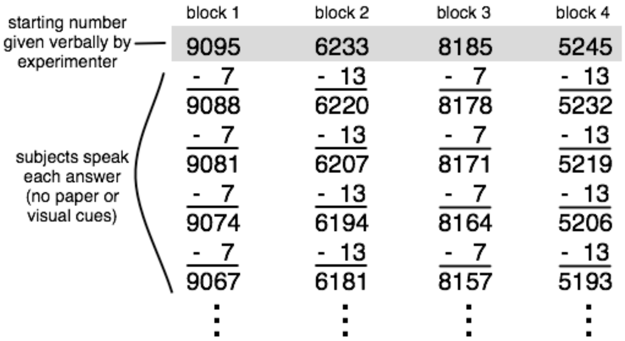
As part of a larger project, human subject data was collected to study the effects of stress and caffeine on cardiovascular health. The authors collaborated with Dr. Laura Klein and her lab in the Biobehavioral Health Department at Penn State University. A mixed experimental design was conducted with 45 healthy men 18-30 years of age (Klein, Whetzel, Bennett, Ritter, & Granger, 2006). (Men are typically used in these types of studies because we also took additional physiological measures and their systems are simpler.)

All subjects were asked to perform a series of three cognitive tasks. Subjects individually performed a simple reaction time (RT) and a working memory (WM) task taking 15 minutes to complete. Then subjects were administered one of three doses of caffeine: none (placebo), 200 mg caffeine (equivalent to 1-2, 8 oz cups of coffee), or 400 mg caffeine (equivalent to 3-4, 8 oz cups of coffee). After allowing absorption time, a 20-minute stress session of the mental arithmetic portion of the TSST was performed. Following completion of this stressor, subjects again were asked to complete the RT and WM tasks. Cognitive performance was determined by calculating accuracy and response time scores.

This paper focuses on one portion of the experiment—the TSST. The TSST protocol has been used for investigating psychobiological stress responses in a laboratory setting since the 1960s (Kirschbaum, Pirke, & Hellhammer, 1993). TSST traditionally consists of an anticipation period and a test period in which subjects have to deliver a free speech and perform mental arithmetic in front of an audience. The mental arithmetic portion of the TSST is a mental serial subtraction task.

Serial Subtraction Task

The serial subtraction task utilized in the experiment consisted of four 4-minute blocks of mentally subtracting by 7s and 13s from 4-digit starting numbers. Figure 1 illustrates the serial subtraction task. These were the four starting numbers used to begin the four blocks of subtraction during the experiment.


starting number given verbally by experimenter
subjects speak each answer (no paper or visual cues)

block 1	block 2	block 3	block 4
9095	6233	8185	5245
- 7	- 13	- 7	- 13
9088	6220	8178	5232
- 7	- 13	- 7	- 13
9081	6207	8171	5219
- 7	- 13	- 7	- 13
9074	6194	8164	5206
- 7	- 13	- 7	- 13
9067	6181	8157	5193
⋮	⋮	⋮	⋮

Figure 1: An illustration of the four blocks of the serial subtraction task as in the experiment.

Before the task begins the experimenter explains that the subject's performance is going to be voice recorded and reviewed by a panel of psychologists for comparison with the other subjects participating in the experiment. The task is performed mentally with no visual or paper clues. After the task is explained to the subject, a task appraisal questionnaire is completed, and the subject begins performing the task. It is thought that this anticipation period, for some subjects, increases anxiety and worry about poor performance on the upcoming task.

Subjects sit in a chair directly in front and near the experimenter who is holding a time keeping device and clipboard of the correct subtraction answers that she checks off as the subject performs the task. Before the task begins the experimenter emphasizes that the task should be preformed as quickly and as accurately as possible. An experimenter tells the subject the starting number; from then on, the subject speaks the answer to each subtraction problem. When an incorrect answer was given, the subject was told to "Start over at <the last correct number>". At two minutes into each 4-minute session, subjects were told that "two minutes remain, you need to go faster". This prompt enhances the time-pressure component of the task.

Task Appraisal

Before and after the serial subtraction stress session, subjects completed pre- and post-task appraisals based on Lazarus and Folkman's (1984) theory of stress and coping. Each subject was asked five questions orally: two focused on the subject's resources or reserves to deal with the serial subtraction task and three focused on the subject's perception as to how stressful the task would be.

For all questions the scale was from 1 to 5 with a value of 3 indicating that the subject is neither challenged nor threatened by the task. After correcting for the imbalance in questions, a ratio of perceived stress to perceived coping resources was created. For example, if a subject's total appraisal score was 1.5 or less, their perceived stress was less than or equal to their perceived ability to cope, which equated to a *challenge condition*. If a subject's appraisal score was greater than 1.5, their perceived stress was greater than their perceived ability to cope, which equated to a *threat condition*.

Each treatment group was composed of 15 subjects. The placebo group had approximately the same number of subjects in each appraisal condition (7 challenge, 8 threat). The 200 mg caffeine group had twice as many challenged subjects as threatened subjects (10 challenge, 5 threat). The 400 mg caffeine group contained only 2 challenged subjects with the remainder (13) subjects reporting a threatening appraisal.

Results and Discussion

For this investigation, the serial subtraction performance data from the placebo group (PLAC), the 200 mg caffeine group (LoCAF), and the 400 mg caffeine group (HiCAF), were analyzed by average across treatment group and by appraisal condition. The performance statistics of primary interest were number of attempted subtraction problems and a percentage correct score. The data are shown in Table 1 where each pair of values represents number of attempts and percent correct. The results discussed in this paper apply to data from the first block of subtracting by 7s.

Table 1: Human performance (average number of attempts and percent correct) by treatment group (each N=15) and appraisal condition (challenge, threat).

Treatment	Average	Challenge	Threat
PLAC	47.3, 81.5	50.7, 83.3	40.4, 77.9
LoCAF	59.1, 86.5	62.4, 88.3	37.5, 74.8
HiCAF	45.7, 79.2	51.6, 82.8	38.9, 75.1

For all treatment groups the challenge condition showed the best performance in both number of attempts and percent correct over the average across treatment and the threat condition. The threat condition showed the worst performance. Performance differences between the challenge and threat conditions were most pronounced in the LoCAF group with an impressive increase of nearly 25 more attempted subtraction problems and a 13.5% increase in subtraction accuracy by challenged subjects over threatened subjects. For the HiCAF group the challenge and threat condition differences were less than LoCAF but still substantial: 13 more attempted problems and a 7.7% increase in subtraction accuracy. Differences between the challenge and threat condition were least visible in the PLAC group, 10 more attempted problems and only a 5.4% increase in accuracy.

Figure 2 better illustrates these performance differences with the treatment groups labeled along the x-axis and the plot subdivided into three sections: averages across treatment groups (not by appraisal condition) in the leftmost section, and averages across treatment groups subdivided by appraisal condition in the center (challenge) and rightmost sections (threat).

The plot visualizes several interesting trends; some supported by existing caffeine and cognition research and others not. In the average across treatments plot (leftmost section), the performance of the HiCAF group drops below that of PLAC for both performance statistics. This supports findings that large doses of caffeine are occasionally associated with anxiety and disrupt performance (Haishman, & Henningfield, 1992; Wesensten, Belenky, & Kautz, 2002). Whether a 400 mg dose is considered 'large' may be in question as some studies administered up to 800 mg doses (McLellian et al., 2007). Generally, 100 to 300 mg doses are categorized as 'low' dosages because 50-300 mg of caffeine is available in a number of forms including tablets, chewing gum, a wide variety of beverages and some food products.

In the challenge condition (middle section), HiCAF performance does not drop below PLAC, but is approximately equivalent or slightly higher. In both the average across treatments and the challenge condition, LoCAF performance is well above that of PLAC. This is also supported in previous research that low doses of caffeine tend to increase performance (Amendola et al., 1998; Smith et al., 1999). In both these cases, the across treatments and challenge plots, the effects of caffeine take on characteristics related to level of arousal studies (i.e., Anderson & Revelle, 1982) and appear to follow the Yerkes-Dodson (1908) law that postulates that the relationship between arousal and performance follows an inverted U-shape curve.

There is no supporting research for the performance trends visible under the threat condition (right section). Threatened subjects self-reported stress and lack of coping skills to adequately perform the serial subtraction task. The threat plot shows performance decreases from PLAC to LoCAF (instead of increases as observed in the other sections of the plot) with HiCAF only very slightly higher than LoCAF (+1.4 attempts, and +0.3% correct). In this case, the U-shape is not inverted, but actually very slightly U-shaped.

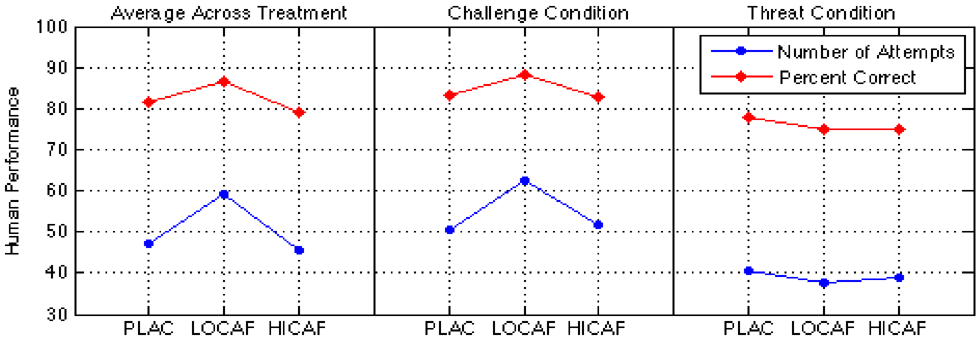


Figure 2: Comparing human performance differences in number of attempts and percent correct by treatment group (x-axis) and appraisal condition: treatment groups not accounting for appraisal (leftmost section), and averages across treatment groups divided by appraisal condition, challenge (middle section) and threat (rightmost section).

More can be discussed about the human performance data by way of analysis and interpretation of caffeine's effect on appraisal and serial subtraction. However, a more important question remains: Can these effects be modeled using a cognitive architecture and what might be learned from the parameters and values generating best fits during optimization of the model?

Modeling Serial Subtraction

Theory about how mental arithmetic is performed combined with observations gathered during the human subjects' performance of serial subtraction laid the foundation for the development of a cognitive model of the serial subtraction task. The ACT-R cognitive architecture (Anderson, 2007) was chosen to model the serial subtraction task for several reasons: it provides a parameter-driven subsymbolic level of processing; it permits the parallel execution of the verbal system with the control and memory systems, and it has been used for other models of addition and subtraction developed by other researchers.

The serial subtraction model performs a block of subtracting by 7s or 13s in a similar manner to that of the human subjects. The model's declarative knowledge consists of arithmetic facts and goal-related information. The model's procedural knowledge is production rules that allow for retrieval of subtraction and comparison facts necessary to produce an appropriate answer. The model performs subtractions by column-by-column.

The model runs under ACT-R 6.0 and utilizes the imaginal module and buffer. The imaginal buffer implements a problem representation capability. In the serial subtraction model the imaginal buffer holds the current 4-digit number being operated on (the minuend) and the number being subtracted (the subtrahend). The goal module and buffer implement control of task execution by manipulation of a state slot. ACT-R's vocal module and buffer verbalize the answer to each subtraction problem as the subjects do.

The model starts with the main goal to perform a subtraction and a borrow goal to perform the borrow operation when needed. Both types of goal chunks contain a state slot, the current column indicator, and the current subtrahend. The current problem is maintained in the imaginal buffer. This buffer is updated as the subtraction problem is being solved. The model begins with an integer minuend of 4-digits. All numbers in the model are chunks of type integer with a slot that holds the number. The model also contains subtraction and addition fact chunks whose slots are the integer chunks described above. This representation of the integers and arithmetic facts has been used in other ACT-R arithmetic models.

The model determines if a borrow operation is required by trying to retrieve a comparison fact that has two slots, a greater slot containing the minuend and a lesser slot containing the subtrahend. If the fact is successfully retrieved then no borrow is necessary, otherwise a borrow subgoal is created and executed. Borrowing is performed by retrieving the addition fact that represents adding ten to the minuend. The subtraction fact with the larger minuend is retrieved. The model then moves right one column by retrieving a next-column fact using the current column value as a cue. If this retrieval fails, there are no more columns so the borrow and the subgoal return back to the main task goal. If there is a next column and its value is not 0 than 1 is subtracted from it by retrieval of a subtraction fact. If the value is 0 then the problem is rewritten in the imaginal buffer with a 9 and the model moves to the next column and repeats the steps discussed above, returning to the main task when there are no more columns.

The model outputs the answer by speaking the 4-digit result. The model has two output strategies. For this paper the data reported are for the calc-and-speak strategy where the model speaks the answer in parallel with the calculation described above. If the answer is incorrect, the problem is reset to the last correct answer. If the answer is correct, the main problem task is rewritten in the imaginal buffer.

After the model has performed a block of subtractions the number of attempted subtraction problems and percent correct, are recorded. The model's performance can be adjusted by varying the values of architectural parameters associated with specific modules and buffers, and subsymbolic processes within the architecture.

Optimizing to Human Data

How does cognition change under stress and caffeine? We can explore this question by adjusting theoretically motivated parameters in architecture. The parameters that lead to better correspondences suggest how cognition changes. This section begins by discussing the architectural parameters selected for adjusting the model's performance to simulate the human data. This process of *fitting* the cognitive model to human data is a form of optimization. The optimization approach to fit the model is briefly described in the second part of the section. The optimization results, accompanied by interpretations of best fitting parameter values, is discussed at the end of the section.

Architectural Parameters

Three ACT-R architectural parameters appeared important in performing serial subtraction and were selected for adjusting the model's performance: seconds-per-syllable, base level constant, and activation noise. The rate the model speaks is controlled by the seconds-per-syllable parameter (SYL). The ACT-R default timing for speech is 0.15 seconds per assumed syllable based on the length of the text string to speak. There is a default of three characters per syllable controlled by the characters-per-syllable parameter. The seconds-per-syllable and characters-per-syllable parameters control subsymbolic processes in ACT-R's vocal module. The vocal module gives ACT-R a rudimentary ability to speak. It is not designed to provide a sophisticated simulation of human speech production, but to allow ACT-R to speak words and short phrases for simulating verbal responses in experiments such as the answers to the subtraction problems.

The other two parameters affect declarative knowledge access: the base level constant (BLC), and the activation noise parameter (ANS). The BLC parameter and a decay parameter affect declarative memory retrieval and retrieval time. The ANS value affects variance in retrieving declarative information and error rate for retrievals in the model. This instantaneous noise value can also represent variance from trial to trial. Other parameters, such as base level learning, decay, and the characters-per-syllable parameters were built into the model as modifiable but were left fixed at their default values for this study. The search space for the model optimization was defined by the parameter value boundaries: ANS and SYL 0.1 to 0.9, and BLC 0.1 to 3.0.

Optimization Approach

Because the search space was large and assumed to be rather complex a departure from the cognitive modeling community's traditional manual optimization technique was initiated (Kase, 2008). A new front-end function for the cognitive model was developed for execution in a parallel processing environment and the ACT-R parameter values (ANS, BLC, and SYL) were passed to multiple instances of running models from a parallel genetic algorithm (PGA). The SYL parameter was chosen for optimization because vocalization of the answer is the most time consuming aspect of this task. The BLC and ANS parameters were chosen because the task is memory intensive. Other memory parameters could have been chosen and ongoing work is exploring the fitting of other parameters. Normally, the parameter values are set within the model code before runtime. Using the PGA to search the parameter space for promising parameter value sets generating best fits between the model and human data saved a substantial amount of modeler time and computational resources. Model-to-data fit was determined by an objective function, or fitness function, defined as the discrepancy between model performance (number of attempts and percent correct) and the corresponding human performance (e.g., 47.3 – 48.1). The fitness is in terms of error (or cost) with a fitness value of 0 representing perfect correspondence between the model predictions and the human data.

Employing this type of 'automated' optimization approach allowed for 20,000 different sets of parameter value to be tested in a directed manner each time the PGA was executed. Using the approach, the model was optimized to nine sets of human performance data (see Table 2).

Results and Discussion

Table 2 shows the resulting model performance compared to the human performance data using parameter value solution sets identified by the PGA that produced the best fits (fitness values less than 1.0) to the human performance, and suggest how cognition changed. Several trends can be observed within the parameter values producing best fits. The parameter values shown in the table are averaged; denoted by the numeric value in parentheses after the parameter set values (i.e., '(3)' in the first row means that the PGA found 3 parameter sets producing fitness less than 1.0, and that these values were averaged). Each parameter set included in the average was run 200 times (i.e., 200 model runs per parameter set).

Beginning with the seconds per syllable parameter, SYL is shown in the last column and last value in the triple of Table 2. The model predictions indicate that challenged subjects speak a syllable more quickly than threatened subjects. This is true for all treatment groups. LoCAF shows the greatest difference in speech rate with challenge SYL at 0.31 (also lowest SYL overall) and threat SYL at nearly two times slower (0.61). HiCAF differences in SYL are less: challenge 0.40 compared to threat 0.57, a difference of 0.17. PLAC shows a slightly less SYL difference of 0.14. Challenge subjects self-report less stress and are generally confident that they can perform the serial subtraction task well. With less stress and a low dose of caffeine more fluid speech appears to result, or possibly the speech rate acts as a window into the cognitive processes required to complete the subtractions (i.e., fact retrieval, working memory and place-keeping operations, and concatenation of subsolutions).

Overall across treatments, the activation noise parameter values (ANS, first value in triple) are high as compared to what would be manually assigned to the model in the ACT-R modeling community. This could be because the nature of the task is stressful (i.e., purposively used to elicited a stress response). The ANS value range in Table 2 is narrow from the lowest ANS of 0.67 to the highest ANS of 0.78, a difference of only 0.11. This hints at the fact that caffeine may not effect this parameter's role in the model's performance of serial subtraction. ANS values are basically equivalent for the PLAC and LoCAF groups for challenge (0.68) and threat (0.71). In this case, the slightly higher ANS in predicting threatened subjects corresponds to the lower performance (less attempts and lower accuracy), and the self-reports where subjects do not believe they will perform well. Worrying or embarrassment about their poor performance is a distraction and may interfere with working memory processes and verbalizing solutions. The greatest variability in ANS values is found in HiCAF. Surprisingly, the trend reverses with HiCAF challenge predictions yielding a higher ANS value (0.75) than threat predictions (0.67).

The base level constant parameter values (BLC, middle value in triple) show a trend of nearly equivalent higher values for LoCAF and HiCAF challenge conditions (2.65 and 2.69) then threat conditions (2.48 and 2.35), and also for all BLC values under PLAC (2.49, 2.48 and 2.53). In this case, caffeine may be causing a 'boost' in the base level activation value of facts in declarative memory promoting higher probability of selection in response to a retrieval request and quicker fact retrieval time.

Table 2: Optimization results for three treatment groups (PLAC, LoCAF, HiCAF) and appraisal conditions (CH=challenge, TH=threat) comparing human performance and model predictions in number attempts and percent correct (both rounded), and fitness value associated with average (over N) of best fitting (less than 1.0) ACT-R parameter values (ANS, BLC, SYL).

	Human Performance	Model Prediction	Fitness Value	ACT-R parameters ANS, BLC, SYL (N)
PLAC (no caffeine)				
ALL	47.3, 81.5	48.1, 81.4	0.83	0.70, 2.49, 0.44 (3)
CH	50.7, 83.3	50.4, 83.0	0.47	0.68, 2.48, 0.41 (6)
TH	40.4, 77.9	40.3, 77.4	0.36	0.71, 2.53, 0.55 (5)
LoCAF (200 mg caffeine)				
ALL	59.1, 86.5	59.1, 86.7	0.12	0.72, 2.64, 0.33 (4)
CH	62.4, 88.3	62.7, 88.4	0.42	0.69, 2.65, 0.31 (3)
TH	37.5, 74.8	37.2, 74.9	0.58	0.71, 2.48, 0.61 (6)
HiCAF (400 mg caffeine)				
ALL	45.7, 79.2	44.7, 80.4	0.50	0.78, 2.65, 0.47 (4)
CH	51.6, 82.8	46.1, 87.7	0.53	0.75, 2.69, 0.40 (3)
TH	38.9, 75.1	50.4, 92.3	0.53	0.67, 2.35, 0.57 (4)

Conclusion

A cognitive model of the serial subtraction task was developed and fit to the human performance data from three caffeine treatments and by challenge and threat appraisal. This fit suggests that there are systematic changes in cognition due to caffeine and appraisal. Most notable is the speaking rate, but declarative memory retrievals are also affected.

These results show that using a cognitive model and parametric optimization approach can further our understanding of caffeine beyond a human experimentation approach. Overall, the cognitive modeling and optimization approach was successful. The preliminary modeling results and interpretations offer insight into the effects of caffeine on task appraisal and subsequent performance of the task, and promise an improved methodology for the study of other behavioral moderators and other cognitive tasks. At this point in our investigation more analysis is needed and additional parameter sets should be examined, along with continued refinement of the serial subtraction model for predicting the effects of caffeine on cognition.

Acknowledgments

This project is partially supported by ONR grant N000140310248. Computational resources were provided by TeraGrid DAC TG-IRI070000T and run on the NCSA clusters. The authors would like to thank Laura Klein and her lab and Jeanette Bennett at the Department of Biobehavioral Health, Penn State University, for collection of the human performance data and data analysis assistance.

References

Amendola, C. A., Gabrieli, J. D. E., & Lieberman, H. R. (1998). Caffeine's effects on performance and mood are independent of age and gender. *Nutritional Neuroscience*, *1*, 269-280.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.

Anderson, K. J., & Revelle, W. (1982). Impulsivity, caffeine, and proofreading: A test of the Easterbrook hypothesis. *Journal of Experimental Psychology: Human Perception and Performance*, *8,* 614-624.

Benitez, P. L., Kamimori, G. H., Balkin, T. J., Greene, A., & Johnson, M. L. (2009). Modeling fatigue over sleep deprivation, circadian rhythm, and caffeine with a minimal performance inhibitor model. *Methods in Enzymology*, *454*, 405-419.

Fredholm, B. B., Battig, L., Homen, J., Nehlig, A., & Zvarlaw, E. E. (1999). Actions of caffeine in the brain with special reference to factors that contribute to its widespread use. *Pharmacological Reviews*, *51*, 83-133.

Haishman, S. J., & Henningfield, J. E. (1992). Stimulus functions of caffeine in humans: Relation to dependence potential. *Neuroscience & Biobehavioral Reviews*, *16*, 273-287.

Kase, S. E. (2008). *HPC and PGA optimization of a cognitive model: Investigating performance on a math stressor task.* Unpublished PhD thesis, College of IST, Penn State University, University Park, PA.

Klein, L. C., Whetzel, C. A., Bennett, J. M., Ritter, F. E., & Granger, D. A. (2006). Effects of caffeine and stress on salivary alpha-amylase in young men: A salivary biomarker of sympathetic activity. *Psychosomatic Medicine*, *68*(1), A-4.

Kirschbaum, C., Pirke, K. M., & Hellhammer, D. H. (1993). The Trier Social Stress Test—A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*, *28*, 76-81.

Lazarus, R. S., & Folkman, S. (1984). *Stress, appraisal and coping*. New York, NY: Springer Publishing.

Lieberman, H. R., & Tharion, W. J. (2002). Effects of caffeine, sleep loss, and stress on cognitive performance and mood during U. S. Navy seal training. *Psychopharmacology*, *164*, 250-261.

McLellan, T. M., Kamimori, G. H., Voss, D. M., Tate, C., & Smith, S. J. R. (2007). Caffeine effects on physical and cognitive performance during sustained operations. *Aviation, Space, and Environmental Medicine*, *78*(9), 871-877.

Nawrot, P., Jordan, S., & Eastwood, J. (2003). Effects of caffeine on human health. *Food Additives and Contaminants*, *20*, 1-30.

Smith, A. P., Clark, R., & Gallagher, J. (1999). Breakfast cereal and caffeinated coffee: Effects on working memory, attention, mood and cardiovascular function. *Physiology & Behavior*, *67*, 9-17.

Wesensten, N. J., Belenky, G., & Kautz, M. (2002). Maintaining alertness and performance during sleep deprivation: Modafinil versus caffeine. *Psychopharmacology*, *159*, 238-47.

Yerkes, F. M., & Dodson, J. D. (1908). The relationship of strength of stimulus to rapidity of habit-formation. *Journal of Comparative Neurology and Psychology, 18*, 459-482.

Using Heuristic Models to Understand Human and Optimal Decision-Making on Bandit Problems

Michael D. Lee (mdlee@uci.edu)
Shunan Zhang (szhang@uci.edu)
Miles Munro (mmunro@uci.edu)
Mark Steyvers (msteyver@uci.edu)
Department of Cognitive Sciences, University of California, Irvine
Irvine, CA, 92697-5100

Abstract

We study bandit problems in which a decision-maker gets reward-or-failure feedback when choosing repeatedly between two alternatives, with fixed but unknown reward rates, over a short sequence of trials. We collect data across a number of types of bandit problems to analyze five heuristics—four seminal heuristics from machine learning, and one new model we develop—as models of human and optimal decision-making. We find that the new heuristic, known as τ-switch, which assumes a latent *search* state is followed by a latent *stand* state to control decision-making on key trials, is best able to mimic optimal decision-making, and best account for the decision-making of the majority of our experimental participants. We show how these results allow human and optimal decision-making to be characterized and compared in simple, psychologically interpretable ways, and discuss some theoretical and practical implications.

Keywords: Bandit problems, heuristic models, reinforcement learning, human decision-making, optimal decision-making

Introduction

In Bandit problems, a decision-maker chooses repeatedly between a set of alternatives. They get feedback after every decision, either recording a reward or a failure. They also know that each alternative has some fixed, but unknown, probability of providing a reward each time it is chosen. The goal of the decision-maker is to obtain the maximum number of rewards over all the trials they complete. In some bandit problems, known as infinite horizon problems, the number of trials is not known in advance, but there is some probability any trial will be the last. In other bandit problems, known as finite horizon problems, the number of trials is fixed, known, and usually small.

Because bandit problems provide a simple task that addresses fundamental issues of learning and optimality in decision-making, they have been widely studied in the machine learning (e.g., Berry & Fristedt, 1985; Gittins, 1979; Kaebling, Littman, & Moore, 1996; Macready & Wolpert, 1998; Sutton & Barto, 1988) and cognitive science (e.g., Daw, O'Doherty, Dayan, Seymour, & Dolan, 2006; Steyvers, Lee, & Wagenmakers, in press) literatures. In particular, bandit problems provide an interesting formal setting for studying the balance between exploration and exploitation in decision-making. In early trials, it makes sense to explore different alternatives, searching for those with the highest reward rates. In later trials, it makes sense to exploit those alternatives known to be good, by choosing them repeatedly. How exactly this balance between exploration and exploitation should be managed, and should be influenced by factors such as the distribution of reward rates, the total number of trials, and so on, raises basic questions about adaptation, planning, and learning in intelligent systems.

In this paper, we focus on finite-horizon bandit problems. We also restrict ourselves to the most basic, and most often considered, case where of there being only two alternatives to choose between. For this class of bandit problems, there is a well known optimal decision process that can be implemented using dynamic programming (see, for example Kaebling et al., 1996, p. 244). The basic approach is that, on the last trial, the alternative with the greatest expected reward should be chosen. On the second-last trial, the alternative that leads to the greatest expected total reward should be chosen, given that the last trial will be chosen optimally. By continuing backwards through the trial sequence in this way, it is possible to establish a recursive process that makes optimal decisions for the entire problem.

A motivating challenge for our work involves interpreting, evaluating and potentially improving human decision-making. Using the optimal benchmark, it is possible to evaluate how well a person solves bandit problems. The conclusion might be something like "you got 67% rewards, but optimal behavior would have given you 75% rewards, so you are falling short". This seems like only a partial evaluation, because it does not explain *why* their decisions were sub-optimal, and it is not clear how to relate the recursive algorithm to their data to provide this information.

Instead, to help us understand human and optimal decision-making on bandit problems, we evaluate a set of heuristic models. These include several heuristics from the existing machine learning literature, as well as a new one we develop. The attraction of the heuristic models is that they provide simple process accounts of how a decision-maker should behave, depending on a small set of parameters. We choose heuristic models whose parameters have clear and useful psychological interpretations. This means that, when we fit the models to data, and estimate the parameters, we obtain interpretable measure of key aspects of decision-making. Instead of just telling people they are falling short of optimal, we now aim also to tell them "the problem seems to be you are exploring for too long: the optimal thing to do is to stop exploring at about the 5th trial", or "you are not shifting away quickly enough from a choice that is failing to reward you: the optimal thing to do is to leave a failed choice about 80% of the time."

The structure of this paper is as follows. First, we introduce the five heuristics used in this study. We then evaluate their ability to mimic optimal decision-making, and their ability to fit human data we collected for this study. Having found some heuristics that are able to describe human and optimal behavior, we finish by discussing the psychological characteristics of optimal behavior in bandit problems, and the properties of human decision-making we observed.

Five Heuristics

Win-Stay Lose-Shift

Perhaps the simplest reasonable heuristic for making bandit problem decisions is the Win-Stay Lose-Shift (WSLS) heuristic. In its deterministic form, it assumes that the decision-maker continues to choose an alternative following a reward, but shifts to the other alternative following a failure to reward. In the stochastic form we use, the probability of staying after winning, and the probability of shifting after losing, are both parameterized by the same probability γ.

Psychologically, the win-stay lose-shift heuristic does not require a memory, because its decisions only depend on the presence or absence of a reward on the previous trial. Nor is the heuristic sensitive to the horizon (i.e., the finite number of trials) in the bandit problem version we consider, because its decision process is the same for all trials.

ε-Greedy

The ε-greedy heuristic is a standard approach coming from reinforcement learning. It assumes that decision-making is driven by a parameter ε that controls the balance between exploration and exploitation. On each trial, with probability $1-\varepsilon$ the decision-maker chooses the alternative with the greatest estimated reward rate (i.e., the greatest proportion of rewards obtained for previous trials where the alternative was chosen). This can be conceived as an 'exploitation' decision. With probability ε, the decision-maker chooses randomly. This can be conceived as an 'exploration' decision.

Psychologically, the ε-greedy heuristic does require a limited form of memory, because it has to remember counts of previous successes and failures for each alternative. It is not, however, sensitive to the horizon, and uses the same decision process on all trials.

ε-Decreasing

The ε-decreasing heuristic is a variant of the ε-greedy heuristic, in which the probability of an exploration move decreases as trials progress. In its most common form, which we use, the ε-decreasing heuristic starts with an exploration probability ε_0 on the first trial, and then uses an exploration probability of ε_0/i on the ith trial. In all other respects, the ε-decreasing heuristic is identical to the ε-greedy heuristic.

This means the ε-decreasing heuristic does more exploration on early trials, and focuses on its estimate of expected reward more on later trials. Psychologically, the innovation of the ε-decreasing heuristic means it is sensitive to the horizon, making different decisions over different trials.

π-First

The π-first heuristic is usually called the ε-first heuristic in the literature. It is, however, quite different from the ε-decreasing and ε-greedy heuristics, and we emphasize this with the different name. The π-first heuristic assumes two distinct stages in decision-making. In the first stage, choices are made randomly. In the second stage, the alternative with the greatest currently observed reward rate is chosen. The first stage can be conceived as 'exploration' and the second stage as 'exploitation'. In our implementation, a discrete parameter π determines the number of exploration trials, so that the π-th trial marks the last trial of exploration.

Psychologically, the π-first requires both the memory of previous successes and failures needed in the exploration stage, and has a clear sensitivity to the horizon. The notion of two decision-making stages is a psychologically plausible and interesting approach to capturing how a decision-making might balance the tradeoff between exploration and exploitation.

τ-Switch

The τ-switch is a new heuristic, motivated by the idea of latent decision-making stages used by the π-first heuristic. The τ-switch heuristic also assumes an initial 'search' stage, followed by a later 'stand' stage. The trial number at which the change in stages takes place is determined by the parameter τ, similarly to the role of the parameter π. The different decision-making strategies employed in each stage in the τ-switch heuristic, however, rely on an analysis of different possible states in bandit problems.

Figure 1 provides a graphical representation of three possible cases. In Case I, both alternatives have the same reward history. The τ-switch heuristic assumes both alternatives are chosen with equal probability when confronted with this state. In Case II, one alternative has more successes and the same or fewer failures than the other alternative (or, symmetrically, it has fewer failures and the same or more successes). This means one alternative is clearly 'better', because it dominates the other in terms of successes and failures. The τ-switch heuristic assumes the 'better' alternative with (high) probability γ.

The crucial situation is Case III, in which one alternative has more successes but also more failures, when compared to the other alternative. This means neither

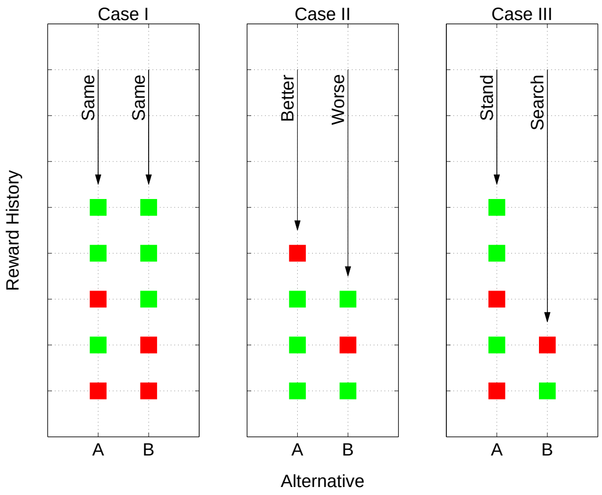


Figure 1: The three different possible cases for a bandit problem considered by the τ-switch heuristic. Green (lighter) squares correspond to previous rewards, while red (darker) squares correspond to previous failures.

alternative can clearly to be preferred. Instead, the alternative chosen more often previously can be conceived as a 'stand' choice, because it is relatively well known. The alternative chosen less often can be conceived as an 'search' choice, because it is relatively unknown. The τ-switch assumes that, faced with an observed State III, the decision-maker chooses the 'search' alternative when they are in the initial latent 'search' stage, with the same (high) probability γ. But, the decision-maker is assumed to choose the 'stand' alternative once they have switched to the latent 'stand' stage.

Psychologically, the τ-switch heuristic has the same memory requirements as the ε-greedy, ε-first and π-first heuristics. The τ-switch heuristic also takes into account the horizon, using the same latent stage approach as the π-first heuristic. It is the detail of the decisions it makes, depending on how its internal state relates to the state of reward history observed, that makes the τ-switch heuristic new and interesting.

Human and Optimal Decision Data

Subjects Data were collected from 10 naive participants (6 males, 4 females).

Stimuli There were six different types of bandit problems, all involving just two alternatives. These six conditions varied two trial sizes (8 trials and 16 trials) and three different environmental distributions ('plentiful', 'neutral' and 'scarce'). Following Steyvers et al. (in press), the environments were defined in terms of Beta (α,β) distributions, where α corresponds to a count of 'prior successes' and β to a count of 'prior failures'. The plentiful, neutral and scarce environments used, respectively, the values $\alpha = 4$, $\beta = 2$, $\alpha = \beta = 1$, and $\alpha = 2$, $\beta = 4$. Within each condition, the reward rates for each alternative in each problem were sampled independently from the appropriate environmental distribution.

Procedure Within-participant data were collect for 50 problems for all six bandit problem conditions, using a slight variant of the experimental interface shown in Steyvers et al. (in press). The order of the conditions, and of the problems within the conditions, was randomized for each participant. All $6 \times 50 = 300$ problems (as well as 5 practice problems per condition) were completed in a single experimental session, with breaks taken between conditions.

Optimal Decision Data We generated decision data for the optimal decision-process on each problem completed by each participant. In generating these optimal decisions, we used the true α and β values for the environment distribution. Obviously, this gives the optimal decision process an advantage, because participants must learn the properties of the reward environment. However, our primary focus is not on measuring people's shortcomings as decision-makers, but in characterizing what people do when making bandit problem decisions, and comparing this to the best possible decision. From this perspective, it makes sense to use an optimal decision process with environmental knowledge. It would also be interesting, in future work, to develop and use an optimal decision process that optimally *learns* the properties of its environment.

Analysis With Heuristic Models

We implemented all five heuristic models as probabilistic graphical models using WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000). All of our analyses are based on 1,000 posterior samples, collected after a burn-in of 100 samples, and using multiple chains to assess convergence using the standard $\hat{R}$ statistic (Brooks & Gelman, 1997).

Characterization of Optimal Decision-Making

We applied the heuristics to behavior generated by the optimal decision process. Table 1 shows the expected value of the inferred posterior distribution for the key parameter in each heuristic model (we observed all of the 'accuracy of execution' parameters were close to 1, as expected). These key parameter values constitute single numbers that characterize optimal decision-making within the constraints of each heuristic. They are shown for each of the plentiful, neutral and scarce environments for both 8 and 16 trial problems.

For WSLS, the parameter values shown in Table 1 correspond to the optimal rate at which a decision-maker should stay if they are rewarded, and shift if they are not. The patterns across environments and trial sizes are intuitively sensible, being higher in more plentiful environments and for shorter trial sizes.

For ε-greedy probability of choosing the most rewarding alternative is high, and very similar for all environments and trial sizes. For ε-decreasing, the starting prob-

Table 1: Expected posterior values for the key parameter in each heuristic model, based on inferences from optimal decision-making, for plentiful, neutral and scarce environments, and 8 and 16 trial problems.

	Plentiful		Neutral		Scarce	
Heuristic	8	16	8	16	8	16
WSLS (γ)	.87	.85	.85	.78	.72	.65
Greedy (ε)	.09	.07	.05	.05	.06	.07
Decrease (ε_0)	.62	.76	.57	.75	.56	.63
First (π)	1.0	1.0	1.0	1.0	1.0	1.0
Switch (τ)	5.1	7.0	4.1	5.0	2.0	2.0

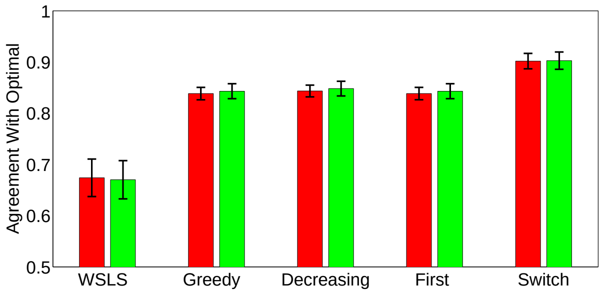


Figure 2: Posterior predictive average agreement of the heuristic models with the optimal decision process for 40 training problems (red, darker) and 10 test problems (green, lighter).

ability of random exploration ε_0, which decreases as trials progress, is higher for more rewarding environments, and also for problems with more trials.

The π-first parameter is the trial at which the switch from random exploration to choosing the most rewarding alternative. This is always the first trial in Table 1, which is essentially a degenerate result. We interpret this as suggesting not that the notion of an exploration followed by an exploitation stage is ineffective, but rather that initial random decisions in a problem with few trials is so sub-optimal that it needs to be minimized.

Finally, the results for the τ-switch heuristic detail the optimal trial to switch moving from 'standing' to 'searching' in the Case III scenario described in Figure 1. This optimal switching trial becomes earlier in a problem as the environment becomes less rewarding, which makes sense. More plentiful environments should be searched more thoroughly for high yielding alternatives. The number of searching trials generally extends moving from 8 to 16 trial problems, but not by much. This also makes sense, since in the fixed environments we consider, longer sequences of exploitation will give many rewards, as long as sufficient exploratory search has been conducted.

All of these optimal parameter settings make sense, and demonstrate how a heuristic can give a straightforward psychology characterization of optimal decision-making for bandit problems. For example, in a neutral environment with 8-trial problems, an optimal decision-maker constrained in their cognitive processing capabilities to applying WSLS should win-and-stay or lose-and-shift 85% of the time. Alternatively, a more cognitive elaborate decision-maker, able to apply the two-stage τ-shift heuristic, should switch from searching to standing after the fourth trial.

How Optimal Are the Heuristics?

Of course, knowing what constitutes optimal behavior within the bounds of a heuristic does not take into account how well decisions will match unboundedly optimal decision-making.

To analyze this aspect of the heuristics' performance, Figure 2 shows the posterior predictive average agreement of the heuristic models with the optimal decision process. The red bars correspond to a training set of the first 40 problems seen by all participants in which the parameters of the heuristic models were inferred by observing the optimal decisions. The green bars correspond to a test set of the final 10 problems seen by all participants, where the inferred parameters for the heuristic models were directly applied with observing the optimal decisions. The relative results between the heuristics are consistent over environments and trial sizes, and so are averaged to give a simple and general conclusion, but include error bars showing one standard error caused by the averaging.

It is clear that training and test performance are very similar for all of the heuristics. This is because the agreement is measured by a complete posterior predictive, which averages across the posterior distribution of the parameters. This means the measure of agreement—unlike measures of fit based on optimized point-estimates for parameters—automatically controls for model complexity. Thus, it is not surprising test performance is essentially the same as training performance.

Most importantly, Figure 2 shows that the WSLS heuristic is not able to mimic optimal decision-making very well, that the ε-greedy, ε-decreasing and π-first heuristics are able to do much better, and that the new τ-switch heuristic is clearly is the best performed.

Heuristics Modeling of Human Performance

We now apply the heuristics to the human data, and explore their ability to account for the way people solve bandit problems. Figure 2 shows the posterior predictive average agreement of the heuristic models with the human decisions. As before, the red bars correspond to a training set of the first 40 problems completed by each participant, and were used to infer posterior parameter distributions for each heuristic. The green bars correspond to agreement on the test set of the final 10 problems, integrating over the already inferred posterior distributions, and without knowing the participants' behav-

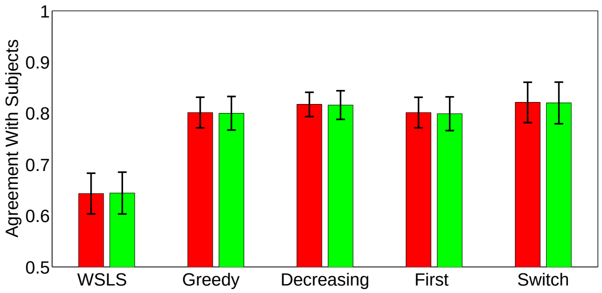


Figure 3: Posterior predictive average agreement of the heuristic models with human decision-making for 40 training problems (red, darker) and 10 test problems (green, lighter).

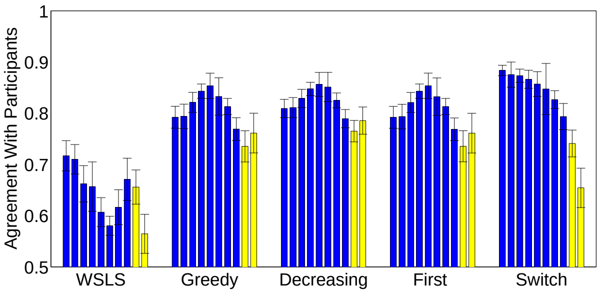


Figure 4: Posterior predictive average agreement of the heuristic models with each individual participant. Two 'outlier' participants, not modeled well by any of the heuristics, are highlighted in yellow (lighter).

ior on the test problems.

Figure 2 shows the ability of the heuristics to model human decision-making follows the same ordering as their ability to mimic optimal decision-making. WSLS is the worst, followed by the three reinforcement learning heuristics, which are approximately the same, and then slightly (although not significantly) improved by the new τ-first heuristic.

Figure 4 examines the ability of the heuristics to account for human decision-making at the level of the individual participants. Each participant is shown as a bar against each of the heuristics. For the first 8 of the 10 participants shown (in blue), the overall pattern seen in Figure 3, holds at the individual level. That is, the τ-switch heuristic provides the greatest level of agreement. For the last 2 of the 10 participants shown (in yellow), this result is not observed, but it is clear that none of the heuristics is able to model these participants well at all. We speculate that these participants may have changed decision-making strategies significantly often to prevent any single simple heuristic from providing a good account of their performance.

In any case, our results show that, for the large majority of participants well described by any heuristic, the τ-switch heuristic is the best. And the complexity control offered by the posterior predictive measure, and verified by the training and test sets, shows that this conclusion takes into account the different model complexity of the heuristics.

Characterization of Human Decision-Making

The analysis in Figure 2 shows the τ-switch heuristic can closely emulate optimal decision-making for bandit problems, and the analysis in Figure 4 shows it can also describe most participants' behavior well. Taken together, these results let us use the τ-switch heuristic to realize our original motivating goal of comparing people's decisions to optimal decisions in psychologically meaningful ways. The key psychological parameters of a well-performed heuristic like τ-switch provide a measure that relates people to optimality.

Figure 5 gives a concrete example of this approach. Each panel corresponds to one of the 8 participants from Figure 4 who were well modeled by the τ-switch heuristic. Within each panel, the large green curves show the switch trial (i.e., the expected posterior value of the parameter τ) inferred from optimal decision-making. These optimal parameter values are shown for each of the plentiful, neutral and scarce environments, for both 8- and 16-trial problems. Overlayed in each panel, using smaller black curves, are the patterns of change in this parameter for the individual participants.

The commensurability of the switch point parameter between people and optimality, and its ease of interpretation, allows for insightful analyses of each participant's performance. Participants like B and F are choosing near optimally, especially in the 8-trial problems, and seem sensitive to the reward rates of the environments in the right ways. Their deviations from optimality seem more a matter of 'fine tuning' exactly how early or late they switch away from exploratory search behavior. Participants like A and D, in contrast, are reacting to the changes in environment in qualitatively inappropriate ways. Participants like C, E, and H seem to perform better on the 8- than the 16-trial problems, and do not seem to be adjusting to the different environments in the 16-trial case. But C is switching at roughly the optimal trial on average, while E is switching too early, and H is too early for the shorter problems and too late for the longer ones. Finally, participant G seems to be employing a 'degenerate' version of the τ-switch heuristic that involves no initial search, but simply stands on the highest success rate alternative throughout the problem.

This analysis is not intended to be complete or exact. Potentially, the other heuristics could provide alternative characterizations with some level of justification. What the sketched analysis does provide a concrete illustration of the way human and optimal performance can be characterized by parametric variation using our best-fitting heuristic model.

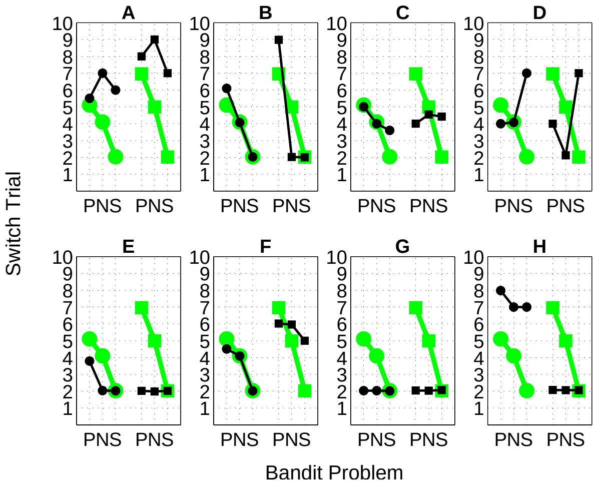


Figure 5: Relationship between the optimal switching point under the τ-first heuristic in (larger, green markers) and inferred switch points for 8 subjects A–H in (smaller, black markers). Comparisons are shown for P=plentiful, N=neutral and S=scarce environments, and 8-trial (circle) and 16-trial (square) environments.

Discussion

One finding from our results is that the τ-switch heuristic is a useful addition to current models of finite-horizon two-arm bandit problem decision-making. Across the three environments and two trial sizes we studied, it consistently proved better able to mimic optimal decision-making than classic rivals from the statistics and machine learning literatures. It also provided a good account of human decision-making, for the majority of the participants in our study.

A potential theoretical implication of the success of the τ-switch heuristic is that people may use latent states to control their search behavior, and manage the exploration versus exploitation trade-off. We think these sorts of models deserve as much attention as those, like ε-greedy, based more directly on reinforcement learning.

One potential practical application of the τ-switch heuristic is to any real-world problem where a short series of decisions have to made be made with limited feedback, and with limited computational resources. The τ-switch heuristic is extremely simple to implement and fast to compute, and may be a useful surrogate for the optimal recursive decision process in some niche applications. A second, quite different, potential practical application, relates to training. The ability to interpret optimal and human decision-making using one or two psychologically meaningful parameters could help instruction in training people to make better decisions. It would be an interesting topic of future research to take the sorts of analysis accompanying Figure 5, for example, and see whether feedback along these lines could improve their decision-making on future bandit problems.

More generally, we think our results illustrate a useful general approach to studying decision-making with heuristic models. Three basic challenges in studying any real-world decision-making problem are to characterize how people solve the problem, characterize the optimal approach to solving the problem, and then characterize the relationship between the human and optimal approach. Our results show how simple heuristic models, using psychologically interpretable decision processes, and based on psychologically interpretable parameters, can aid in all three of these challenges. While our specific results are for short-horizon two-alternative bandit problems, and involve a small set of heuristics, we think the basic approach has more general applicability. We think heuristic models, and their inferred parameter values, are useful for understanding and comparing human and optimal decision-making.

Acknowledgments

This work was funded by award FA9550-07-1-0082 from the Air Force Office of Scientific Research.

References

Berry, D. A., & Fristedt, B. (1985). *Bandit problems: Sequential allocation of experiments* . London: Chapman & Hall.

Brooks, S. P., & Gelman, A. (1997). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*, 434–455.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, *441*, 876–879.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, *41*, 148–177.

Kaebling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* , *4*, 237–285.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS: A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing* , *10*, 325–337.

Macready, W. G., & Wolpert, D. H. (1998). Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on evolutionary computation*, *2*(1), 2-22.

Steyvers, M., Lee, M. D., & Wagenmakers, E.-J. (in press). A Bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*.

Sutton, R. S., & Barto, A. G. (1988). *Reinforcement learning: An introduction* . Cambridge, MA: MIT Press.

Two routes to cognitive flexibility: Learning and response conflict resolution in the dimensional change card sort task

Michael Ramscar, Melody Dye, Jessica Witten & Joseph Klein
Department of Psychology, Stanford University,
Jordan Hall, Stanford, CA 94305.

Abstract

There are at least two ways in which response conflict can be handled in the mind: dynamically, so that conflicting response demands are resolved on-line, and discrimination learning, which reduces the amount of on-line response conflict that needs to be resolved in context. While under fours are perfectly capable of discrimination learning, they appear to lack the ability to dynamically resolve response conflict. They can match their behavior to context in remarkably subtle and sensitive ways when they have learned to do so, but if they have not learned to match a response or a behavior to a context, their inability to handle on-line response conflict is their undoing (for example, in the dimensional change card sort task; DCCS). We present an analysis of how learning in context might aid children's performance in the dimensional change card sorting (DCCS) over time, and a training study in which three groups of age matched under fours attempt to complete the DCCS. We find that appropriate training enables children to flexibly switch between their responses in the DCCS. Without training supporting discrimination learning, children's performance is far worse, and when the task contexts are novel, children fail as expected.

Introduction

Thanks to the insight and inventiveness of developmental psychologists, we know that very young children are different. A three-year-old might girl seem simply a slightly smaller version of her four-year-old brother, however, while he will sail effortlessly, through the battery of tasks that psychologists have devised to expose the shortcomings of the very young, his sister will likely fail every one of them. Her 4-year-old brother will switch responses and probability match in binary choice tasks, understand false belief and the conflicting dimensions of appearance and reality, and switch easily between competing rules in dimensional change card sorting (DCCS; Zelazo, 2006) task, whereas our three-year-old will maximize in binary choice tasks (fixating on the most likely response), fail false belief tasks, be unable to switch from describing the appearance of an object to answering questions about what it really is, and fail to switch from one sorting rule to another, even though the rule is clearly stated (see Ramscar & Gitcho, 2007, for a review).

This raises two questions: first, why do children under four fail to switch to the conflicting but more contextually appropriate response in these tasks; and second, given the inflexibility of thought that these tests reveal, why in the normal course of events do children appear to be perfectly capable of changing their responses and behavior according to context?

Many proposals have been made in trying to answer the first of these questions (see e.g., Zelazo, Müller, Frye & Marcovitch, 2003). In what follows, we seek to answer both of them by examining the different ways in which the conflict between potential responses might be resolved, so that an appropriate response can be given in context. We suggest that that there are at least two ways in which response conflict can be handled in the mind: dynamic response conflict resolution, which enables conflicting response demands to be processed and resolved on-line, and discrimination learning, which enables the strengths by which responses are evoked by contexts to be modulated, reducing the amount of on-line response conflict that needs to be processed and resolved. We suggest that while under fours are perfectly capable of discrimination learning, they lack the ability to resolve response conflict on-line. Under fours are able to match their behavior to context in remarkably subtle and sensitive ways when they have learned to do so. If they have not learned to match a response or a behavior to a context, under fours' inability to handle on-line response conflict is their undoing (for example, in the novel contexts psychologists devise for their tests).

In what follows, we describe the neurological and computational bases for these ideas, and present a computational simulation of how discrimination learning and context might affect children's performance in the dimensional change card sorting (DCCS) over time. The model explains the observed failure of under fours at the DCCS as resulting from a lack of discrimination learning in the context of the "games" children play in the task. Further, it predicts that these children are exposed to the game contexts in ways that promote discrimination learning, they should later succeed at the task with relative ease. We then present a training study in which three groups of age matched under fours attempt to complete the DCCS after exposure to the games that promotes discrimination learning, exposure to the games that does not promote discrimination learning, and where the DCCS games are novel contexts. Consistent with the predictions of the model, we find that after appropriate discrimination learning, children are able to flexibly switch between the various responses required by the DCCS in a contextually appropriate manner. Without appropriate discrimination learning, children's performance is far worse, and when the task contexts are novel, children fail as expected.

The Dimensional Change Card Sort Task

In the Dimensional Change Card Sort (DCCS) Task, three and four year-old children are asked to sort cards with two prominent linked dimensions—a color and shape—into bins in which these dimensions have been reversed. For example, if the child is holding cards with red stars and blue trucks, the bins will be marked with blue stars and red trucks. If the child is asked to sort by color, the red stars will go with the red trucks and the blue stars will go with the blue trucks; if the child is asked to sort by shape, the red stars will go with the blue stars, and the red trucks will go with the blue trucks. When a child is asked to sort by one dimension—say, shape, switching the sort dimension to color will switch the correct sort bins for the card; e.g., red stars match to the truck bin when sorted by color, but the star bin when sorted by shape. For older children and adults, this is a straightforward task.

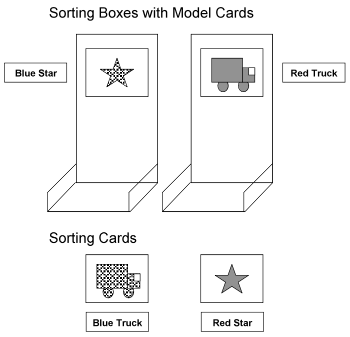


Figure 1: The basic DCCS task. Cards can be sorted by shape (in which case, the red star is sorted into the left bin) or color (in which case, the red star is sorted to the right bin).

When young children are asked to begin sorting by shape, they can easily answer questions regarding the rules for correctly sorting either by shape or by color. In addition, after switching from sorting by shape to sorting by color, children can correctly answer questions about how to correctly sort according to the new rule. However, once children are actually handed a card and asked to sort according to the second rule they have learned, their success in the task varies markedly with age. Generally, 3-year-old children are unsuccessful at this part of the task; they continue to sort the cards according to the first rule (i.e., whatever was learned first, whether it be sorting by shape or color). After age 4, however, children tend to pass the DCCS task and successfully match the cards to the bins both before and after the sorting rules are switched (Zelazo, Frye & Rapus, 1996).

Why do three year olds fail this task? One suggestion is that their poor performance is a related to the late development of prefrontal cortex. Like many other primates, humans are born with an immature brain. In monkeys the post-natal development of the brain occurs at the same rate in all cortical areas (Rakic, Bourgeois, Eckenhoff, Zecevic, & Goldman-Rakic, 1986). In the human cortex, however, while synaptogenesis peaks in visual and auditory cortex within a few months of birth, these developments occur later in prefrontal cortex (Huttenlocher & Dabholkar, 1997; for reviews see Thomson-Schill, Ramscar & Chrysikou, in submission).

One interesting behavioral consequence of this slow prefrontal development is that children appear unable to engage in behaviors that conflict with prepotent responses (see Ramscar & Gitcho, 2007 for a review). The adult ability to select a less well learned, but goal appropriate response is seen in the Stroop Task, in which the subject is asked to identify the ink color of a conflicting color word (e.g., if the word "green" were printed in red ink, red would need to be identified). Performance in this task involves resolving the conflict between the over-learned response (reading) and the appropriate response (ink naming). Adults typically complete the Stroop Task with ease, but young children repeatedly fail similar tasks. In adults, this is made possible by pre-frontal control mechanisms that bias one response over another according to goals or context (Yeung, Botvinick, & Cohen, 2004). The prefrontal cortex functions as a dynamic filter, selectively maintaining task-relevant information and discarding task-irrelevant information (Shimamura, 2000).

If three year olds lack (or are deficient in) the ability to dynamically filter responses in accordance with the demands of a context or goal, this may explain both why they fail at the Stroop Task and why they fail to switch rules in the DCCS. If a card depicts a red star, "red" elicits one response (sorting into the color bin) whereas "star" elicits a different conflicting response (sorting into the shape bin). Thus in the standard DCCS task, successfully switching rules involves changing from one response associated with a given cue—the card—to an alternative, conflicting response. Since this kind of response conflict processing appears to be the preserve of the frontal areas of the brain (Yeung, Botvinick, & Cohen, 2004; Thomson-Schill et al, *in press*), it seems that the failure of three year olds in the DCCS task—that is, their failure to mediate response conflict—may be related to slow pre-frontal development.

Discrimination Learning

If young children lack the ability to resolve conflict on-line, discrimination learning provides another means by which they might still learn to succeed on the DCCS. This is because the games associated with each sorting rule provide cues to the appropriate responses, in addition to the shape and color in the cards themselves. The "shape game" is a cue to the response "sort into the shape bin" and the "color game" is a cue to the response "sort into the color bin." Since children fail the task despite the presence of these cues, it is clear that under ordinary circumstances, the game cues do not provide sufficient extra scaffolding to enable three year olds to pass the DCCS. However, an obvious difference between the cards and the games is that children have a lot of experience with colors and shapes and the various responses they elicit, whereas they have comparatively little experience with sorting games.

To explain why this might matter, we need to consider the way that responses that lead to response conflict in the DCCS are learned and discriminated. Discrimination learning is a process by which information is acquired about the probabilistic relationships between important regularities in the environment (such as objects or events) and the cues that allow those regularities to be predicted (see e.g., Rescorla & Wagner, 1972; Gallistel & Gibbon, 2000).

Crucially, the learning process is driven by discrepancies between what is expected and what is actually observed in experience (termed "error-driven" learning). The learned predictive value of cues produces expectations, and any difference between the value of what is expected and what is observed produces further learning. The predictive value associated with cues is strengthened when relevant events (such as events, objects or labels) are under-predicted, and weakened when they are over-predicted (Kamin, 1969; Rescorla & Wagner, 1972). As a result, cues compete for relevance, and the outcome of this competition is shaped both by positive evidence about co-occurrences between cues and predicted events, and negative evidence about non-occurrences of predicted events. This produces patterns of learning that are very different from those that would be expected if learning were shaped by positive evidence alone (a common portrayal of Pavlovian conditioning). Learners discover the predictive structure of the environment, and not just simple patterns of correlations in it.

To briefly illustrate how discrimination learning works, imagine a child learning to play the games associated with the DCCS. We shall first consider a case where the experimenter shows the child the card, and is asked to sort them by color.

We can assume that previously the child has heard objects referred to before in terms of both their shape and their color because, though they usually fail to sort using these dimensions, they can reliably name the shapes and colors on the cards (Kirkham, Cruess & Diamond, 2003). The problem, therefore, seems to be that children experience more response-conflict with regards the correct dimension to attend to in order to sort by the rule than they do when it comes to selecting an appropriate dimension for naming (this is perhaps unsurprising, since children will have more experience with names than sorting). That is, when children are asked to *sort* the cards, both shape and color appear to be active as relevant dimensions to sort by.

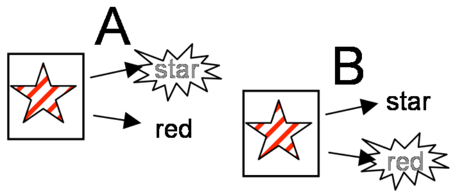


Figure 2. If a child has learned that a card with a red star on it might be sorted by *red* or *star*, when the card is presented she will expect to sort by *red* and *star*. In sorting by *red* (A), the child's expectations will weaken the association between the card and *star* in this context. The converse is true in the (B).

If the cards cause a child to expect both dimensions to be relevant, but only one is used in sorting, there will be a violation of expectation (Figure 2). Given that a response to the relevant dimension event didn't occur, she will begin to adjust her expectations accordingly. This may then cause problems when the child is asked to sort by the other dimension, because the child will have learned to *ignore* the now relevant dimension on the earlier sort trials.

This is because in the *color game* the *red star card* is sorted by "red." Because the *red star card* has been previously associated with both "red" and "star", it also incorrectly cues "star." As a result, the value of the association between *red star card* and "star" will decrease ("star" will be learned about even though it is not heard). Further, because the context *color game* has been introduced, in subsequent color game trials, a conjunctive cue *red star card* + *color game* (e.g., Gluck & Bower, 1988) can compete with *red star card* (and *color game*) for associativity to "red".

The converse will occur if the child switches to the shape game. Because all of the dimensions of the *red star card* will be present in both the color and the shape games, *red star card* alone will prove to be a less useful cue than the conjunctive cues *color game* + *red star card* and *shape game* + *red star card*.

To formally test these ideas, we simulated the competition between conjunctive cues representing *color game* + *red* and *shape game* + *star* and the individual cues *red* and *star* across repeated DCCS trials using the Rescorla & Wagner (1972) model.[1] The

[1] In the Rescorla-Wagner (1972) model the change in associative strength between a stimulus *i* and a response *j* on trial n is defined to be:

$$\Delta V_{ij}^{n} = \boldsymbol{\alpha}_i \boldsymbol{\beta}_j \, (\lambda_j - V_{total})$$

simulation assumes that the output is the appropriate sorting response, and that *red* and *star* have been previously learned as sorts for the *red star card* "red" 50% of the time each, and that *color game* + *red* will predict "red" 100% of the time. The individual cue was initially trained on with color and shape as alternate labeling events, and then the *color game* was introduced, and that *color game* was present on all color trials (there are two colors, equally represented).

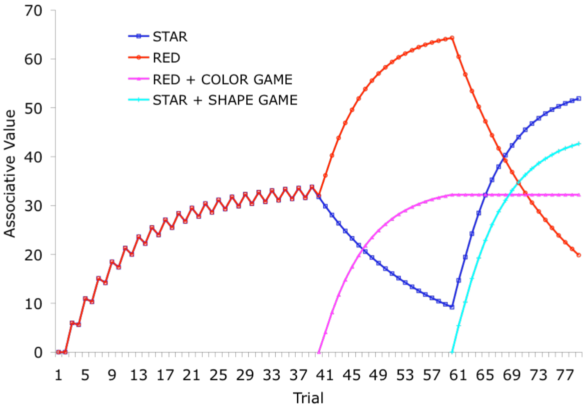


Figure 3: Rescorla-Wagner simulation of cue competition in two DCCS trials. The erroneous expectations *shape* produces in *color game* trials cause it to be unlearned, resulting in red is being a far more active cue on the switch trial (trial 61).

In the first DCCS game shown in Figure 3, *red* and the conjunctive cues the *color game* + *red* gain in associative value as a result of the diminishing value of the *star* cue. Importantly, even though all of the cues co-occur with exactly the same frequency with "red," learning effectively dissociates *red star card* and *color game* from "red" in this situation.

As can be seen in Figure 4, assuming correct sorting, the erroneous expectations produced by *red* and *star* cause them to lose out in competition with the conjunctive cues that embody the games as contexts, such that the dimensional cues alone are effectively unlearned in this context, even though they co-occur with the appropriate responses with exactly the same frequency as the conjunctive cues. This is because in error-driven learning predictive power, not frequency or simple probability, determines cue value. Thus, as long as the cards are correctly labeled in each context, a child will learn to ignore the ambiguous cues, thereby improving response discrimination.

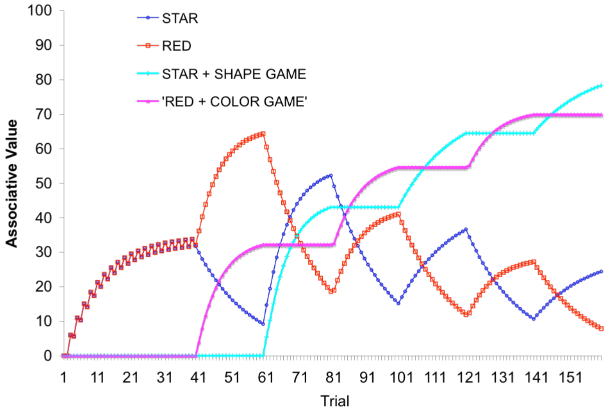


Figure 4: Rescorla-Wagner simulation of cue competition in six DCCS trials. Each peak represents a rule switch.

Cue competition devalues the cues that result in prediction error and increases the value of those that do not, emphasizing the value of reliable cues. To illustrate the importance of cue competition to discrimination learning, it is useful to consider the effect of learning in the absence of cue competition.

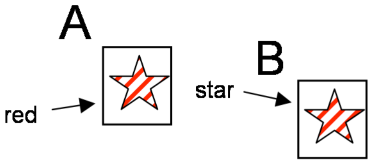


Figure 5: When labels precede the cards as discrete events, there may be no opportunity for cue competition. Each cue will simply come to predict the card to asymptote.

We call learning in the situation just described, where card Features predicted Labels, *FL-learning*. We can then define the situation in which Labels predict Features as *LF-learning* (Ramscar, Yarlett, Dye, Denny & Thorpe, in submission). In this situation, something very different will happen in learning. To explain why, we need to consider how the structure of cues and predicted events conspire to produce cue competition. In the FL-learning scenario described above, the labels for the relevant dimensions are discrete, and only one occurs at any one time. This results in prediction error if cues present on trials when "red" is subsequently labeled are present on trials when "star" is subsequently labeled. Cues not present on one or the other type of trial come to be favored as a result of cue competition. However, if the labels (or the labels in context) are presented prior to the cards (Figure 5), because the labels are discrete as events and as stimuli (whereas the dimensions of the cards in context are not), they cannot compete as cues, so no discrimination learning will take place.

The model thus specifies how the associative strength (V) between a conditioned stimulus (CS_i) and an unconditioned stimulus (US_j) changes as a result of discrete training trials, where n indexes the current trial. $0 \leq \alpha_i \leq 1$ denotes the saliency of CS_i, $0 \leq \beta_j \leq 1$ denotes the learning rate of US_j, λ_j denotes the maximum amount of associative strength that US_j can support, and V_{total} is the sum of the associative strengths between all CSs present on the current trial and US_j. Learning is governed by the value of (λ_j - V_{TOTAL}) where λ_j is the value of the predicted event and V_{total} is the predictive value of a set of cues. In the simulation, all λ = 100%, α_i=0.2 and β_j=0.3.

Because there are no other labels (cues) to compete for associative value, there can be no loss of potential associative value to other cues over the course of learning. Because of this, the effect of prediction-error on cue value will be very different. In the absence of cue competition, the cue value of a label will simply come to represent the proportion of successful predictions it has made relative to the proportion of unsuccessful predictions; the cue value of a label will simply approximate the conditional probability of a feature given the label (in the DCCS, where cards vary in color or shape, this variance will be represented probabilistically after LF-learning). LF-learning thus provides little help when it comes to learning about situations in which response conflict is inherent (Ramscar et al, in submission).

Error-Driven Learning and the DCCS

The analysis above suggests that if children correctly respond to the appropriate dimensions in the early stages of the DCCS, contextual learning will reduce response conflict in later trials. Children trained to associate sorting by shape with a "shape game" and sorting by color with a "color game" can eliminate the response-conflict normally associated with the DCCS by learning context-dependent rules; for example, *"**red** star card + **shape game** sort by red. "*

Given stimulus generalization (Shepher, 1987), one might expect that these will generalize to a degree to, *"**color** shape card + **color game** sort by color"* Similarly, we might expect that if children learn to attend to one dimension in learning about a response in context, they might transfer that learning to another response. Since children can *name* the appropriate dimensions of the cards in the DCCS before they can sort them, we expected that if children were taught to associate naming the appropriate contexts with the game rules in an FL-training configuration, they would learn the high predictive value of these specific cue configurations and that this contextual learning might then enable them to successfully sort in the same contexts in the DCCS task.

Since we would expect that similar training in LF configuration would result only in the learning of the transitional probabilities between the dimension labels and the cards (as described above), the lack of cue competition in this condition ought to result in far less reduction in the amount of response conflict in the task than FL-Learning. To test these ideas, we examined the effect this kind of off-line discrimination training on children's on-line performance in the DCCS.

Training Experiment

Participants

47 children between 3- and 4-years-old (M = 3 years, 6.8 months) participated in this study.

Methods and Materials

Two groups of children received either Label-Second (FL) or Label-First (LF) training on the cards, before completing standard DCCS tasks (Zelazo, 2006). A control group was tested on the DCCS without training.

In the XL (label-second) condition, children were introduced to the shape and color games prior to the DCCS. They were told, "In the shape game, we name the different shapes on these cards." The experimenter then presented the first card to the child and asked the child to label it. After children correctly labeled the first 6 of the 12 cards, the experimenter said, "we're going to play the color game. In the color game, we are going to say what colors are on these cards." Children then labeled the remaining 6 cards in the new game.

While children in the FL-condition saw the card and labeled it, children in the LF-condition were asked to say the label first and then saw the card. They were told, "In the shape game, we name the different shapes on these cards. The first card is going to be a flower– can you say 'flower'?" The experimenter showed the card to the child only after the child had repeated the label. The structure of the LF-training was the same as the FL-training: naming 6 cards by one dimension and then switching to the other dimension.

The two training groups (FL and LF) then completed two standard DCCS tasks, with the first testing dimension (either shape or color) counterbalanced across children. There were 12 test trials completed by each child (six consecutive trails for the first dimension and six for the second dimension). Children were required to correctly sort six cards in the pre-switch, and before each trial, children were either reminded of the current game's rules or asked to answer "knowledge questions," such as, "Where do the flowers go? Where do the boats go?" Children were given no feedback about their sorting of the cards.

Once a child had sorted six cards along the pre-switch dimension, the sorting dimension was switched. Exactly six cards were sorted in the post-switch test. After the first DCCS task, the children completed a second standard DCCS task with new cards.

Results

All the children in the two training conditions correctly labeled the cards. Children were considered to have "passed" the DCCS task if they sorted at least 5 out of 6 of the post-switch cards correctly. 69% of the FL-trained children successfully switched rules in the first DCCS task, and 75% in the second DCCS task. By contrast, in the 33% LF trained children completed the first rule switch, and 40% the second. 19% of the control children switched rules in each test (Figure 4).

Chi-square ($\chi 2$) tests revealed that children in the FL (Label-Second) condition had significantly higher passing rates (11/16 children passed) in the first DCCS as compared to children in the LF (Label-First) condition (5/15); χ2 [1, N = 31] = 9.7, p = 0.005; second test, label first, 12/16 children passed as compared to 6/15 in the label second condition, χ2 [1, N = 31] = 17.0, p = 0.001). Against the control group (3/16), the comparisons with the FL (Label-Second) group were, first switch, χ2 [1, N = 33] = 14.9, p = 0.001; second switch, χ2 [1, N = 33] = 23.7, p = 0.001.

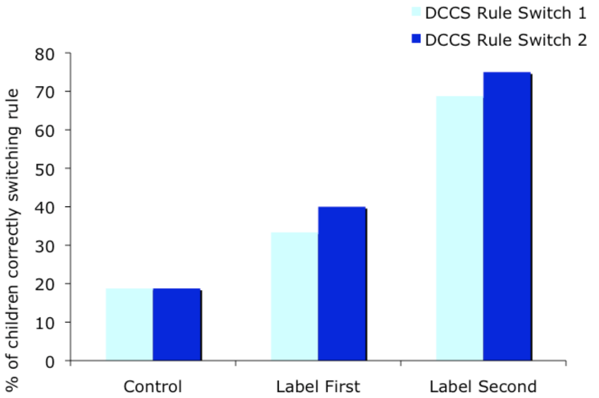


Figure 4: Percentage of children successfully switching rules in the first and second DCCS tasks by condition.

Discussion

We suggested that the observed failure of under fours in the DCCS might result from a lack of discrimination learning about the contexts provided by the "games" children play in the task. We predicted that if children were exposed to the game contexts in ways that promote discrimination learning, they would later succeed at the task with relative ease. Consistent with these predictions, we found that after appropriate discrimination learning, children were to flexibly switch between the various responses required by the DCCS in a contextually appropriate manner. With less appropriate discrimination learning, children's performance was far worse, and when the task contexts were novel, children failed as expected.

This finding is consistent with our suggestion that that there are at least two ways in which response conflict can be handled in the mind: dynamic response conflict resolution, which enables conflicting response demands to be processed and resolved on-line, and discrimination learning, which enables the strengths by which responses are evoked by contexts to be modulated, reducing the amount of response conflict that needs to be processed and resolved. It appears that while under fours are perfectly capable of discrimination learning, they lack the ability to resolve response conflict on-line (see also Ramscar & Gitcho, 2007; Thomson-Shill et al, in submission). As the children who received FL-Training show, discrimination learning allows under fours to match their behavior to context in remarkably subtle and sensitive ways once they have learned to do so. However, as the performance of children in the LF-training and control groups shows, if children have not learned context appropriate behavior, their inability to resolve response conflict dynamically causes problems when dealing with the demands of responding flexibly in ambiguous situations.

Acknowledgments

This material is based upon work supported by NSF Grant Nos. 0547775 and 0624345 to Michael Ramscar.

References

Gallistel, C.R., & Gibbon, J. (2000). Time, rate and conditioning. *Psychological Review*, 107, 344-389

Gluck and Bower (1988). Evaluating an Adaptive Network Model of Human Learning, *Journal of Memory and Language*, 27, 166-195

Huttenlocher, P. R., & Dabholkar, A. S. (1997). Regional differences in synaptogenesis in human cerebral cortex. *Journal of Comparative Neurology*, 387(2), 167-178

Kamin L.J. (1969). *Predictability, surprise, attention, and conditioning.* In: Campbell B, Church R (eds). Punishment and Aversive Behaviour. Appleton-Century-Crofts: New York.

Kirkham, N.Z. , Cruess, L.M., & Diamond, A. (2003). Helping Children Apply their Knowledge to their Behavior on a Dimension-Switching Task. *Developmental Science*, 6(5), 449-467.

Ramscar, M. Yarlett, D., Dye, M., Denny, K., & Thorpe, K. (in submission) The Feature-Label-Order Effect In Symbolic Learning

Rescorla R.A. and Wagner A.R. (1972). A Theory of Pavlovian Conditioning: *Variations in the Effectiveness of Reinforcement and Nonreinforcement.* In Black & Prokasy (Eds.), Classical Conditioning II: Current Research and Theory. New York: Appleton-Century-Crofts.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237, 1317-1323

Thompson-Schill, S. L., Ramscar, M., & Evangelia, C. G., (in submission) Cognition without control: When a little frontal lobe goes a long way

Yeung, N., Botvinick, M. M., & Cohen, J. D. (2004). The neural basis of error detection: conflict monitoring and the error-related negativity. *Psychological Review*, 111(4), 931-959

Zelazo, P. D., Frye, D., & Rapus, T. (1996). An age-related dissociation between knowing rules and using them. *Cognitive Development*, 11, 37-63.

Zelazo, P. D. (2006). The dimensional change card sort (DCCS): A method of assessing executive function in children. *Nature Protocols*, 1, 297-301.

Milliseconds Matter But So Do Learning and Heuristics

Bella Z. Veksler (zafrib@rpi.edu)
Rensselaer Polytechnic Institute
110 8th St., Troy, NY 12180

Abstract

Prior work has shown that the interleaving of perceptual, motor, and cognitive components results in a considerable speedup in the performance of a simple decision making task (Veksler, Gray, & Schoelles, 2007). The current modeling effort conducted using the ACT-R cognitive architecture (Anderson & Lebiere, 1998) is intended to demonstrate how this interleaving might be learned, and how decision-making in this task might take place. The model learns the interleaving and exhibits a speedup in performance similar to that of human participants (RMSE=4.3sec). Furthermore, the model matches human accuracy by using a simple heuristic to make decisions.

Introduction

Previous work has shown that milliseconds matter in understanding human performance (Gray & Boehm-Davis, 2000; Veksler et al., 2007). This millisecond improvement has been shown to occur in a table-based, decision-making task (Lohse & Johnson, 1996) without resorting to changes in higher-order decision-making strategies. Furthermore, exploratory modeling revealed the necessity to focus on the millisecond level considerations in skilled task performance. It was found that an important aspect of the model in mirroring the speedup in performance observed in human participants was the interleaving of cognitive, perceptual, and motor operations. An additional speedup was observed in human data as participants minimized the distance they moved the mouse while interacting with the interface.

Our current modeling effort seeks to extend this by (1) including a learning component to the model whereby the model learns the interleaving and distance minimization on its own, and (2) implementing a higher order strategy to match human accuracy performance.

The Task

The experimental environment used in this research was designed to study and model how information access influences the way in which a decision is made – specifically what information is considered and how it is integrated given the environmental constraints and accessibility of information. In particular, we were interested in whether or not people would take advantage of particular regularities in the environment in order to maximize their score. We hypothesized that this exploitation would occur more when the cost of information acquisition was higher (longer lockout durations).

We used a simple table task (see Figure 1) similar to the one used in a previous study (Veksler et al., 2007) with a few important alterations. The current task environment contained five alternatives (arranged in rows) with a value on each of five attributes (arrayed in columns). In addition, each attribute had an assigned probability value which indicated that attribute's relative importance to the alternative's total score. However, the values in the grid were not visible to the participant and they could only uncover one value at a time. The task environment also allowed us to manipulate the duration of the lockout between a participant selecting a cell in the grid and the value of that cell appearing on the screen, so as to allow us to determine the cognitive and perceptual-motor tradeoffs involved.

In the previous study we conducted in the lab, we manipulated how information was accessed – whether participants could see an entire row, an entire column, or only one cell at a time. In the current study, we instead wanted to explore what particular pieces of information people would gravitate towards given a different cost of exploring the grid – how long they had to wait for information to appear. We hypothesized that the cost of information acquisition would influence not only the exploration of the task environment but also the accuracy of the decisions.

Another important change from the original study, is that we went back to the original decision-making table task and implemented different 'gambles,' composed of various sets of probability values for the attributes, in order to see how they would affect performance (Payne, Bettman, & Johnson, 1988) since that work indicated that the probability landscape of the task influenced the strategies people used to complete the task.

Method

We used a traditional decision-making table task for the study.

Participants

A total of 75 undergraduates (22 females and 53 males) from Rensselaer Polytechnic Institute participated in the study. The average age was 19.21 years (*SD* = 2.05). Students received extra credit for their participation.

Design

There was one between-subjects independent variable of lockout duration with 3 levels. The levels varied the duration of the lockout prior to a value appearing on the screen when a participant clicked on a cell. The three lockouts were 0s (0-lock), 2s (2-lock), and 4s (4-lock). However, for purposes of the models we only focused on the 0-lock condition. There was a within-subject independent variable of gamble type with 4 levels. The gamble types are listed in Table 1. Each gamble type consisted of 5 column (outcome) probabilities that were randomized on each trial within a block of 10 trials. The dispersion of each gamble type refers to the degree to which one of the column probabilities ‘dominates’ the others. For example, Gamble Type 0 has one column probability of .6, which is significantly greater than any of the other column probabilities. Gamble Type 0 therefore has a higher dispersion value than any of the other gambles since cell values in the column containing a probability of .6 would contribute more to the final value of an alternative (row) as compared to any other columns. The order of the gambles was randomized within each epoch of 40 trials (10 consecutive trials in each block contain the same gamble type). There were two epochs in the study resulting in 80 trials.

Table 1 : Gamble types used in the study. Column probabilities are randomized from trial to trial within a block of 10 of a particular gamble type. The dispersion value is the standard deviation of the 5 probabilities comprising the gamble.

Gamble Type	Column Probabilities	Dispersion
0	.6,.1,.1,.1,.1	.22
1	.4,.3,.1,.1,.1	.14
2	.3,.2,.2,.2,.1	.07
3	.2,.2,.2,.2,.2	0

Materials

The experiment was presented using a computer running Mac OS X on a 17” flat-panel LCD monitor set to 1024x768 resolution. The software used for the experiment was written in LispWorks 5.0. Each trial consisted of a blank grid being presented to participants (Figure 1).

Along the top of the grid were listed the corresponding column probabilities for that column. The alternatives to choose among were the rows in the grid and participants had to click on the radio button to the left of the alternative to make their choice. Each cell in the grid could be uncovered by clicking on it. Once a cell was clicked, any cell clicked prior to the current one would be covered up. Therefore, only one cell value was visible at any given time. Since we found that in our original study, the task was easier for the participants than we originally anticipated, in order to make the current version a bit more difficult, the cell values were randomly selected from the range 11 to 50 rather than being one of 0, 2 or 4.

Procedure

Each participant was run separately. Participants were asked to turn off their cell phones for the duration of the study. After signing informed consent forms and going through the instructions on how to do the task, each participant completed 80 decision-making trials. These were broken down into blocks of 10 and each block of 10 had one of the 4 gamble types. Participants were instructed to choose the alternative (row) that had the highest weighted summed value. Specifically, the expected value of any given alternative can be calculated by:

$$EV(alt_j) = \sum_{i=1}^{5} p_i v_{j\,i}$$

p: outcome (column) probability in column i
v: cell value of cell in row j and column i

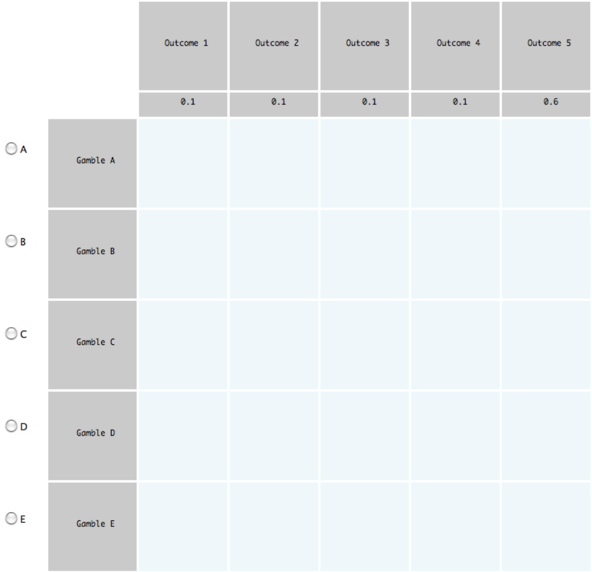


Figure 1: Task Environment

The reward given for each trial was the ratio of the alternative chosen by the participant compared to the best alternative’s expected value. Therefore, if the participant chose the best alternative they received a reward of 100 points, if the next best alternative (and its ratio to the best was 98) then they would receive 98 points.

Participants were given feedback on their score after each trial, along with how long they spent on the trial and how many cells they uncovered. At the end of a block of trials they were given feedback on their average score for that block. At the end of each epoch they were given feedback on the average score over the 40 trials.

Results

Several participants had to be excluded from the analysis due to software malfunction. Consequently, only data from 58 participants (16 females and 42 males) was used for the analysis, 20 participants in the 0-lock condition, 19 in the 2-lock and 19 in the 4-lock. However, it should be noted that the current modeling work only addresses the 0-lock condition of this study. Future work will also incorporate the other conditions.

Accuracy

A 4x3 repeated measures ANOVA on the effects of lockout and gamble type on average accuracy over 80 trials was conducted. The repeated variable was gamble type. There was not a significant gamble*lockout interaction, $F(6, 165) = 1.11$, $p = 0.358$. There was a significant main effect of gamble type, $F(3, 165) = 62.2$, $p < 0.001$. There was also a significant main effect of lockout, $F(2, 55) = 6.87$, $p < .01$. Figure 2 illustrates the trends in accuracy across the four gamble types with respect to the lockout condition.

There was a significant linear trend, $F(1, 55) = 179.6$, $p < .01$, $\omega = .46$, indicating that as the dispersion of the gambles decreased, average score increased. Post-hoc tests revealed significant differences between 0-lock and 4-lock conditions, with a mean difference of 3.28, $p < .01$.

These results indicate that participants in longer lockouts had on average less accurate choices and that accuracy was worse for gambles that had more 'dominating' probability columns.

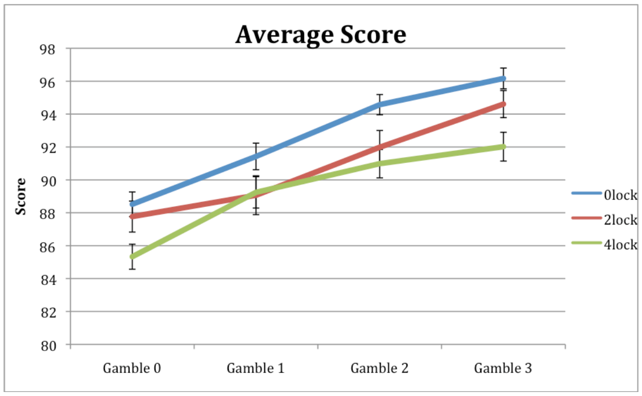


Figure 2: Average Accuracy across Gambles and Lockout Conditions. Error bars are standard error.

Duration of Trial

A 8x3 repeated measures ANOVA was conducted on the effects of lockout and block on how long cell values appeared on the screen. The repeated variable was block number. There was not a significant block*lockout interaction, $F(5.7, 156.66) = 2.07$, $p = 0.06$. There was a main effect of block, $F(2.85, 156.66) = 25.41$, $p < 0.01$. There was also a significant main effect of lockout, $F(2, 55) = 11.94$, $p < .01$. Figure 3 illustrates the trends in average trial duration. Of note here is that there is a significant speedup over the course of the study, in all of the conditions.

Location of Cell Clicks

In order to better understand the strategies people were using to do the task, we looked at which cells participants tended to uncover. In the previous study (Veksler et al., 2007), we found that when given the opportunity to view values by rows vs. by columns, participants chose to check cell values within a row before transitioning to the next row, rather than clicking consecutive cells within a column. We subjected the data of the 0-lock group from the current study to the same analysis. We examined the percent of cell clicks that were either on two consecutive cells in a row or in a column (henceforth referred to as cell transitions). We found that about twice as many cell transitions occurred within a row rather than within a column (Figure 4).

A paired sample t-test revealed a significant difference between the percent of cell transitions within a row ($M = .59$, $SE = .04$) versus within a column ($M = .29$, $SE = .04$), $t(19) = 3.96$, $p < .001$. This suggests that people tended to use a by-row strategy of evaluating alternatives rather than focusing on the columns and our current modeling effort reflects this strategy as well.

We were also interested in whether participants tended to consider the probability values assigned to the columns in their decision making process. In particular, we hypothesized (and previous work by Payne et. al. has shown) that gambles that had higher dispersion values should have more cells uncovered containing the higher probability columns. For the sake of brevity, our findings were that there was not a significant difference between the percent of cells participants clicked in the different probability columns as compared to what would be expected by chance.

We also hypothesized that cells in the higher probability columns would be uncovered earlier in the trial rather than later. However, we found that although there was a considerable bias toward checking grid values starting at the

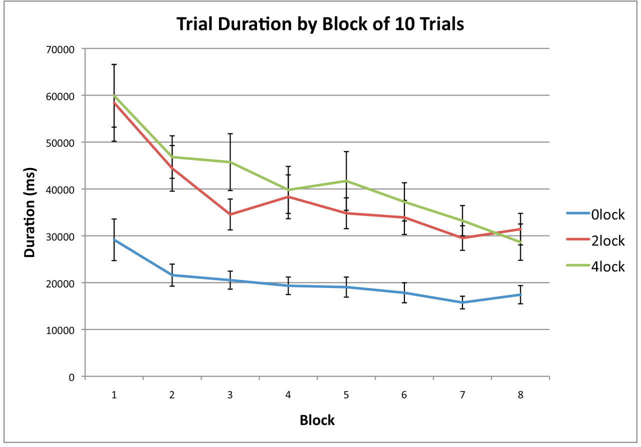


Figure 3: Average duration of trial by block of 10 trials. Error bars are standard error.

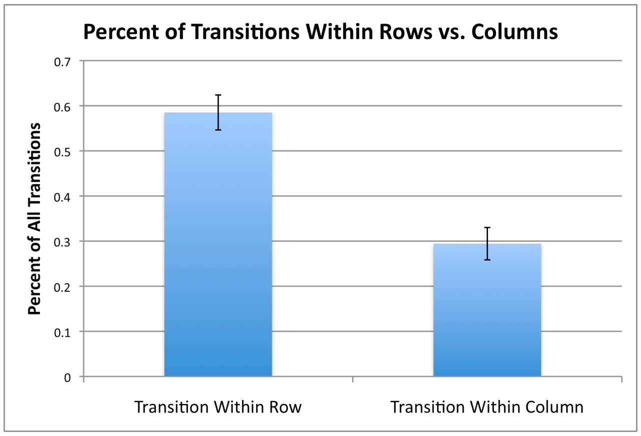


Figure 4: Percent of cell click transitions occurring within a row versus within a column. Error bars are standard error.

top row and moving down (average first click on top row = 1.07, average first click on bottom row = 12.08) and a bias toward checking cells in the left hand columns first (left column = 5.02, right column = 8.7), there was not a significant bias toward checking higher probability columns first.

The Model

To model human performance on this task, we used the ACT-R cognitive architecture (Anderson et al., 2004). ACT-R is a modularized production system with a subsymbolic memory module. It has visual and motor modules to embed it in the task environment. It also has declarative memory and a procedural module. In addition, it has imaginal and goal buffers to store its working memory and goal chunks, respectively. Thus, it serves as a good framework to model human performance on this simple table task.

The current modeling work combined the static models of previous modeling work (Veksler et al., 2007), to demonstrate the learning component in order to fit human data on the task. Furthermore, whereas the previous modeling effort was more concerned with the speed of the interactive routines, the current model also attempts to reproduce accuracy.

The structure of the current model is similar to that of the previous models and is briefly described here. There are roughly four components to the model: switching between alternatives, moving through the cell values within an alternative, comparing the current alternative's value to the best so far, and answering. Figure 5 illustrates the flow of the model and the various productions involved. There are two important changes from the previous models (Veksler et al., 2007) to the current model. The first is the introduction of two sets of competing productions intended to produce a learning effect in the model. The second is the change in strategy implemented by the model to complete the task. We will address each of these important changes in turn.

Competing Productions – Learning Speedup

In matching trial duration of the human data, we implemented two sets of competing productions intended to demonstrate the speedup in performance.

The first two productions that compete occur in the "Switching Between Alternatives" part of the model. As per the previous modeling effort, we found that human participants initially clicked on cells in a left to right fashion whereas later they alternated the direction depending on their ending position in a given row. We thus incorporated this alternating behavior into the model thereby decreasing the distance the mouse had to move when a new alternative was encountered. Since move-mouse execution time in ACT-R is closely related to the distance that the mouse must move, as per Fitts' Law (Fitts, 1954; MacKenzie, 1992), this feature allowed the model to transition faster between alternatives (about 900ms faster over the course of the trial). The two competing productions 'change-row l->r' vs.

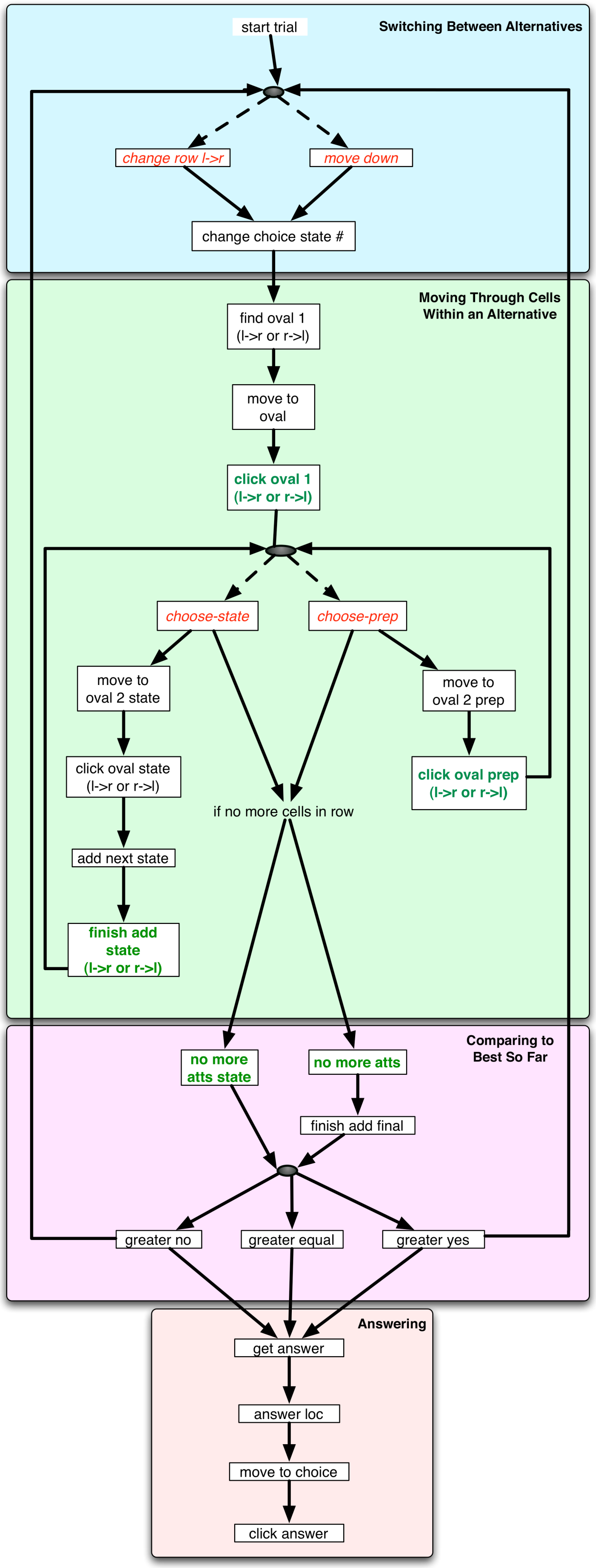


Figure 5: Schematic of the Model. Dashed lines indicate competing productions. Productions in green propagate a reward. Productions in red are competing

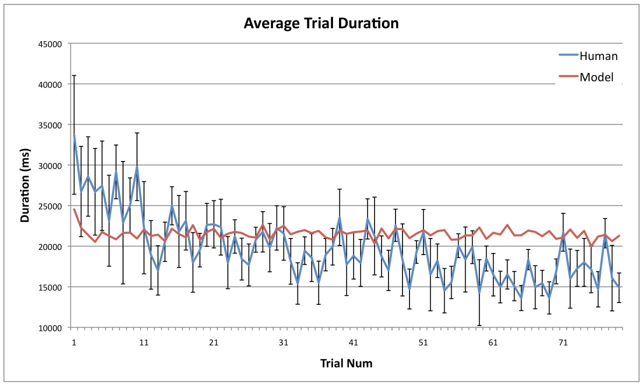


Figure 6: Average trial duration comparison between model and human data. Error bars are standard error.

'move down' are the two types of transitions that we noticed in our human data. Initially the utility of the 'change-row l->r' production is considerably greater than the 'move-down' production, however, the model quickly learns the greater utility of choosing to move down to the next row rather than always resorting to reading the cell values left->right.

The second set of productions that compete occurs in the "Moving Through Cells Within an Alternative" part of the model. Again, as per our previous modeling effort, we noticed that a considerable speedup in performance could be attained by having the model interleave cognitive, perceptual, and motor components (Veksler et al., 2007). The two competing productions are 'choose-state' and 'choose-prep.' The productions following the 'choose state' production all have no interleaving of the perceptual-motor-cognitive components whereas the productions following the 'choose-prep' production do include all the interleaving as described in previous work, and as can be seen in Figure 5, comprise half as many productions.

ACT-R uses a reinforcement learning mechanism for updating production utilities and is based on the amount of reward and time since the production fired that the reward has been triggered as well as a noise parameter. The utility of a production *i* at time *n* is defined by the equation (Bothell, 2004):

$$U_i(n) = U_i(n-1) + \alpha[R_i(n) - U_i(n-1)]$$

α is learning rate (set to .2)

$U_i(0)$ is set to 1000 for 'choose state' and 1 for 'choose prep'

$R_i(n)$ is the effective reward given to production *i* at time *n* calculated by subtracting the reward at time *n* minus the time since production *i* was selected

In order to even the playing field, in all cases the same amount of reward is triggered by the rewarding production (in this case we used a reward of 1). However, based on the current model's competing productions, it turns out that the major factor influencing how much reward each of the competing productions receives (and thereby alters its utility) is the time since the competing production fired compared to the reward production. The average difference between how long this interval was for 'change-row l->r' vs. 'move down' is 85ms. The average difference between how long this interval was for 'choose-state' vs. 'choose-prep' is 471ms. Over the course of the 80 trials, the model quickly learns the higher utility of using the 'move down' and 'choose prep' productions.

Figure 6 illustrates the average trial duration for both human and model data, which is a direct result of which of the competing productions are selected during a particular trial. Qualitatively, there is a learning curve for both humans and the model over the course of the first few trials, *RMSE* = 4.35s and the correlation coefficient is .21. The low level analysis of the time it takes both the model and the human participants to transition between consecutive cells in the grid indicates similar trends, *RMSE* = 131.74ms and the correlation coefficient is 0.28. Past work has addressed this low level analysis and for brevity only the fit is mentioned here (Veksler et al., 2007). Future work will need to address how to account for the remainder of the speedup seen in human data, perhaps as strategy shifts come into play later during the course of the experiment.

Model's Strategy – Accuracy Matching

The model just described was also outfitted with a simple heuristic in order to match human accuracy on the task. The strategy change that we implemented had to do with our analysis of cell clicks in the human data and the current task environment's setup. In particular, since we no longer had easy values in the cells of the grid, computing the normative value of an alternative is much more difficult than in our original task. Instead, given our human data analysis and how quickly participants were transitioning between cells in the grid, we suspected that rather than multiplying out the values and probabilities and summing these across the alternative, our participants were using a simpler heuristic to determine the best alternative.

This heuristic strategy was implemented in the model whereby as the model uncovered cell values, it simply kept a count in its imaginal buffer as to the number of cells in a particular row whose values exceeded some predetermined threshold value. Thus, rather than doing any sort of computation per se, the model was merely keeping count. At the end of a trial, the choice the model made was based on the alternative that it found to have the most cells above a threshold. If there were ties among alternatives, the more recent alternative looked at was chosen.

The implementation of this strategy also led to an important consideration – where to place the threshold. We explored the threshold parameter space in closed form to determine which threshold resulted in the best fit to human accuracy data. The procedure used is described below.

Threshold Consideration

A closed form model of the threshold parameter was developed to explore the model's accuracy given one of 35 threshold values (15 to 49). At first, 24 random 80-trial stimuli were used and run through each of the 35 threshold values and it was determined that a threshold value of 40 provided the best fit to average human performance, *RMSE* = 0.61. We then took all of the stimuli from the human participants (actual trials participants saw) and ran those through the model using the threshold of 40. Figure 7 depicts the fit of the model with a threshold of 40 to human data.

A 2x4 repeated measures ANOVA was conducted to compare human and model accuracy (type) with the repeated measure being gamble. There was not a significant gamble*type interaction, $F(3, 151) = 1.16$, $p = 0.33$. There was a significant main effect of gamble, $F(3, 151) = 143.65$, $p < 0.001$. There was not a significant main effect of type, $F(1, 151) = 0.08$, $p = 0.78$.

This analysis indicates that there was not a significant difference between human and model accuracy across the 4 gamble types. However, there was a significant difference between the gambles for both humans and the model.

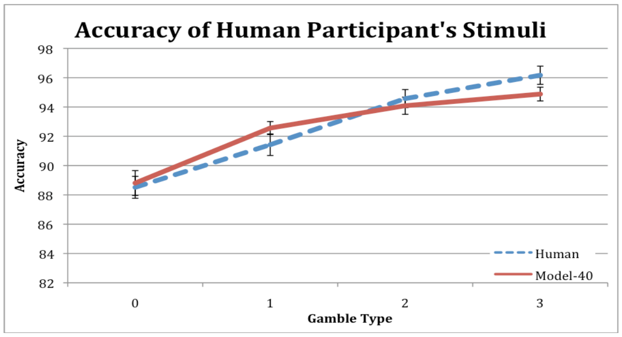


Figure 7: Accuracy comparison of model with threshold 40 across all 80 trials of human participant's stimuli. Error bars are standard error.

Conclusions

The current modeling work had a twofold purpose. The first was to demonstrate that the model could learn the cognitive, perceptual, motor interleaving resulting in the speedup in performance shown in previous work. The second was to implement a decision-making strategy that human participants most likely utilized in order to do the task.

Given the human data collected from a study of a decision-making table task, we found accuracy differences dependent on the constraints of the task environment (both lockout durations and types of gambles used). We also found that over the course of the 80 trials, participants completed trials considerably faster. The current model also completes the trials faster over the course of the task.

Furthermore, a more rigorous analysis of the human data indicated some biases in the way participants interacted with the task environment and we have implemented these biases in the strategy the model uses to complete the task. Namely, the model goes through the grid of cells in a top-down manner, and begins with the left-most column in the first row that it uncovers. In addition, the lack of a bias to click on the higher probability columns and the fact that gambles with higher dispersion values also had lower average scores, indicates that human participants tended to disregard the probability data, at least as far as the 0 second lockout group was concerned, and our model did as well. Future work will need to address how to reconcile this result with previous results of Payne et. al. (1988) in which it was found that probabilities played a role in decision strategies.

Future work will also incorporate the data we have from the other two conditions of the study as it relates both to strategy selection and timing. We also plan to further explore the factors influencing how quickly the model can perform the task as it seems human participants are nevertheless faster.

Acknowledgements

Thanks to Wayne Gray, Mike Schoelles, and Vladislav Veksler for many useful discussions regarding this ongoing project. The work was supported, in part, by grant N000140710033 to Wayne Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglas, S., Lebiere, C., & Quin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47*(6), 381-391.

Fu, W.-T. (2007). A rational-ecological approach to the exploration/exploitation trade-offs: Bounded rationality and suboptimal performance. In W. D. Gray (Ed.), *Integrated models of cognitive systems*. (pp. 165-179). New York, NY: Oxford University Press.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied, 6*(4), 322-335.

Lohse, G. L., & Johnson, E. J. (1996). A comparison of two process tracing methods for choice tasks. *Organizational Behavior and Human Decision Processes, 68*(1), 28-43.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction, 7*(1), 91-139.

Payne, J. W., Bettman, J. R., & Johnson, E. J. (1988). Adaptive strategy selection in decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*(3), 534-552.

Veksler, B. Z., Gray, W. D., & Schoelles, M. J. (2007). *From 1000ms to 650ms: Why Interleaving, Soft Constraints, and Milliseconds Matter.* Paper presented at the 8th International Conference on Cognitive Modeling, Ann Arbor, MI.

Un-learning *Un*-prefixation Errors

Ben Ambridge (Ben.Ambridge@Liverpool.ac.uk)
Daniel Freudenthal (D.Freudenthal@Liverpool.ac.uk)
Julian M. Pine (Jpine@Liverpool.ac.uk)
Rebecca Mills (md0u419e@student.Liverpool.ac.uk)
Victoria Clark (ps0u51a2@liverpool.ac.uk)
Caroline F. Rowland (Crowland@Liverpool.ac.uk)
School of Psychology, University of Liverpool, Eleanor Rathbone Building
Bedford St South, Liverpool L69 7ZA, UK

Abstract

A simple three-layer feed-forward network was trained to classify verbs as reversible with *un*- (e.g., *unpack*) reversible with *dis*- (e.g., *disassemble*) or non-reversible (e.g., *squeeze*), on the basis of their semantic features. The aim was to model a well-known phenomenon whereby children produce, then subsequently retreat from, overgeneralization errors (e.g., **unsqueeze*). The model learned to correctly classify both the verbs in the training set and verbs held back during training (demonstrating generalization). The model demonstrated overgeneralization (e.g., predicting *unsqueeze* for *squeeze*) and subsequent retreat, and was able to predict adult acceptability judgments of the different *un*- forms.

Keywords: Un-prefixation, overgeneralization; language acquisition; no negative evidence problem

Overgeneralization in Language Acquisition

A central question in the cognitive sciences is that of how children acquire their native language. Since speakers do not simply repeat whole utterances verbatim, the key question is how children are able to form the generalizations that allow for the production of novel utterances whilst avoiding *over*-generalizations (i.e., utterances that adults would consider ungrammatical).

One generalization that English-speaking children must acquire (presumably on the basis of hearing such forms as *unpack*, *unhook* and *unfold*) is that it is possible to add the prefix *un*- to a verb to specify the reversal of an action (i.e., they must acquire an *un-VERB* construction). This allows a child who hears (for example) the verb *fasten* to produce *unfasten,* even if she has never previously heard this form. Evidence that speakers do acquire a productive *un-VERB* construction (as opposed to simply learning all *un*- forms by rote) comes from overgeneralization errors attested in children's speech (e.g., *I'm gonna *unhang it*; *How do you *unsqueeze it*?; Bowerman, 1988).

Given that children do produce such errors, the challenge for acquisition researchers is to explain how children "un-learn" these overgeneralizations, whilst retaining the capacity for productive forms. Because children do not seem to receive feedback from caregivers when they produce overgeneralization errors, this has become known as the 'no-negative-evidence' problem (Bowerman, 1988).

One proposed solution is the 'entrenchment' hypothesis. This was originally developed for verb argument structure overgeneralization errors where a verb (e.g., the intransitive verb *disappear*) is overgeneralized into a construction (e.g., the transitive causative *SUBJECT VERB OBJECT* construction as in **The magician disappeared the rabbit*). The entrenchment hypothesis states that repeated presentation of a verb (e.g., *disappear*) in one (or more) attested construction (such as the intransitive construction; e.g., *The rabbit disappeared*) causes the learner to gradually form a probabilistic inference that adult speakers do not use that particular verb in non-attested constructions. A number of studies (e.g., Ambridge et al, 2008; in press; submitted) have shown that, as predicted by this hypothesis, speakers rate argument structure overgeneralization errors as less acceptable for high- than low frequency verbs (e.g., *disappear* vs *vanish*).

Whilst this proposal appears to work well for argument-structure overgeneralization errors, it is less clear that the account can be applied to morphological overgeneralization errors such as *un*-prefixation (Bowerman, 1988). A learning mechanism that deems *un*- forms ungrammatical when the observed frequency of the bare form (or the ratio of the bare to the *un*-prefixed[1] form) reaches a certain threshold would seem likely to make errors. For example, based on frequencies in the *British National Corpus*, a learner would have to hear around 500 occurrences of *twist* before encountering the (perfectly acceptable) form *untwist*. On the other hand, the non-reversible forms *embarrass* and *detach* each occur only around 500 times in the entire corpus. Worse still, some verbs are far more frequent in *un*- than bare form (e.g., *unleash* = 365; *leash* = 9).

An alternative proposal is that children use verb semantics to determine the syntactic and morphological constructions in which particular verbs can and cannot appear (e.g., Pinker, 1989; Ambridge et al, 2008; in press; submitted). For example, with reference to the intransitive/transitive causative alternation, Pinker (1989) proposed that children form narrow-range semantic classes of verbs that are restricted to the intransitive construction (e.g., verbs of ''coming into or going out of existence'' such as *disappear* and *vanish*). In support of this proposal, Ambridge et al (2008) found that when taught *novel* verbs of "coming into

[1] Here and throughout, '*un*-prefixed' means 'prefixed with *un*-' not 'with no prefix'

or going out of existence", both children and adults rejected (i.e., judged as ungrammatical) transitive causative uses.

Li and MacWhinney (1996) sought to extend this verb-semantics account to the domain of *un*-prefixation errors. Again, verbs that may appear in this construction appear to share certain meaning components such as *covering, enclosing, surface-attachment, circular motion* and *hand-movements*. Whorf (1956) argued that it is not possible to specify which verbs may and may not appear in the *un*-construction with reference to a list of necessary and sufficient semantic features (as Pinker, 1989, argued for verb argument structure constructions). Rather, these meaning components seem to combine interactively in a manner that is not straightforwardly predictable.

Li and MacWhinney (1996) developed a computational model designed to test Whorf's (1956) speculation that the *un*- construction constitutes a semantic "cryptotype". These authors trained a standard three-layer backpropagation network (with six hidden units) to produce an output of *un*- *dis*- or *zero*- (the three output units) for each of 160 English verbs (49 of which take *un*-, 19 –*dis* and 92 no prefix [termed "zero" verbs]). The model had 20 input units, each representing a particular semantic feature (e.g., *circular movement*; *change of state*). For each verb, the input to the model was a 20-bit vector representing the extent to which the verb was deemed to instantiate each of the semantic features (as rated by 15 adult participants). Verbs were presented to their model in proportion to their type and token frequency in a corpus of adult speech. The model's task was to learn to categorize each verb as (a) reversible with *un*-, (b) reversible with *dis*- or (c) non-reversible. The model performed reasonably well under a variety of different training regimes, correctly classifying between 50% and 75% of *un*- taking verbs (depending on the simulation).

It is important to note at the outset that Li and MacWhinney's (1996) model (like the model presented in the current paper) does not solve the no-negative-evidence problem. The pre-classification of verbs as *un*-, *dis*- or *zero* means that the model is given exactly the information that the child would need but does not receive (i.e., which verbs can and cannot be reversed). However, the model is valuable in that it demonstrates that, in principle, (a reasonable approximation of) the input available to children contains sufficient information to allow for the formation of a semantic "cryptotype" for the construction. For example, one strength of the model is that it uses this cryptotype to produce "overgeneralization errors" similar to those produced by children (e.g., **unhold*, **unpress*, **unfill*, **uncapture*, **unsqueeze*, **unfreeze*, **untighten*).

Nevertheless, Li and MacWhinney's (1996) model does exhibit a number of shortcomings. First, this model actually has great difficulty learning some forms. In the first simulation, the model learned to correctly classify (defined as an RMSE < .25) only 15% of the *dis*- verbs. In a second simulation, where *dis*- verbs were entered into the training set early in training, performance on *dis*- verbs improved. However, this was at the expense of the model's performance on the *zero* verbs (25% correct, vs 74% in Simulation 1) and *un*- verbs (51% correct, vs 76% in Simulation 1).

Second, this finding suggests that the particulars of the training regime may have been instrumental in shaping the particular pattern of results obtained. An incremental training regime was used such that the model was pre-trained on a set of 20 high frequency zero-verbs with verbs gradually added to the training set based on their type (*un*-, *dis*- or *zero*) and token frequency. The rate at which items were added furthermore changed during training. This incremental training regime was aimed at reflecting the realities of acquisition. While it has been shown that such manipulations may be crucial for successfully simulating developmental data (e.g. Elman, 1993), the very fact that they can influence results suggests that caution may be required when developing incremental training regimes.

A third shortcoming of Li and MacWhinney's (1996) model is that it actually lacks an important source of information that is available to children; namely, the distribution of surface forms. Reversible and non-reversible verbs differ not only in their semantics (information which is available to the model) but also their distribution: The former sometimes occur with *un-/dis-*, whilst the latter do not. Because the input to the model is simply a set of semantic vectors, this information is not available.

The final shortcoming of Li and MacWhinney's (1996) model is that it has great difficulty in retreating from overgeneralization errors. This would seem to be a consequence of the fact that the model produces overgeneralization errors in a way that is quite different to children. The model's overgeneralization errors result from mis-classification of items (e.g., *squeeze* is incorrectly classified as an *un*- verb, presumably because it shares a number of semantic features with genuine *un*- verbs). The model has great difficulty in re-classifying such verbs correctly (presumably because much of the semantic overlap that caused the erroneous classification remains even after learning has reached asymptote). Intuitively, it would seem that at least some of children's overgeneralizations are caused not by misclassification, but by *functional pressure*: Presumably, children produce forms such as **unsqueeze* because they want to denote the reversal of (in this case) a squeezing action, have learned that the *un*- prefix serves this function and do not have an alternative form that expresses the required meaning. Later in development, children are able to avoid producing *un*- forms for verbs such as *squeeze*, even when they are under functional pressure to do so (note, however, that even adults occasionally produce forms that they would probably regard as "overgeneralizations" in such circumstances; as in the form **unlearn*, which appears in the title of this paper). Li and MacWhinney's (1996) model does not simulate this situation as it is never 'asked' to produce a reversed (or non-reversed) form of a particular verb, as required for the discourse context; verbs are simply probabilistically assigned to one of three categories.

Our goal in the present study was to address these shortcomings with a new version of the *un*-prefixation model. This model differs from that of Li and MacWhinney in a number of important ways. First, the model was trained using a regime that more accurately reflects the frequency of individual forms in the input. This allows us to achieve more accurate classifications, whilst avoiding the need for discontinuities in the training regime.

Second, we aimed to determine whether a model trained on the semantic features of a subset of the verbs is able to successfully generalize its acquired structure to novel items when presented with their semantic features. Although the ability to generalize will be a crucial feature of any model of this phenomenon, no such test was conducted by Li and MacWhiney (1996). This test is crucial in determining whether a semantics-based model can account not only for the retreat from overgeneralization errors, but also for the formation of the generalizations that allow for such errors (and correctly produced novel forms) in the first place.

Third, the new model was designed to simulate not only overgeneralization - which was observed in Li and MacWhinney's study - but also, crucially, the *retreat* from overgeneralization, which was not. This was achieved by including in the input signal a 'reversative feature', which was switched on for reversed forms and off for base (non-reversed) forms. The model was trained on reversible items in both their base (e.g., *pack*, *appear*) and reversed forms (e.g., *unpack*, *disappear*). For example, the set of semantic vectors representing the verb *pack* was trained with the reversative feature off (corresponding to presentation of *pack*) for some trials and on (corresponding to presentation of *unpack*) for others. This feature makes it possible to explicitly 'ask' the model to produce a reversed form for verbs that were never presented in this form during training. This maps closely onto the scenario where children produce overgeneralization errors (e.g., to denote the reversal of a squeezing action) and hence allows us to model both overgeneralization and the *retreat* from overgeneralization in a realistic way. The inclusion of this feature has two further advantages that would seem likely to facilitate learning and generalization. First, it makes it possible to present reversible verbs to the model with the relative frequencies of the reversed and non-reversed forms in speech to children. Second, the information that a verb has occurred in reversed form constitutes a powerful cue that the verb (or collection of semantic features) is indeed reversible.

The final advantage of the new model is that it allows us to simulate adult acceptability judgment data. The inclusion of the reversative feature means that the output (i.e., the activation of the *un-/dis-* units) of the model when asked to produce a reversative form for a verb never presented in this form during training (e.g., *squeeze*) can be taken as analogous to an "acceptability judgment" for the reversed form (e.g., *unsqueeze*). This makes it possible to evaluate the model's performance in a very fine-grained way, by investigating whether its "acceptability ratings" of the various verbs in *un-* form correlate with adults' judgments.

Method

Our learning task was designed to more closely mirror that faced by real learners. In particular, our models were trained on both the base form and the reversed form of reversible verbs. The simulation used the same set of 160 verbs used by Li and Macwhinney (1996), pre-classified as *un*-taking (*N*=40), *dis*-taking (*N*=19) or *zero* (*N*=92). The input to the model consisted of the 20-bit semantic vector employed by Li and Macwhinney (whom we thank for making these data available to us) as well as a one bit 'reversative' feature. The reversative feature was set to 0 when a verb was presented in its base form, and to 1 when a verb was presented in its reversed form (either *un-* or *dis-*). The model had three output units, one for each of the three prefixes 'zero' 'un' and 'dis', and six hidden units. The task of the model (during training) was to predict whether each verb was a *zero* verb, an *un-* verb or *dis-* verb. Training items were presented in their base- (i.e., with the reversative feature off) and reversed forms (i.e., with the reversative feature on) relative to their (log) frequency in the British National Corpus (BNC). For example, the model was presented with *fasten* (BNC frequency 667) both in its base form (i.e., with the reversative feature off) and in its reversed form (i.e., with the reversative feature on; BNC frequency of *unfasten* = 97). In both cases, the "correct" activation pattern of the *un-*, *dis-* and *zero* output units (for the purposes of backpropogation) was 1 0 0 (i.e., activation of the *un-* unit only). Likewise, zero verbs (which take neither *un-* nor *dis-*) were never presented with the reversative feature during training. The formal classification of items as *zero, un-* or *dis* was the same as that used by Li and MacWhinney (which was determined by adult raters). Whilst this classification can on occasion clash with BNC usage, this often represents cases where a prefixed form does not in fact represent the reversal of an action (e.g., *disapprove* has a meaning that is opposite to that of *approve*, but does not denote the reversal of this action). Thus we decided to respect the classifications of the adult raters, rather than determining classifications on the basis of corpus usage.

During testing, the model was presented with the training set with the reversative feature switched on for all items. The activations of each of the three output units were then read off. For *un-* and *dis-* verbs, the reversative feature had occasionally been switched on during training (and was always associated with a target of *un-* or *dis-*). For zero verbs, which had never been paired with the reversative feature this was a novel situation. This corresponds to a scenario in which a human learner is attempting to produce a reversative form of a verb never encountered in this form (e.g., *squeeze*) or judge the acceptability of a reversed form offered by an experimenter (e.g., **unsqueeze*). Early in development, children are quite willing to produce overgeneral forms like **unsqueeze*, before learning to reject them later on. In these simulations, the relative activation of the *un-* and *dis-* output units was taken to reflect the model's acceptability rating of these forms.

The model was implemented using LENS, with all parameters set to their default values. The model was trained for a total of 100,000 trials (with one verb presented each trial) and tested after every 5,000 trials. Individual forms were included in the training relative to their log frequency in the *British National Corpus*. The order of presentation of items was randomized.

Results

Classifying verbs in the training set

The first simulation was designed to investigate the model's ability to correctly classify the training items. In this simulation, an item was considered correctly classified if the activation of the target output node exceeded 0.7. The results for this simulation (averaged over 5 runs of the model) are depicted in Fig. 1.

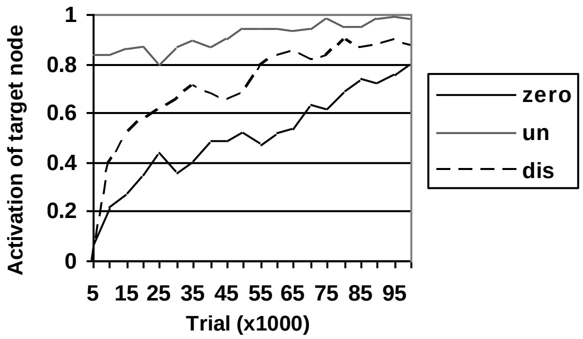


Fig.1: Proportion of correctly classified forms

As can be seen in Fig.1, the model is capable of correctly classifying an increasing number of items with increased training. Learning is particularly fast for the *un-* verbs, followed by the *dis-* verbs, and is slowest for *zero* verbs. However, even for the *zero* verbs, the model learns to ignore the fact that the reversative feature has been switched on (recall that the reversative feature is always switched on during testing). Thus despite the fact that the reversative feature was *always* associated with activation of either the *un-* or *dis-* output unit (and never the *zero* unit) during training, the model learns to correctly map 80% of *zero* verbs to the *zero* output unit when the reversative feature is switched on at test. This can be thought of as analogous to a child refusing to produce a form such as **unsqueeze* despite being under functional pressure to do so (or rating such a form as ungrammatical).

Generalization

Generalization – the ability to apply previously acquired "rules" or patterns to new items – is a key aspect of human linguistic competence. Given the semantics of novel verbs, both adults and children are able to determine whether or not this verb can be used in a particular construction (Ambridge et al, 2008; in press; submitted). (It is worth noting in passing that such findings are problematic for a purely statistical entrenchment account). Although we are aware of no studies that have investigated this phenomenon with regard to *un*-prefixation, it is reasonable to suppose that adults would be able to generalize in this way.

The second simulation was therefore designed to investigate the model's ability to generalize the knowledge it has extracted from the training set to novel items. This was done by removing 25% of the items from the training set (a different random set was held out for each of five runs). Testing then took place only on the items that were held out during training. Fig 2 shows the performance (average activation of the correct output node) of the model for these items, averaged over the five runs. As with the previous simulations, the model was trained for 100,000 trials. As can be seen from Fig. 2, the model is successful in generalizing its acquired knowledge to all three classes.

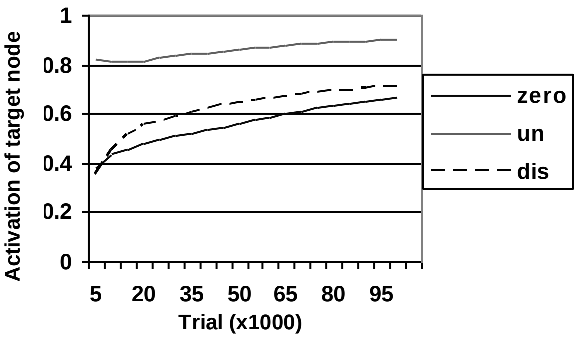


Fig. 2: Performance of the model on novel items.

Retreat from Overgeneralization

While the data presented thus far demonstrate that the model is capable of generalization, they do not demonstrate that the model – like children – produces, and then retreats from – overgeneralizations.

The data presented in Fig 1 suggest that the model produces overgeneralization errors, in that many *zero* verbs are incorrectly classified as *un-* or *dis-* verbs until relatively late in training. Nevertheless, this pattern is not necessarily indicative of overgeneralization behaviour. Even if a large percentage of *zero* verbs are not classified as such by a .70 criterion, it does not necessarily follow that the model is willing to overgeneralize on these items. For example, a verb activating the *zero* unit at 0.6 and the *un-* and *dis-* units each at 0.2 would be said to have failed in classifying the verb as a *zero* verb, but it would be odd to claim that the model was overgeneralizing the verb to *un-/dis-*. In order to more closely determine the model's willingness to over-generalize, we determined which output node showed the highest activation level for each of the *zero* verbs (for the simulation in which no verbs were held out). The results of this analysis are shown in Fig. 3. Early in training the *zero* node is most active for about 45% of *zero* verbs. Thus, when the reversative feature is switched on, the most active node is the *un-* or *dis-* node for 55% of zero-verbs (i.e, the model can be said to overgeneralize 55% of *zero* verbs to either *un-* or *dis-* when under functional pressure to do so). This decreases to around 10% at the end of training. Thus the model can be said to show the retreat from overgeneralization that is characteristic of children's learning.

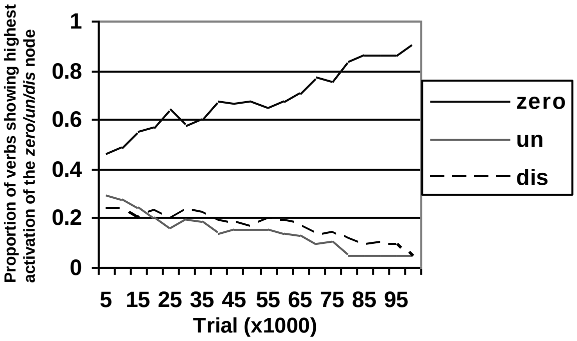


Fig. 3: Most active nodes for presentations of *zero* forms.

There are two possible reasons for the model's successful simulation of the pattern of overgeneralization on *zero* verbs. The theoretically interesting possibility is that this is caused by the presence of the reversative feature at test. On this explanation, it is the functional pressure of 'wanting' to reverse a verb (rather than an incorrect classification) that causes the model to overgeneralize (as we would argue is the case for children). A less interesting possibility, however, is that the class of *zero* verbs may simply be difficult for the model to learn (for example, it may be that *zero* verbs form a class that is less semantically cohesive than either *un*- or *dis*- verbs). This may cause the model to incorrectly classify *zero* verbs as either *un*- or *dis*- verbs. Indeed, misclassifications were the major cause of overgeneralization errors in Li And MacWhinney's (1996) simulations.

This possibility was tested by re-running our first simulation (with no items held out), with the modification that the reversative feature was switched on (when relevant) during training, but not at test, thus providing a baseline measure of the model's ability to classify items into the correct category. As in the first simulation, an item was considered correctly classified when the activation on the target node exceeded 0.7. The results of this analysis are shown in Fig. 4. As this figure demonstrates, the model is actually very successful in learning the zero-class. Thus, after a mere 5,000 trials, the model correctly classifies 75% of the zero-verbs.

These data suggest that the cause of the model's overgeneralizations is not the fact that the model incorrectly classifies many of the zero-verbs (though it may incorrectly classify some). Rather (as we would argue happens with children) the functional pressure to produce a reversative form (as instantiated in the model with the reversative feature) overrides the semantics of the *zero* class. With increased training the model (like children) learns to ignore this pressure and retreats from overgeneralization.

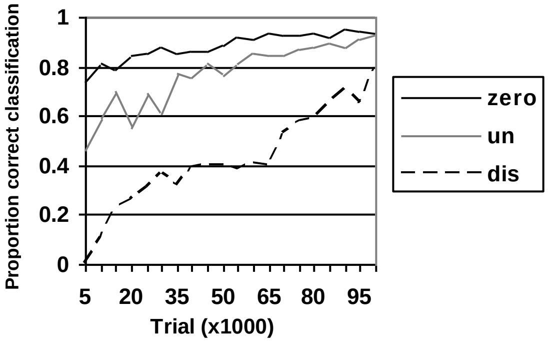


Fig. 4 Proportion of correctly classified items in the absence of the reversative feature.

Modeling Adult Acceptability Judgments

The data presented thus far show that the model displays a pattern of learning, generalization, overgeneralization and retreat from overgeneralization that is generally similar to that shown by children. In order to determine if the behaviour displayed by the model matches human behaviour more closely, we assessed the extent to which the model can simulate adult acceptability judgments.

Acceptability judgments of the base and *un*- form of each of the 160 verbs were obtained from 20 adult speakers of (British) English. Forms were presented in sentences with two different versions counterbalanced across participants. For example, 10 participants rated *Lisa bandaged her arm* and *Lisa unbandaged her arm* whilst 10 rated *Marge bandaged her friend's leg* and *Marge unbandaged her friend's leg*. Looking across all verbs, the correlation between the two sets was *r*=0.76 for the *un*-prefixed forms and *r*=0.55 for the base forms (both significant at *p*<0.001). This represents a reasonable upper-bound when assessing the model's ability to predict the human acceptability judgments.

In order to determine how well the simulation modeled the adult acceptability judgments, the mean adult judgments of (sentences containing) the *un*- forms were correlated with the model's activation of the *un*- node in the output bank (after 25,000 trials). Across all 160 verbs the correlations ranged from .68 to .73 for the five different runs of the model (all highly significant at *p*<0.001). Thus the model simulates to an impressive extent adults' ratings of the relative (un)acceptability of different *un*- forms.

This high correlation reflects the fact that adult judgments of overgeneralization errors are not binary but graded. Verbs that are highly incompatible with the semantic cryptotype for the construction (e.g., *talk*) are rated as extremely ungrammatical with *un*- (all raters gave **untalk* the lowest possible rating of 1/5). Non-reversible verbs that are, nevertheless, less semantically incompatible with the cryptoptype receive higher acceptability ratings (e.g., **unturn* = 1.67/5), whilst still being rated as unacceptable.

Indeed, even when looking only at the non-reversible (i.e,, *zero*) verbs, the model was able to predict the *extent* to which adults would consider the *un-* forms to be acceptable (notwithstanding the fact that all were, to some degree, unacceptable). Although the correlations were low (range .20-30) they were statistically significant for four of the five runs (*p*'s 0.01 – 0.05) and borderline for one (*p*=0.053). This is an important finding as the correlations calculated across all verbs will be somewhat inflated by the fact that verbs naturally cluster into two types: verbs that are reversible with *un-* and those that are not. Thus the adults and model could show a high correlation simply by rating the *un-* forms of all *un*-verbs as maximally acceptable (5/5 for adults, 1.0 *un-* unit activation for the model) and the *un-* forms of all *zero* and *dis-* verbs as maximally unacceptable (1/5 and 0.0). The fact that significant correlations between the predicted and actual acceptability of *un-* forms was observed, *when looking only at verbs that are not reversible*, demonstrates that the correlation observed was not simply an artefact of the fact that the verbs can be divided into two classes (*un*-taking and not-*un*-taking).

No significant model-adult correlations were found for acceptability ratings of the *un-* form of verbs that do take *un-* (i.e., where the *un-* form *is* acceptable, the model cannot predict the relative acceptability of the different un- forms). However, this is probably simply due to the fact that there is little relative acceptability (i.e., little variance) to explain, with most forms being rated as close to 5/5 (*M*=4.41, *SD*=0.76). The only *un*-taking verbs that received *un-* form ratings lower than 4/5 were *unarm, undelete, unmask* and *unscramble* (with the first two probably representing misclassifications). In any case, this issue is irrelevant to the question of the retreat from overgeneralization, as all these *un-* forms were acceptable (indeed, all had been encountered by the model and, presumably, the adults).

Discussion

The aim of the present study was to replicate and extend Li and MacWhinney's (1996) simulation of children's learning of *un-* prefixation. Specifically, we sought to implement a more plausible training regime in which both non-reversed and (where appropriate) reversed *un-/dis-* forms were presented in proportion to their frequency in a representative corpus. Another innovation was the introduction of a functional 'probe' for the reversative form which allowed us to investigate children's overgeneralization errors, and the retreat from such errors, in a more plausible way.

The first point to note is that the present model actually displayed better learning of the training set than Li and MacWhinney's (1996) original model. Thus we can be confident that the success of the previous model did not depend on unrealistic assumptions concerning the input or learning task, as a version of the simulation with (we would argue) more realistic assumptions actually performed better. The two key improvements would seem to be the more realistic training regime (including presentation of both reversed and non-reversed forms) and the presence of the reversative feature, which helps the model distinguish between reversible and non-reversible forms.

In addition to improved learning of the training set, the model was able to demonstrate generalization, overgeneralization and subsequent retreat from overgeneralization in a way that maps onto reports of children's performance. More impressively, the model was able to predict the relative (un)acceptability of the different *un*-prefixed forms as determined by adult raters.

With regard to theories of acquisition, the model adds to a growing body of evidence which suggests that pure statistical learning cannot explain how children form and retreat from grammatical (over)generalizations. Instead, what seems to be required is an account in which probabilistic learning of the semantics of particular verbs and constructions plays a key role (e.g., the *ILVACS* account of Ambridge et al, in press).

Of course, this model as it currently stands does not solve the 'no-negative-evidence' problem. To do so a model would need to determine which verbs are non-reversible or reversible with *un-* or *dis-*, without being given this information in the form of the correct output activation pattern. Such a model would likely need a more complex architecture than the simple feed-forward network used here. Nevertheless, the present set of simulations has demonstrated that a model that uses verb semantics to probabilistically learn verbs' argument-structure and morphological privileges is on the right tack with regards to solving the 'no-negative-evidence' problem.

References

Ambridge, B., Pine, J. M., Rowland, C. F., Jones, R., & Clark, V. (in press). A semantics-based approach to the 'no-negative-evidence'problem *Cognitive Science*.

Ambridge, B., Pine, J. M., Rowland, C. F., & Clark, V. (submitted). Restricting dative argument-structure overgeneralizations: A grammaticality-judgment study with adults and children. *Language*

Ambridge, B., Pine, J. M., Rowland, C. F., & Young, C. R. (2008). The effect of verb semantic class and verb frequency (entrenchment) on children's and adults' graded judgments of argument-structure overgeneralization errors. *Cognition, 106*(1), 87-129.

Bowerman, M. (1988). The "no negative evidence" problem: how do children avoid constructing an overly general grammar? In J. A. Hawkins (Ed.), *Explaining language universals* (pp. 73-101). Oxford: Blackwell.

Li, P., & Macwhinney, B. (1996). Cryptotype, overgeneralization, and competition: a connectionist model of the learning of English reversive prefixes. *Connection Science, 8*, 3-30.

Pinker, S. (1989). *Learnability and Cognition: The Acquisition of Argument Structure.* Cambridge, MA: MIT

Whorf. B. K., (1956) *Language, Thought, and Reality. Selected Writings of Benjamin Lee Whorf*. John B. Carroll, ed. New York: Wiley

Modelling the Dynamics of Cognitive Depressogenic Thought Formation

Azizi A. Aziz (mraaziz@few.vu.nl)
Agent Systems Research Group, Department of Artificial Intelligence
Faculty of Science, Vrije Universitiet Amsterdam,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Abstract

Cognitive vulnerabilities provide a clear link of how individuals are exposed to the elements of the risk in hopelessness and later to the formation of recurrence and relapse in depression. It has also been associated with the negative social support and inferential styles. Therefore, it is crucial to understand how these concepts are interrelated, and defined. This paper presents a model of the dynamics of a human's developmental state in relation to the social support and negative cognitive thought formation (cognitive depressogenic thought). Theory in cognitive vulnerability is used to serve as a foundation of this model. Simulation experiments under different parameter settings pointed out that the model is able to produce related behaviour as described in several literatures. In addition, using a mathematical analysis, the equillibria of the model has been determined and analyzed.

Keywords: Risk of Relapse and Recurrence in Depression, Hopelessness, Cognitive Depressogenic Thought, Cognitive Vulnerability.

Introduction

Cognitive vulnerability is one of the main concepts that play an important role to escalate the risk of relapse in affective disorder (depression). In a broader spectrum, it is a defect belief, or structures that are persistently related for later emergent in psychological problems. Therefore, by understanding this vulnerability and ways to overcome it, a risk of relapse or recurrence in depression can be reduced. Before further reviewing the underlying concepts of the vulnerability, it is essential to understand its connection between relapse condition in unipolar depression and social support. Unipolar depression is a mental disorder, distinguished by a persistent low mood and loss of awareness in usual activities (Segal, et al. 2003). Normally, under a certain degree of stressors exposure, an individual with a history of depression will develop a negative cognitive content (thought), associated with the past losses. Such cognitive content is often related to the maladaptive schemas, which in a long run will cause individual's ongoing thought capability to be distorted and later to be dysfunctional (Beck, 1987).

However, this cognitive distortion can be reduced through appropriate supports from other members within the social support network (Roberts & Gotlib, 1997). Social support network is made up of friends, family and peers. Some of it might be professionals and support individuals in very specific ways, or other people in this network might be acquaintances in contact with every day (Heller & Rook, 1997). It has been suggested that social support naturally can help to prevent and decrease stress through positive inferences, which later curbs the formation of cognitive vulnerability. However, some literatures have shown that certain supports provide contrast effects (Coyne, 1990; Panzarella & Alloy, 1995; DeFronzo et al., 2001). Rather than attenuating the negative effects from stressors, it will eventually amplify the individual's condition to get worse.

In this paper, these positive and negative effects from social support interaction and its relation with cognitive thought are explored. To fulfil this requirement, a dynamic model about cognitive depressogenic thought is proposed. The proposed model can be used to approximate a human's cognitive depressogenic thought progression throughout time. This paper is organized as follows. The first section introduces main concepts and existing theory of cognitive depressogenic thought and hopelessness. Thereafter, a formal model is described. The model has been simulated and later followed by a mathematical analysis. Finally, conclusion summarizes the paper with a discussion and future work for this model.

Underlying Concepts in Cognitive Depressogenic Thought

Although it is well documented that social support mitigates a risk of relapse, but there is a condition where feedbacks from the social support members may indirectly escalate the risk of relapse (DeFronzo et al., 2001). Such feedbacks are considered as "maladaptive inferential feedback" (MIF), and normally increase the negative thought formation. Prolong exposure towards this effect will later develop a serious cognitive vulnerability. Contrary to this, an adaptive inferential feedback (AIF) provides a buffer to reduce the threat, by countering negative inferences for negative event (DeFronzo et al., 2001). AIF asserts that when a social support member offers comfort by attributing the source of negative event to be unstable, or implies that event directs neither negative consequence (characteristics) towards that individual, it will later diminish the risk of creating maladaptive inferences.

These conditions also can be explained through the Expanded Hopelessness Theory of Depression. It elaborates the possibility of social processes with the presence of negative cognitive thought, and stress will later contribute to the development of vulnerability towards depression (Dobkin et al., 2004; Panzarella et al., 2006). Major focus of this theory is the specific mechanisms which inferential feedback (both AIF and MIF) may influence the development of hopelessness, cognitive depressogenic thought, and later vulnerability in depression. However, this paper will be focusing more to the formation of cognitive depressogenic thought while retaining important aspects of theory.

According to Alloy et al. (1999), there is an evident to show that individuals response differently towards stressful life events. Some individuals may develop severe or long lasting depression, while others stay healthy or develop mild and short-lived depression. This is the result from individuals' interpretation towards their experience influences over the negative event, resulting from the formation of cognitive depressogenic thought. Cognitive depressogenic thought refers to the negative style of thinking, characterized by a tendency to attribute negative events to be persistent and widespread in many aspects of life (Abramson et al., 1999; Alloy et al., 2004). Individuals with this condition are likely to infer the negative life events as self-attributions of being worthless and flawed. As a result, these particular individuals are exposing themselves towards vulnerability of recurrence or relapse in depression.

The Expanded Hopelessness Theory of Depression relates the development cognitive depressogenic thought through two precursors. First, the present of positive social support feedback (AIF) acts as a buffer to reduce individuals' possibility of having cognitive depressogenic thought over time. Second, individuals with cognitive depressogenic thought will make negative inferences when facing negative events. This condition is also associated with less AIF from the social support members (Panzarella et al., 2006). Moreover, both of these conditions capable to predict changes in stressful events. Therefore, it can be further used to elaborate the immunity level of individuals (as contrast in vulnerability concept). In addition, many studies have also associated the lower risk of depression with the presence of AIF (Alloy et al., 2000; Crossfield et al, 2002).

As indicated in several previous works, inferential feedbacks provide one of the substantial factors towards the development of cognitive depressogenic thought over time. By combining either one of these two factors together with situational cues, it leads to the formation of either cognitive depressogenic inference or positive attributional style. Situational cues refers to a concept that explains individuals' perception that highly influenced by cues from events (environment). Individuals under the influence of negative thought about themselves will tend to reflect these negative cognitions in response to the occurrence of stressors. These later develop the conditions called "stress-reactive rumination" and "maladaptive inference". Stress reactive rumination reflects a condition where individuals have difficulty in accessing positive information, and further develop a negative bias towards inference (maladaptive inference) (Spasojevic & Alloy, 2001; Robinson & Alloy, 2003). This process is amplified by previous exposures towards cognitive depressogenic thought episode. After a certain period, both conditions are related to the formation of hopelessness. Hopelessness is defined by the expectation that desired outcome will not occur, or there is nothing one can do to make it right (Abramson et al., 1989). Prolong and previous exposure from hopelessness will lead to the development of cognitive depressogenic thought. However, this condition can be reduced by having a positive attributional style, normally existed during the presence of AIF and low situational cues perception (Dobkin et al., 2004).

In short, the following relations can be identified from the literature: *(1)* prolong exposure towards MIF, negative events, and high-situational cues can lead to the development of cognitive depressogenic thought. *(2)* a proper support (AIF) will reduce the risk of further development of future cognitive depressogenic thought. *(3)* Individuals with high situational cues and proper support will be less effective in reducing the progression of cognitive depressogenic thought, compared to the individuals with less situational cues.

Modelling Approach

This section discusses the details of the dynamic model. The characteristics of the proposed model are heavily motivated by the research discussed in the previous section. In this model, three major components will represent the dynamic of interactions between social support feedback and individuals involved in negative thought formation during the brink of relapse and recurrence in depression. These components are; environment, inferential feedbacks, and thought formation. Environment explains the condition of stressors, while inferential feedbacks represent the inferential style communicated by the social support members to the individuals and, finally thought formation summarizes the interaction results from those conditions. By coupling these main concepts, it provides a building block in designing an individual model for cognitive depressogenic thought dynamics. Figure 1 depicts the relationship between the details of these components.

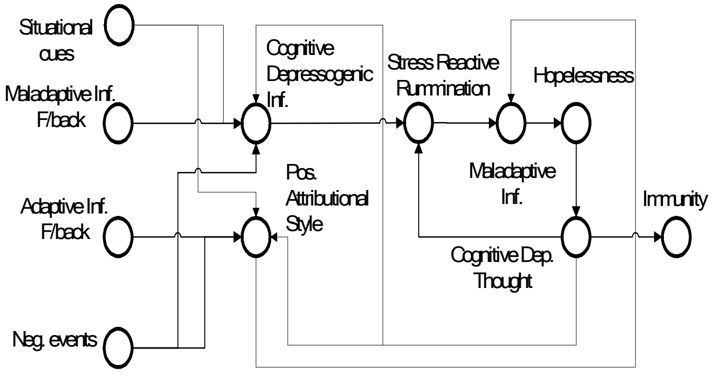


Figure 1: Model of Cognitive Depressogenic Thought Dynamics

Once the structural relationships in the model have been determined, then model later can be formalized. During the formalization process, all nodes are designed in a way to hold values ranging from 0 (low) to 1 (high). Interaction among interrelated nodes will determine the new value of it, either by a series of accumulative or instantaneous effects. The following explains the detail of the model.

Negative events (*NEVt*): In the model, the negative events are generated by simulating several dynamic *t* time conditions using weighted sum *w*, of major events; life events (*Le*), chronic (*Ce*), and daily (*De*) events.

$$NEVt(t) = w_1.Le(t) + w_2.Ce(t) + W_3.De(t)\ , \quad \sum_{n=1}^{i} w_n = 1\ , i = 3$$

In this case, the role of these events is to represent the condition of stressors. These events are seen as very intense (high negative event) when *NEVt(t)* → *1*, and less-intense when *NEVt(t)* → *0.*

Situational cues (*SiC*): Situational cues are computed by combining three factors together; consistency (*CtC*), consensus (*CsC*), and distinctiveness (*DtC*) cues. Higher situational cues represent a condition where an individual will behave according to the external environment rather than individual's intellect or dispositional.

$$SiC(t) = \varpi_1.CtC(t) + \varpi_2.CsC(t) + \varpi_3.DtC(t), \quad \sum_{n=1}^{i} \varpi_n = 1\ , i = 3$$

Cognitive depressogenic inferences (*CDi*) explains the combination of a maladaptive inferential style (*MiF*) with several components, namely; situational cues (*SiC*), cognitive depressogenic thought (*CdT*), and negative events (*NEVt*). The α value is used to distribute the proportion of contributions among these variables in this equation.

$$CDi(t) = \alpha.MiF(t) + (1-\alpha).\ [SiC(t).CdT(t).NEvt(t)].\ MiF(t)$$

Positive attributional style (*PtS*) is an attributional style that is highly related to an adaptive inferential style (*AiF*). It also has a negative relationship with bad situational cues, negative events, and cognitive depressogenic thought.

$$PtS(t) = \eta.AiF(t) + (1-\eta).\ [1-(SiC(t).\ NEvt(t).CdT(t))].AiF(t).$$

Stress reactive rumination (*SrR*) is based on the interaction between cognitive depressogenic inference and cognitive depressogenic thought. Parameter β is used to regulate the contribution of these variables.

$$SrR(t) = \beta.CDi(t) + (1-\beta).\ CdT(t)$$

Maladaptive inference (*MdI*) has a positive relationship with the stress reactive rumination, and contrary for the positive attributional style. This opposite effect reflects the condition of stress buffering concept delivered by positive social support feedbacks. The intensity of this inference process is controlled by parameter γ.

$$MdI(t) = \gamma.SrR(t).(1-PtS(t))$$

Hopelessness (*Hps*) and Cognitive depressogenic thought (*CdT*) are derived from the accumulative (temporal relation) process of certain cases in a time interval between *t* and *Δt*. Hopelessness relates with the formation of maladaptive inference, while the hopelessness is related to the development of cognitive depressogenic thought. These relationships are formulated as the following;

$$Hps(t+\Delta t) = Hps(t) + (1-Hps(t)).\ \psi.(MdI(t)-\phi.Hps(t)).\ Hps(t).\Delta t$$

$$CdT(t+\Delta t) = CdT(t) + (1-CdT(t)).\ \varphi.(Hps(t)-\tau.CdT(t)).\ CdT(t).\Delta t$$

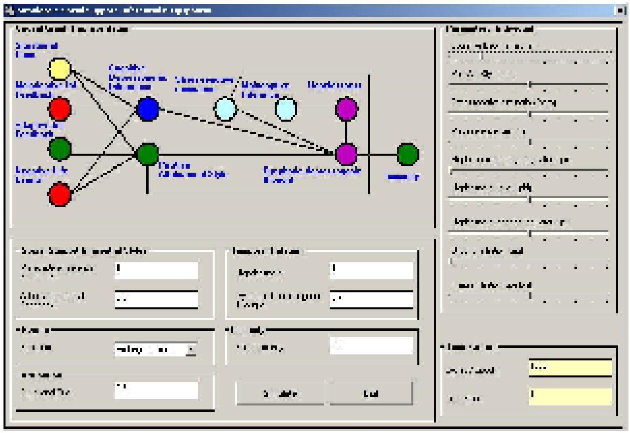
where ψ, φ, ϕ, and τ denote the proportion of changes for all respective equations.

Immunity (Im) has a negative relationship with the formation of cognitive depressogenic thought. The value of Υ provides the proportional rate of the contribution between based-immunity (*IMnorm*) and cognitive depressogenic thought. *IMnorm* represents the baseline immunity for each individual.

$$Im(t) = \Upsilon.\ Im_{norm} + (1-\Upsilon).(1-CdT(t)).\ Im_{norm}$$

Using all defined equations, a simulator has been developed for experimentation purposes, specifically to explore interesting patterns on inferential feedbacks and depressogenic thought. Figure 2 depicts the screenshot of the simulator.

Figure 2: A Screenshot for the Developed Simulator



This simulator is designed and developed under a visual programming platform. It allows a graphical user interface for experimental and parameters settings purposes.

Simulation Traces

In this section, the model was executed to simulate several conditions of individuals with the respect of exposure towards negative events, feedbacks from the social support members, and situational cues. With variation of these conditions, some interesting patterns can be obtained, as previously defined in the earlier section. For simplicity, this paper shows several cases of cognitive depressogenic thought levels formation using three different individual attributes. These cases are; (i) an individual ***A*** with a good feedbacks from the social support members, and using a good judgment about the situation, (ii) an individual ***B*** that receives good feedbacks but with bad judgment about the situation, and (iii) an individual ***C*** with bad feedbacks from the social support, and bad judgment about the situation.

Table 1: Individual Profiles

Individual	**Parameters Setting**
A	*SiC=0.2, MiF=0.1, AiF=0.8*
B	*SiC=0.8, MiF=0.1, AiF=0.9*
C	*SiC=0.9, MiF=0.8, AiF=0.1*

The duration of the simulated scenario is up to *t* = 1000 (to represent the conditions within 42 days) with three negative events. The first event consisted of the prolonged and gradually decreased stressors, while the second event dealt with the decreased stressor. The third event simulates the repeated stressors. For all conditions, the initial cognitive depressogenic thought was initialized as 0.5.

Case # 1: Prolonged Repeated Stressor with Different Individuals Inferential Feedback and Situation Cues

During this simulation, each type of individual attribute has been exposed to a prolonged stressor condition. The result of this simulation is shown in Figure 3.

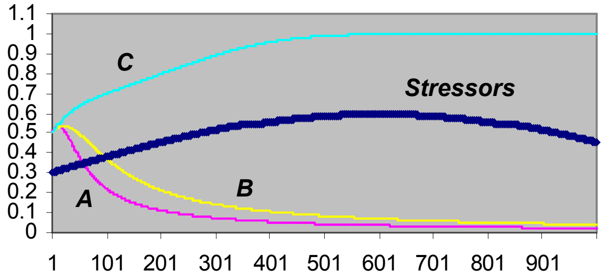


Figure 3 Cognitive Depressogenic Level for Each Individual during Prolonged Stress

In this simulation trace, it shown that an individual *C* (*high situational cues,* and *negative inferential feedback*) tends to develop a cognitive depressogenic thought, in contrast with the others. Individual *A* (*low situational cues,* and *positive inferential feedback*) shows a rapid declining pattern in developing the cognitive condition. Note that the individual *B* (*high situational cues* and *positive inferential feedback*) has also developed a decreasing pattern towards the cognitive condition. However, the individual *B* has a lesser decreasing effect towards a negative thought despite a high positive support, given that this individual tends to perceive negative view about the situation. Persistent positive support from the social support members helps him/her to reduce the development of cognitive thought throughout time.

Case #2: Decreased Stressor with Different Individual Inferential Feedback and Situational Cues

In this simulation trace, there are two conditions were introduced, one with a very high constant stressor, and with no stressor event. These events simulate the condition of where individuals were facing a sudden change in their life, and how inferential feedbacks and perceptions towards events play important to role towards the diminishing of cognitive thought. The result of this simulation is shown in Figure 4.

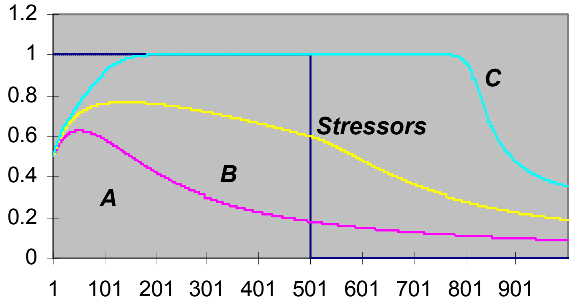


Figure 4: Dysphoric Depressogenic Level for Each Individual during Fluctuated Stressors

A comparison for each individual shows that individual *C* gets into a sharp progression towards a high cognitive thought after direct exposure towards a heighten stressor. At the start of a high constant stressor, both individuals *A* and *B* develop cognitive thought. However, after certain time points, those progressions dropped and reduced throughout time. As for the individual *C*, even the stressors have been diminished, the level cognitive depressogenic thought was still high for several time points until it decreased.

Case # 3: Rapid Repeated Stressors with Different Individual Inferential Feedback and Situational Cues

For this simulation, each type of individual has been exposed to a stream of repeated stressors, with a rapid alteration between each event. In a real situation, it simulates the cummulative effect conditions, where repeated strikes had the effect of escalating the overall intensity of stressors.

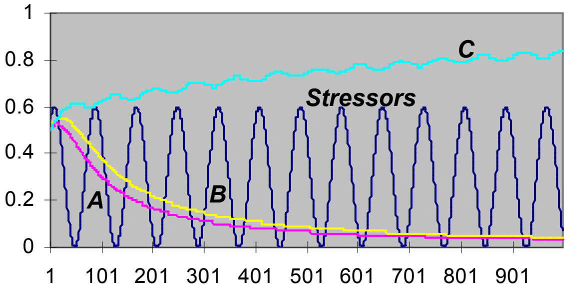


Figure 5: Cognitive Depressogenic Level for Each Individual during Repeated Stressors

Figure 5 illustrates the effects of repeated stressors condition towards different individuals. Note that the individual *C* develops a gradual increasing level of cognitive thought, while both individuals *A* and *B* show a contrast effect. Using a similar experimental setting, by using t_{max}=5000, the end of the experimental results show individual *C* will have a persistent cognitive depressogenic value equal to 1.

Mathematical Analysis

In this section, the equillibria properties are analyzed using a mathematical formal analysis. The equillibria explains condition where the values for the variables which no change occur. This condition can be represented as having any differences in temporal function between time point t and t+Δt are equal to zero (in particular for both temporal relations in *Hps* and *CdT*). To obtain possible equilibria values for the other variables, first the model is described in a differential equation form. In addition, to achieve these equilibria, all external conditions are assumed constant.

i) $dCdT(t)/dt=(1-CdT).\varphi.(Hps-\tau.CdT).CdT$
ii) $dHps(t)/dt = (1-Hps).\psi.(MdI-\phi.Hps).Hps$

Next, the equations are indentified describing

i) $dCdT(t)/dt= 0$
ii) $dHps(t)/dt = 0$

Therefore, these are equivalent to;

i) $CdT=1$ or $Hps=\tau.CdT$ or $CdT=0$
ii) $Hps =1$ or $MdI=\phi.Hps$ or $Hps=0$

From here, a first of conclusions can be derived where the equilibrium can only occur when the cognitive depressogenic thought level is equal to 1, hopelessness equals the cognitive depressogenic thought (if τ=1), or no cognitive depressogenic thought takes place. By combining these three conditions, it can be re-written into a set of relationship in $(A \vee B \vee C) \wedge (D \vee E \vee F)$ expression:

$(Hps =1 \vee MdI=\phi.Hps \vee Hps=0) \wedge (CdT=1 \vee Hps=\tau.CdT \vee CdT=0)$

From this, this expression can be elaborated using the *law of distributivity* as $(A \wedge D) \vee (A \wedge E) \vee,\ldots, \vee (C \wedge F)$. This later provides possible combinations equillibria points to be further analyzed.

Condition # 1: *CdT*=1

From this case, it can be further derived that respective values for the equilibrium condition to take place. These values can be calculated from the following formulae.

$CDi = \alpha.MiF + (1-\alpha).(SiC.NEvt.MiF)$
$PtS = \eta.AiF + (1-\eta). (1-(SiC. NEvt)).AiF$
$SrR = \beta.[\alpha.MiF + (1-\alpha).(SiC.NEvt.MiF)] + (1-\beta)$
$MdI = \gamma.[\beta.(\alpha.MiF + (1-\alpha).(SiC.NEvt.MiF)) + (1-\beta).(1-((\eta.AiF + (1-\eta).(1-(SiC.NEvt).AiF)))]$
$Im = \Upsilon. Im_{norm}$

This equillibria describes the condition when individuals are experiencing an intense negative cognitive thought throughout time will eventually have their level immunity reduced to the lowest boundary of individuals' limit. This condition creates higher vulnerability towards the development of onset during the present of negative events. It also represents the conditions where individuals with high maladaptive inferential feedbacks and situational cues levels over prolong period tend to develop cognitive depressogenic thought. Simulation traces in Case #1 and #3 confirm this equilibrium condition.

Condition # 2: *CdT*=0

Another special case of an equilibrium condition is when the *CdT* is zero. In this case, the following values are found:

$CDi = \alpha.MiF$
$PtS=\eta.AiF$
$SrR = \beta.(\alpha.MiF)$
$MdI = \gamma.\beta.(\alpha.MiF).(1-\eta.AiF)$
$Im = \Upsilon. Im_{norm}+ (1-\Upsilon) Im_{norm}$

From this, it is an equilibrium, which would be considered as a good condition since the stable individuals' immunity describes people with a good mental condition (less vulnerable towards stressors). Having this, it shows that individuals with high adaptive inferential feedbacks and low situational cues tend to have a low cognitive depressogenic thought level even during prolonged exposure towards stressors. All simulation traces from experiments (case #1, #2, and #3) confirm this condition. This condition is imperative to reduce the formation of potential relapse / recurrence caused by negative events.

Condition # 3: *Hps*=*τ.CdT*

In this condition (if τ=1), the following values are found:

$CDi = \alpha.MiF + (1-\alpha). (SiC.Hps.NEvt). MiF$
$PtS=\eta.AiF + (1-\eta). (1-(SiC.NEvt.HpS)).AiF$
$SrR = \beta.(\alpha.MiF + (1-\alpha). (SiC.Hps.NEvt). MiF) + (1-\beta).HpS$
$MdI = \gamma.[\beta.(\alpha.MiF + (1-\alpha). (SiC.Hps.NEvt). MiF) + (1-\beta).HpS.(1-(\eta.AiF + (1-\eta)).(1-(SiC.NEvt.HpS)).AiF)]$
$Im = \Upsilon. Im_{norm}+(1-\Upsilon).(1- HpS). Im_{norm}$

This equilibrium condition represents where the individuals remain constant in a cognitive depressogenic thought state over time points. If $Hps >\tau..CdT$, this condition illustrates the individuals are progressing to have a positive cognitive thought and vice versa.

Conclusion

In this paper, a model to investigate the phenomenon of the cognitive depressogenic thought has been developed. The proposed model is designed from several scientific findings in cognitive depressogenic thought and hopelessness. It provides a useful insight to understand the dynamics of related concepts in individual's cognitive depressogenic thought, inferential feedbacks, and negative events. To this end, the model is presented in a dynamic model, to allow possible experimental settings for a variety of different conditions. Using a visual programming language, several numbers of simulation experiments under different parameter settings have being performed. Despite of validating the model will be carried out in future, these experimental results pointed out that the model is able to produce behaviour of different types of inferential feedback, and it is bear a resemblance of several results in related literatures.

In addition, by a mathematical analysis, equillibria conditions of the model have been determined. This mathematical analysis is equally essential to reveal the occurrence of equilibrium conditions, primarily to illustrate the convergence and stable state of the model. Future work of this model will be specifically focus for potential integration with our existing relapse and recurrence model in unipolar depression. Having this model coupled, it will provide a better cognitive perspective on how cognitive depressogenic thought is related to the recurrence and relapse in depression. Furthermore, it will promote a better way to formulate support in automated monitoring and health informatics systems.

Acknowledgments

The preparation of this paper would not have been possible without the support, and endless efforts of Prof. Dr. Jan Treur and Dr. Michel Klein. The author is particularly pleasure to thank both of them for ideas, and refinement of this paper.

References

Abramson, L.Y., Alloy, L.B., Hogan, M.E., Whitehouse, W.G., Donovan, P., Rose, D.T., Panzarella, C., & Raniere, D. (1999). Cognitive vulnerability to depression: Theory and evidence. *Journal of Cognitive Psychotherapy*, 13, 5-20.

Abramson, L.Y., Metalsky, G.I., & Alloy, L.B. (1989). Hopelessness depression: a theory-based subtype of depression, *Psychol. Rev.* 96, pp. 358–372

Alloy, L. B., Abramson, L. Y., Gibb, B. E., Crossfield, A. G., Pieracci, A. M., Spasojevic, J., & Steinberg, J. A. (2004). Developmental antecedents of cognitive vulnerability to depression: Review of findings from the cognitive vulnerability to depression project. *Journal of Cognitive Psychotherapy*, 18(2), 115-133.

Alloy, L. B., Abramson, L. Y., Tashman, N. A., Berrebbi, D. S., Hogan, M. E., Whitehouse, W. G., et al. (2001). Developmental origins of cognitive vulnerability to depression: Parenting, cognitive, and inferential feedback styles of the parents of individuals at high and low cognitive risk for depression. *Cognitive Therapy and Research*, *25*(4), 397–423.

Beck, A.T. (1987). Cognitive models of depression, *Journal of Cognitive Psychotherapy* 1, pp. 5–37.

Coyne, J.C (1990). Interpersonal process in depression. In: G.I. Keitner, Editor, *Depression and families: Impact and treatment*, American Psychiatric Press, Washington, DC , pp. 31–53.

Crossfield, A. G., Alloy, L. B., Gibb, B. E., & Abramson, L. Y. (2002). The development of depressogenic cognitive styles: The role of negative childhood life events and parental inferential feedback. *Journal of Cognitive Psychotherapy*, *16*(4), 487–502.

DeFronzo R., Panzarella, C. & Butler, A.C. Attachment, support seeking, and adaptive feedback: Implications for psychological health (2001), *Cognitive and Behavioral Practice* 8, pp. 48–52

Dobkin, R.D, Panzarella, C., J. Fernandez, Alloy, L.B., & Cascardi, M. (2004). Adaptive inferential feedback, depressogenic inferences, and depressed mood: A laboratory study of the expanded hopelessness theory of depression, *Cognitive Therapy and Research*, pp. 487–509.

Heller, K. and Rook, K.S. (1997). Distinguishing the theoretical functions of social ties: Implications for support interventions. In: S. Duck, Editor, *Handbook of personal relationships: Theory, research, and interventions*, John Wiley and Sons, Chichester, England, pp. 649–670.

Panzarella, C., & Alloy, L. B. (1995). Social support, hopelessness, and depression: An expanded hopelessness model. In L.B. Alloy (Chair), *The hopelessness theory of depression: New developments in theory and research.* 29th Annual Convention of the Association for Advancement of Behavior Therapy, Washington, DC.

Panzarella, C., Alloy, L.B., & Whitehouse, W.G. (2006). Expanded hopelessness theory of depression: On the mechanisms by which social support protects against depression, *Cognitive Therapy and Research* 30 (3), pp. 307–33

Roberts, J.E, & Gotlib I.H (1997), Social support and personality in depression: Implications from quantitative genetics. In: R. Pierce, B. Lakey, I.G. Sarason and B.R. Sarason, Editors, *Sourcebook of social support and personality*, Plenum Press, New York, pp. 187–214.

Robinson, M. S., & Alloy, L. B. (2003). Negative cognitive styles and stress-reactive rumination interact to predict depression: A prospective study. Cognitive Therapy & Research, 27(3), 275-292.

Segal, V.Z., Pearson, L.J., and Thase, M.E (2003). Challenges in Preventing Relapse in Major Depression: Report of a National Institute of Mental Health Workshop on State of the Science of Relapse Prevention in Major Depression. *Journal of Affective Disorders* 77, 97-108.

Spasojevic, J., & Alloy, L. B. (2001). Rumination as a common mechanism relating depressive risk factors to depression. *Emotion*, 1(1), 25-37.

Distinguishing Between Intentional and Unintentional Sequences of Actions

Elisheva Bonchek Dokow (le7.bonchek.dokow@gmail.com)
Gonda Multidisciplinary Brain Research Center, Bar-Ilan University

Gal A. Kaminka (galk@cs.biu.ac.il)
Department of Computer Science and Gonda Multidisciplinary Brain Research Center, Bar-Ilan University

Carmel Domshlak (dcarmel@ie.technion.ac.il)
Faculty of Industrial Engineering and Management, Technion

Abstract

Human beings, from the very young age of 18 months, have been shown to be able to extrapolate intentions from actions. That is, upon viewing another human executing a series of actions, the observer can guess the underlying intention, even before the goal has been achieved, and even when the performer failed at achieving the goal. We identify an important preliminary stage in this process, that of determining whether or not an action stream exhibits any intentionality at all. We propose a model of this ability and evaluate it in several experiments.

Keywords: Intention; Cognitive Modeling.

Introduction

The topic of imitation has been the focus of much research in cognitive science and psychology (Meltzoff & Decety, 2003), neurophysiology (Rizzolatti, Fogassi, & Gallese, 2001), and artificial intelligence. Understanding the mechanisms underlying imitation and the time-line of their development is a part of understanding *Theory of Mind* and other aspects of social cognition. The AI community tries to model and implement this ability in software agents and robots, for the purpose of producing socially intelligent systems that can interact more meaningfully and usefully with humans.

Many different types of imitation exist, from the lower levels of gestural, facial and vocal mimicking to the higher level of goal imitation. The latter—the ability to understand the intention underlying a stream of actions, and reproduce the intended goal—is the type that we focus on here. How exactly this process takes place is yet an open question, and different researchers have addressed different aspects of it.

One of the more intriguing studies done in this area is by Meltzoff (1995), who has shown that 18-month old children are able to imitate the goal of an acting adult, *even when all they see is a series of failed attempts*. However, children are not able to do this when they observe arbitrary, intention-less, motions. These results, according to Meltzoff, assert the presence of some form of Theory of Mind at this young age.

Artificial systems have yet to reach a performance level comparable to that reported by Meltzoff. Much of the work on modeling this ability has focused on identifying the goal itself. Rao, Shon, and Meltzoff (2007) lay forth a Bayesian model for imitating goals that have been realized, and state that they intend to develop it in order to handle unrealized goals as well. Hongeng and Wyatt (2008) parse visual input and attempt to infer the goal before it is completed based on visual cues such as color and shape. However, when dealing with intentions that have not been realized—i.e., when the acting agent failed at achieving its goal—the problem becomes much more challenging. Since the observed end-state in this case is not necessarily a goal, the observing agent must first determine whether or not there is anything worth imitating here, that is, if the actions were performed with a goal in mind, and only then can it proceed to attempt to infer what exactly that goal was.

Indeed, the open challenge we tackle in this work is that of identifying whether or not an action stream has any underlying intention at all. In Meltzoff's setup (described in more detail later), the behavior of the control groups has shown that when action streams did not have any underlying intention, the observing children did not attempt to imitate the acting adult. This is crucial, since before the observing agent embarks on the intimidating task of guessing what the goal actually is, it would be wise to first decide whether there is any goal to look for.

In this paper we model this ability of discerning intentional action from unintentional action. The key idea underlying our work is the principle of rational action, which states that an agent that has a goal will take actions to achieve this goal. Inspired by this principle, we determine the intentionality of observed sequences of actions by looking at whether they are *efficient*, i.e., they monotonically move the agent further away—in problem state space—from its initial state.

We evaluate the model in two very different environments. First, we reproduce two of Meltzoff's experiments in a discrete version, using STRIPS notation[1], and show that our method results are compatible with his. Second, we report on experiments in which our method results were contrasted with adult human judgment of surveillance videos. While we only have preliminary results in this environment, they are very promising and show that our method tends to evaluate motions similarly to humans.

Background and Related Work

There is a vast amount of literature on the general topic of imitation and on, specifically, goal imitation. We cannot hope to cover it all here. We note that throughout the paper, we use the terms ”goal” and ”intention” colloquially, while a

[1] Formal language for describing states and actions in AI planning (Fikes & Nilsson, 1971).

clear distinction is sometimes made between them in previous work, e.g., (Tomasello, Carpenter, T. Behne, & Moll, 2005).

From the computational research, we refer here only to two of the more recent ones on goal inference. Meltzoff himself took a first step in this direction (Rao et al., 2007), by modeling the task in a Bayesian framework. They trained their model on several example trajectories leading to different goals, so that when given a test scenario the model could determine the goal, before it was reached. Hongeng and Wyatt (2008) analyze real-world video input, and use learning algorithms to determine higher-level goals from low level movement. Both these works build on past experience—multiple exposures to a limited set of possible goals, and learning actions that are associated with them. They also both assume intentionality, and therefore go directly to the task of inferring what that intentionality is. Thus our work on recognizing intentionality complements theirs.

Harui, Oka, and Yamada (2005) attempt to determine whether intentionality is present at all. However, their results are based mainly on vocal cues, such as ”oops”, to signal an accidental action as opposed to an intentional one. We ignore such features, since in Meltzoff (1995)’s paradigm they were neutralized. No one else, to the best of our knowledge, has attempted to computationally identify intentionality in action.

There are several psychological theories regarding the stance taken when dealing with intentionality. Meltzoff (2002) takes the mentalistic stance that infants’ ability to interpret intentionality makes use of an existing theory of mind—reasoning about the intents, desires and beliefs of others. Gergely and Csibra (2003), on the other hand, take a teleological stance, that infants apply a non-mentalistic, reality-based action interpretation system to explain and predict goal-directed actions. As Gergely and Csibra say themselves, this teleological evaluation should provide the same results as the application of the mentalistic stance as long as the actor’s actions are driven by true beliefs, as is our case.

The principle of rational action (Gergely & Csibra, 2003; Watson, 2005) plays a major role in intentional action. It states that intentional action functions to bring about future goal states by the most rational actions available to the actor within the constraints of the situation. In other words, intentional action is necessarily efficient and as such, proceeds monotonically away from the initial state.

A Method of Intentionality Recognition

We first describe briefly Meltzoff’s 1995 experiments. We then present our technique for determining intentionality.

Motivation

In order to understand the motivation for our model, as well as the setup used to evaluate it, we briefly describe some details of Meltzoff’s experiment. The purpose of his experiment was to test whether children of 18-months of age are able to understand the underlying intention of a sequence of actions, even when that intention was not realized (the acting agent failed to achieve the goal).

For five different novel toy objects, a target action was chosen. For example, for a two-piece dumbbell-shaped toy, the target action was pulling it apart. For a loop and prong device, the target action was to fit the loop onto the prong. The children were divided into four groups—”Demonstration Target”, ”Demonstration Intention”, ”Control Baseline” and ”Control Manipulation”. The children in the ”Demonstration Target” group were shown three repetitions of a successfully completed act, such as pulling apart the dumbbell, or hanging the loop on the prong; their voluntary response was to reproduce the same act when the objects were handed to them. The children in the ”Demonstration Intention” group were shown three *failed attempts* by the adult to produce the goal, where the adult (seemingly) failed at reaching it. These children’s re-enactment of the goal reached a level comparable to that of the children who saw the successful attempts. This shows that children can see through the actions to the underlying intention, and extrapolate the goal from the actions. The children in the ”Control Manipulation” group saw the object manipulated three times in ways that were not an attempt to reach the chosen target act. This was done in order to make sure that mere manipulation of the object is not enough for the children to reproduce the goal. The last control group—”Control Baseline”—had the children just see the object, without it being manipulated at all. Both control groups did not show significant success at reproducing the target act.

Meltzoff’s experiment shows that when children discern an underlying intention, as in the two Demonstration groups, they attempt to imitate it. When they do not detect such an intention, as in the Control groups, they do nothing, or sometimes mimicked the arbitrary acts of the adult (in the ”Control Manipulation” group; obviously, children were imitating *what they understood to be* the intention of the adult).

Thus a model of goal imitation must first be able to model the ability to discern whether there is an underlying intention. Only then is it relevant to attempt to discern what that intention is. This would explain why children in both ”Demonstration” groups were motivated to look for an underlying intention, while children in the ”Control Baseline” group were not. This also explains why children in the ”Control Manipulation” group sometimes reproduced the actions of the adult, even when it was not exactly what the experimenter had in mind. As long as the trace exhibited some ”rationality of action”, or efficiency, the children concluded that there was an intention worth imitating.

Recognizing Intentionality

We denote the observation trace by $t = s_0,...,s_k$, i.e. a sequence of states, brought about by the actions of the demonstrating agent. s_0 is the initial state, and s_k is the terminal state. The task of the observing agent is to decide, given this trace, whether there was an underlying intention or whether the acting agent behaved unintentionally.

Inspired by the principle of rational action, we check for some form of efficiency in the trace. It is reasonable to expect that a trace with an underlying intention will exhibit a clear progression from the initial state towards the goal state, which is the most efficient way to bring about that goal, starting from the initial state. Note that we do not know at this stage whether or not there is an underlying goal to the trace, and even if there is, if it is reached successfully. On the other hand, unintentional traces would not be driven by such efficiency, and would fluctuate towards and away from the initial state, without any clear directionality.

To do this, we define a distance measure *dist*. This distance measure is dependent on the nature of the world being modeled. For example, when dealing with geographical targets, the distance could simply be the Euclidean (and indeed it is, in one of our experiments). In a discrete state-space, defined by STRIPS notation, we use Bonet and Geffner (1999)'s Heuristic Search Planner to generate optimal plans from the initial state to every state in the trace, and the number of action steps in each generated plan is taken to be the distance to the respective state. If the demonstrating agent acts efficiently—taking only optimal action steps that bring it closer to the goal—then the distance will keep increasing. While if it acts randomly, executing various actions that do not necessarily lead anywhere, the distances will fluctuate.

There are a few requirements for the distance measure. We do not require this distance to obey symmetry ($d(s_1,s_2) = d(s_2,s_1)$). However, this distance should always be positive and equal 0 only from a state to itself. Using any such distance measure, we capture the notion of optimality, in the sense of a shortest path from one state to another.

Thus from the original state trace we induce a sequence of distance measurements $d_1 = dist(s_1,s_0),\ldots,d_k = dist(s_k,s_0)$, measuring the *optimal (minimal) distance* between each state in the sequence, and the initial state. Thus, for every state, we have an indication of how much the demonstrating agent would have had to invest (in time, number of elemental actions, or any other resource, depending on how the distance is defined), had it been intending to reach that state. We argue that enough information is preserved in this sequence for our observing agent to come to a satisfying decision.

We want to calculate from this sequence a measure of intentionality, which we take to be the proportion of local increases in the sequence—at how many of the states along the trace has the distance from the initial state increased as compared to the previous state, out of the total number of states in the trace. This will give us an idea of how efficient the action sequence is. More formally,

$$u = |\{d_i > d_{i-1}\}_{i=1}^{k}| \qquad (1)$$

is the number of states in the trace where the distance from the initial state increases, as compared to the distance at the previous state. Taking this number and dividing it by the total number of states in the trace,

$$p = \frac{u}{|\{d_k\}_{i=1}^{k}|} \qquad (2)$$

gives us a measure of intentionality for the action sequence.

The higher the resulting p, the more intentionality is attributed to the action. If a binary answer is preferred, we can determine a cutoff level above which we conclude intentionality is present, and below which we conclude it is not.

For example, in the case of clear intentionality, we would expect a strictly monotonically increasing sequence of distances; the agent proceeds from the initial state, at each step moving farther and farther away from it, and closer and closer to the intended goal. At the other end, if the observed agent is not driven by an intention to reach any particular state, we would expect the sequence to fluctuate in a seemingly random fashion, with the agent sometimes moving away from the initial state and sometimes moving back towards it. Of course, this is merely a motivational argument. In the next section we show that this simple intuitive method does indeed produce the expected results.

Implementation and Evaluation

In order to evaluate the success of our proposed measure of intentionality, we implemented it in two different environments. The first uses a discrete abstraction of Meltzoff's experiments, modeled in standard AI planning problem description (STRIPS), and the second uses surveillance videos.

Discrete Versions of Meltzoff's Experiments

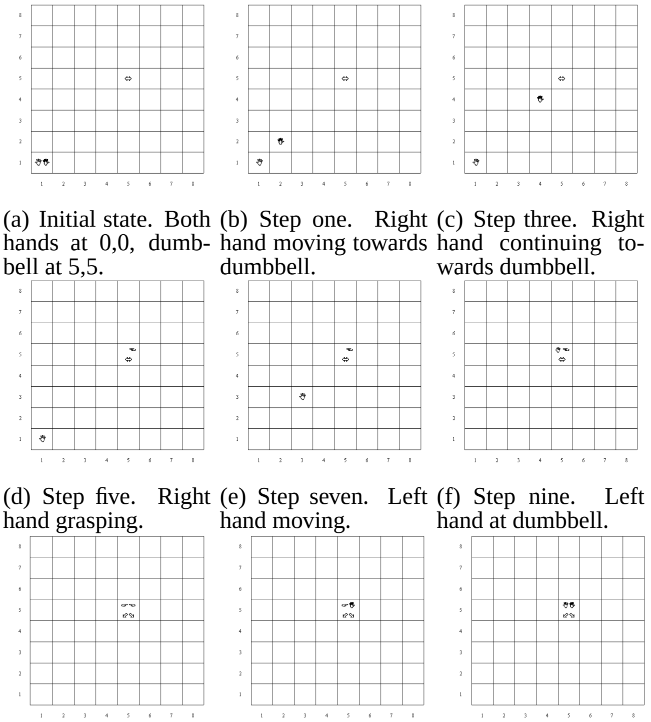
We model Meltzoff's experiment environment as an 8-by-8 grid, with several objects and several possible actions which the agent can execute with its hands, such as grasping and moving. We implemented two of the five object-manipulation experiments mentioned by Meltzoff: The dumbbell and the loop-and-prong. For the dumbbell, there is one object in the world, which consists of two separable parts. The dumbbell can be grasped by one or both hands, and can be pulled apart. For the loop-and-prong, there are two objects in the world, one stationary (the prong), and one that can be moved around (the loop). The loop can be grasped by the hand, and released on the prong or anywhere else on the grid. As previously described, we use Bonet and Geffner (1999)'s HSP to compute the distance measure.

We manually created several traces for the dumbbell and for the loop-and-prong scenarios, according to the descriptions found in Meltzoff's experiment, to fit the four different experimental groups. For example, a visual representation of the ”Demonstration Target” trace is given for the dumbbell object in Figures 1(a)–1(i).

In addition, we created a random trace, which does not exhibit any regularity. We added this trace since the children in Meltzoff's ”Control Manipulation” group were sometimes shown a sequence with underlying intention, albeit not the target one. For each trace we calculated the sequence of distances, using the above mentioned HSP algorithm, and then computed the proportion p.

Results

Figure 2 show some plots of the sequence of distances associated with the Dumbbell experiments. The step number in the sequence is measured in the X axis. The Y axis shows



(a) Initial state. Both hands at 0,0, dumbbell at 5,5.

(b) Step one. Right hand moving towards dumbbell.

(c) Step three. Right hand continuing towards dumbbell.

(d) Step five. Right hand grasping.

(e) Step seven. Left hand moving.

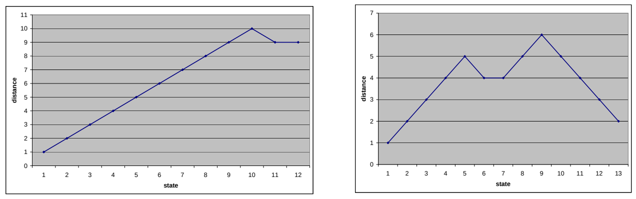
(f) Step nine. Left hand at dumbbell.

(g) Step eleven. Pulling apart.

(h) Step twelve. Releasing one hand.

(i) Step thirteen. Releasing other hand.

Figure 1: Dumbbell Demonstration Target (left to right, top to bottom).



(a) Demons. Intention I.

(b) Random.

Figure 2: Distance as a function of state in sequence in the Dumbbell experiments.

the distance. Figure 2(a) shows an almost perfectly monotonically increasing distance trace for the ”Demonstration Intent II” trace, where the right hand slips off the dumbbell, and so returns to the state it was at before it grasped it. Since only 10 out of 12 of the states showed an increase in the distance from the initial state, relative to the previous state, the intentionality score is 10/12. Figure 2(b) shows the distance sequence for the ”Random” trace. Here the graph fluctuates, demonstrating the unintentionality of the trace.

Table 2 shows the calculated measure of intentionality, for each of the traces in the prong-and-loop experiment, and Table 1 shows the same for the dumbbell experiment. In both tables, each row corresponds to a different type of state sequence. The right column shows the measure of intentionality as computed by the method described above.

In Meltzoff’s experiments, every child was shown three traces, and only then was handed the objects. There is certainly information in this seeming redundancy; see (Meltzoff, Gopnok, & Repacholi, 1999) who show that when only one trace was shown to the ”Demonstration Intention” group, the children were unable to reproduce the goal. However, we do not treat this at this stage in our model. So, while every child was shown three possibly different traces, we calculated our measure of intentionality separately for each of these traces, which is why we have more than one row in the table for some of the groups.

For example, the prong-and-loop procedure failed in two different ways in Meltzoff’s ”Demonstration Intention” experiment—either with the loop being placed too far to the right of the prong (”Demonstration Intention I” in Table 2), or too far to the left (”Demonstration Intention II”). Both these actions received an intentionality score of 1, since the end-state was reached in the most efficient possible way. In the discussion section we elaborate on the meaning of this.

The dumbbell procedure as well failed in two different ways—with the right hand ”accidentally” slipping off the dumbbell while trying to pull it apart (”Demonstration Intention I” in Table 1), or with the left hand slipping off (”Demonstration Intention II”). When the right hand slipped off it ended up slightly closer to the point where it was before the action was initiated, as opposed to where the left hand ended up when it slipped off. For this reason, the intentionality measure for ”Demonstration Intention I” is slightly lower than for ”Demonstration Intention II”.

Trace	Measure of Intentionality
Demonstration Target	1
Demonstration Intention I	0.8333
Demonstration Intention II	0.9166
Control Baseline	0
Control Manipulation	0.8333
Random	0.5384

Table 1: Calculated measure of intentionality for STRIPS implementation of the dumbbell experiment.

Trace	Measure of Intentionality
Demonstration Target	1
Demonstration Intention I	1
Demonstration Intention II	1
Control Baseline	0
Control Manipulation I	0.7777
Control Manipulation II	0.7777
Control Manipulation III	1
Random	0.5555

Table 2: Calculated measure of intentionality for STRIPS implementation of the prong-and-loop experiment.

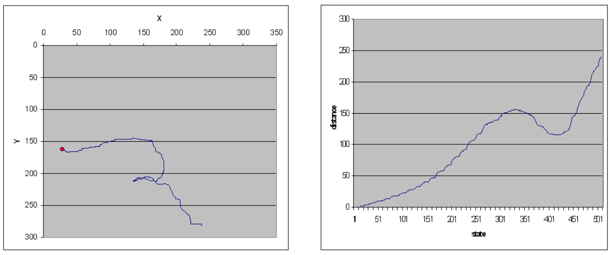
In both experimental setups, the ”Demonstration Target” trace received a clear score of 1, the highest possible intentionality. This happened because every step in the trace was necessary for bringing about the goal in the most efficient way—each and every state progressed away from the initial state and towards the goal state. The ”Control Baseline” trace received a 0, since nothing at all happened in that trace—the world remained static, at the initial state, without any change throughout the trace. The ”Random” trace received a low score, just a bit above 0.5, since the number of states progressing away from the initial state was roughly equal to the number of states returning towards it. The ”Demonstration Intention” traces exhibited a significant measure of intentionality, as did the ”Control Manipulation”. The latter can be explained by observing, as mentioned above, that even when the adults manipulated the objects in a way that was not the original intention of the experimenter, nevertheless the manipulation **did** exhibit an intentionality to reach **some** state, as opposed to just wandering about aimlessly in the space of possible states. For the dumbbell object, the arbitrary act was pushing the ends inwards (this same act was demonstrated three times). For the prong-and-loop object, the arbitrary acts were moving the loop along an imaginary line above the prong, from right to left (”Control Manipulation I”), from left to right (”Control Manipulation II”), and placing it just below the prong (”Control Manipulation III”). This last act received the ultimate intentionality score, since the end-state was reached by the most direct path.

Video Experiment

A second set of experiments was carried out in order to compare our model’s results to those of human observers. In particular, we are interested in how human observers classify real-life human movement, and whether their judgment of intentionality correlates with those of our model. To test this, we used the CAVIAR video repository of surveillance videos. We selected a dozen movies from the repository. With respect to intentionality, these range from movies that show very deliberate movements (a person crossing a lobby towards an exit), to some that are less clear (a person walking to a paper stand and browsing, then moving leisurely to a different location, etc.). We compared human subjects’ judgment of the intentionality of motions in these videos, to the predictions of our model.

Let us begin by describing how we measure intentionality using our model. The ground truth position data of the selected videos is a part of the repository, and we use it as a basis for our intentionality measurements. The planar coordinates of the filmed character in every frame in the video were taken as a state in the trace, and the distance measure we used was the Euclidean distance. As above, for every state we calculated the distance from the initial state, and then checked for how many of those states the distance increased, relative to the previous state.

Figure 3(a) shows a graph of the path of movement of the observed character, in planar coordinates, in one of the videos from the repository (video bww1_gt). Because we are plotting planar coordinates, the amount of time spent at each point is not represented here. Figure 3(b) shows a plot of the distances of each state in the path, from the initial state. The X axis measures the video frame number. The Y axis measures the distance from the initial location of the person in question. For example, the measure of intentionality for this movement path was $p = 0.48133$. Using a cutoff value of 0.5, this movement was classified as non-intentional. The interested reader is invited to watch the video and compare it to the graphs presented here.



(a) Path of movement.

(b) Distances of each state from initial state.

Figure 3: Examples from the bww1_gt video.

Those same videos were shown to human subjects who were asked to write down their opinion regarding the intentionality of the viewed character. They were given the option of segmenting the video if they thought the character changed its intention along the trace. Here we faced some difficulty in the experiment design. In pilot experiments, it became clear that asking the subjects to directly rank the “strength of intentionality” of a video segment leads to meaningless results. For instance, some subjects in pilot experiments chose to give high intentionality marks to a video segment showing a person seemingly walking around aimlessly. When we asked for an explanation, the answer was that the person in the video clearly intended to pass the time.

We thus needed to measure intentionality indirectly. To do this, subjects were requested to write down a sentence describing the intent of the person in the video, typically beginning with the words “The person intends to ... ”. The idea behind this is that in segments where there is clear intentionality, a clear answer would emerge (for instance, “The person intends to exit the room”); in other video segments, the unclear intentionality would result in more highly varied answers (e.g., some would write “intends to pass the time”, while others would write “intends to walk”, etc.). This divergence can be measured by various means; we chose the information entropy function as it is used in statistics to measure dispersion of categorical data.

Results

We unfortunately did not complete the final analysis of the results. However, preliminary results seem to indicate that our model’s classification of the movement as intentional correlates with the results obtained from the human subjects. In particular, in videos showing clear goals the human subjects tend to agree on the way the intention is described. In videos that are less clear, there is indeed divergence of the answers. Moreover, the divergence is also temporal: In movies where the goal is unclear, subjects disagreed not only on the description, but also on the internal segmentation of the video clip into segments of changing intentions. Some subjects cut the movie into several segments, while others did not. They also did not agree on the timing of the segments. Such disagreement was not noticed in the clearer movie clips.

Discussion

This work measures intentionality using a very basic feature of the stream of action. We ignore other aspects of the dynamics of the movement that certainly contain information regarding intentionality. Moreover, we assume a state-space of sufficient resolution and detail. We find justification for this in the psychological literature. Blakemore and Decety (2001) quote several works on how static images convey dynamics. Meltzoff (2007) himself uses such a discretization in yet another variant of his original experiment. In this version, instead of showing the children the full dynamics of the action, he showed them three successive static states. This technique assumes that such a representation contains enough of the information regarding the intent of the actor. In the same paper, Meltzoff also describes the failed attempt to separate the dumbbell as ”hold the dumbbell and then remove one hand quickly”, which is again a very physical description, similar to the way we modeled the experiment. Although it does not convey the notion of ”effort”, this description is yet enough to give the children a sense of intentionality.

Another point worth addressing is the high intentionality scores that some of the demonstrations received—at times the highest possible ($p = 1$), equal to that of the ”Demonstration Target” group. We stress again that we are dealing here with a preliminary stage in the process of goal imitation, that of intentionality detection. It would be wrong to conclude that a maximal score of intentionality indicates *success* at achieving the goals. Rather, we only conclude intentionality of the action and leave the question of whether the reached end-state was indeed the intended goal for a later stage.

Our model also does not deal with the fact that the demonstrations were repeated three times for every child. This information can also be used in determining intentionality (see, for example, Watson (2005) who mentions persistence as a sign of intentionality), as well as for the later stage of determining whether the reached end-state is the intended goal.

Future Work

Having only just touched the tip of the iceberg regarding the intriguing phenomena of intentionality detection and goal imitation, there is yet much work to be done. In addition to more rigorously testing and evaluating our current model, we intend to broaden it to deal with the notions of persistence and equifinality—information carried by the repetition of every demonstration three times. It would also be interesting to add the possibility of handling varying environmental constraints, such as obstacles, which affect the calculation of the distance measure, as well as treating false beliefs regarding those environmental constraints, and seeing how they affect the conclusion reached regarding intentionality.

Acknowledgments. Videos were taken from EC Funded CAVIAR project/IST 2001 37540. This research was partially supported by ISF grant #1357/07.

References

Blakemore, S. J., & Decety, J. (2001). From the perception of action to the understanding on intention. *Nature reviews, neuroscience*.

Bonet, B., & Geffner, H. (1999). Planning as heuristic search: new results. In S. Biundo & M. Fox (Eds.), *Procceedings of the 5th european conference on planning.* Durham, UK: Springer: Lecture Notes on Computer Science.

Fikes, R. E., & Nilsson, N. J. (1971). Strips: a new approach to the application of theorem proving to problem solving. *Artificial intelligence*.

Gergely, G., & Csibra, G. (2003). Teleological reasoning in infancy: the naive theory of rational action. *TRENDS in cognitive science, 7*(7).

Harui, K., Oka, N., & Yamada, Y. (2005). Distinguishing intentional actions from accidental actions. In *Proceedings of the 4th IEEE international conference on development and learning.*

Hongeng, S., & Wyatt, J. (2008). Learning causality and intentional actions. In E. Rome, J. Hertzberg, & G. Dorffner (Eds.), *Towards affordance-based robot control.* Springer.

Meltzoff, A. N. (1995). Understanding the intentions of others: re-enactment of intended acts by 18-month-old children. *Developmental psychology, 31*(5).

Meltzoff, A. N. (2002). Imitation as a mechanism of social cognition: origins of empathy, theory of mind, and representation of action. In U. Goswami (Ed.), *Blackwells handbook of childhood cognitive development.* Blackwell.

Meltzoff, A. N. (2007). The ”like-me” framework for recognizing and becoming an intentional agent. *Acta psychologica*.

Meltzoff, A. N., & Decety, J. (2003). What imitation tells us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience. *Philosophicla transactions of the royal society of London*.

Meltzoff, A. N., Gopnok, A., & Repacholi, B. M. (1999). Toddlers' understanding of intentions, desires and emotions: exploration of the dark ages. In P. D. Zelazo, J. W. Astington, & D. R. Olson (Eds.), *Developing theories on intention: social understanding and self control.* Lawrence Erlbaum Associates.

Rao, R. P. N., Shon, A. P., & Meltzoff, A. N. (2007). A bayesian model of imitation in infants and robots. In C. L. Nehaniv & K. Dautenhahn (Eds.), *Imitation and social learning in robots, humans, and animals: behavioural, social and communicative dimensions.* Cambridge University Press.

Rizzolatti, G., Fogassi, L., & Gallese, V. (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature reviews in neuroscience*.

Tomasello, M., Carpenter, M., T. Behne, J. C. abd, & Moll, H. (2005). Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences*.

Watson, J. S. (2005). The elementary nature of purposive behavior: evolving minimal neural structures that display intrinsic intentionality. *Evolutionary psychology*.

A Neural Model for Adaptive Emotion Reading Based on Mirror Neurons and Hebbian Learning

Tibor Bosse, Zulfiqar A. Memon, Jan Treur
(http://www.few.vu.nl/~{tbosse, zamemon, treur})
Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, the Netherlands

Abstract

This paper addresses the use of Hebbian learning principles to model in an adaptive manner capabilities to interpret somebody else's emotions. First a non-adaptive neural model for emotion reading is described involving (preparatory) mirror neurons and a recursive body loop: a converging positive feedback loop based on reciprocal causation between mirror neuron activations and neuron activations underlying emotions felt. Thus emotion reading is modelled taking into account the Simulation Theory perspective as known from the literature, involving the own emotions in reading somebody else's emotions. Next the neural model is extended to an adaptive neural model based on Hebbian learning within which a direct connection between a sensed stimulus concerning another person's body state (e.g., face expression) and the emotion recognition state is strengthened.

Introduction

In the Simulation Theory perspective on emotion reading (or Theory of Mind) it is assumed that a person uses the facilities involving the own mental states that are counterparts of the mental states attributed to another person; e.g., (Goldman, 2006). For example, the state of feeling pain oneself is used in the process to determine whether the other person has pain. More and more neurological evidence supports this perspective, in particular the recent discovery of mirror neurons that are activated both when preparing for an action (including a change in body state) and when observing somebody else performing a similar action.; e.g., (Rizzolatti, Fogassi, and Gallese, 2001; Wohlschlager and Bekkering, 2002; Kohler, Keysers, Umilta, Fogassi, Gallese, and Rizzolatti, 2002; Ferrari, Gallese, Rizzolatti, and Fogassi, 2003; Rizzolatti, 2004; Rizzolatti and Craighero, 2004; Iacoboni, 2008).

Mirror neurons usually concern neurons involved in the preparation of actions or body states. By Damasio (1999) such preparation neurons are attributed a crucial role in generating and feeling emotional responses. In particular, using a 'body loop' or 'as if body loop', a connection between such neurons and the feeling of emotions by sensing the own body state is obtained; see (Damasio, 1999) or the formalisation presented in (Bosse, Jonker and Treur, 2008). Taken together, the existence of mirror neurons and Damasio's theory on feeling emotions based on (as if) body loops provides strong neurological support for the Simulation Theory perspective on emotion reading.

An extension of this idea was adopted by assuming that the (as if) body loop is processed in a recursive manner: a positive feedback loop based on reciprocal causation between feeling state (with gradually more feeling) and body state (with gradually stronger expression). This cycle is triggered by the stimulus and ends up in an equilibrium for both states. In (Bosse, Memon, and Treur, 2008; Memon and Treur, 2008) it was shown how a cognitive emotion reading model based on a recursive body loop can be obtained based on causal modelling using the hybrid modelling language LEADSTO (Bosse, Jonker, Meij and Treur, 2007). In (Bosse, Memon, and Treur, 2009) it was shown how this hybrid causal model can be extended to obtain an adaptive cognitive emotion reading model. The adaptation creates a shortcut connection from the sensed stimulus (observed facial expression) to the imputed emotion, bypassing the own emotional states.

In the current paper a different model is presented for similar mind reading phenomena. This time, instead of a causal modelling approach, a more neurological point of departure is chosen by using a neural network structure which is processed in a purely numerical manner using generic principles for neural activation and Hebbian learning. In this way the obtained model stays more close to the neurological source of evidence and inspiration.

The structure of this paper is as follows. First, the basic neural emotion reading model is introduced. Next, it is shown how the model can be made adaptive, by adopting a Hebbian learning principle that enables the model to strengthen the connections between neurons. For both the basic model and the adaptive model, some simulation results are shown, and different variations are discussed. The paper is concluded with a discussion.

A Neural Emotion Reading Model

In this and the next section the model to generate emotional states for a given stimulus is introduced. It adopts three important concepts from Damasio (1999)'s theory of consciousness: an *emotion* is defined as 'an (unconscious) neural reaction to a certain stimulus, realised by a complex ensemble of neural activations in the brain', a *feeling* is 'the (still unconscious) sensing of this body state', and a *conscious feeling* is what emerges when 'the organism detects that its representation of its own body state has been changed by the occurrence of the stimulus' (Damasio, 1999). Moreover, the model adopts his idea of a 'body loop' and 'as if body loop', but extends this by making these loops recursive. According to the original idea, from a neurological perspective emotion generation roughly proceeds according to the following causal chain; see (Bosse, Jonker and Treur, 2008; Damasio, 1999) (in the case of a body loop):

sensing a stimulus →
sensory representation of stimulus →
(preparation for) bodily response →
sensing the bodily response →
sensory representation of the bodily response →
feeling the emotion

As a variation, an 'as if body loop' uses a causal relation

preparation for bodily response →
sensory representation of the bodily response

as a shortcut in the neurological chain. In the model used here an essential addition is that the body loop (or as if body loop) is extended to a recursive body loop (or recursive as if body loop) by assuming that the preparation of the bodily response is also affected by the state of feeling the emotion (also called emotional feeling):

feeling the emotion → preparation for bodily response

as an additional causal relation. Damasio (2004) also assumes such recursively used reciprocal causal connections:

> '… feelings are not a passive perception or a flash in time, especially not in the case of feelings of joy and sorrow. For a while after an occasion of such feelings begins – for seconds or for minutes – there is a dynamic engagement of the body, almost certainly in a repeated fashion, and a subsequent dynamic variation of the perception. We perceive a series of transitions. We sense an interplay, a give and take.' (Damasio, 2004, p. 92)

Within the neural model presented here both the neural states for preparation of bodily response and the feeling are assigned a level of activation, expressed by a number, which is assumed dynamic. The cycle is modelled as a positive feedback loop, triggered by the stimulus and converging to a certain level of feeling and body state. Here in each round of the cycle the next body state has a level that is affected by both the level of the stimulus and of the emotional feeling state, and the next level of the emotional feeling is based on the level of the body state.

This neural model refers to activation states of (groups of) neurons and the body. An overall picture of the connection for this model is shown in Figure 1. Here each node stands for a group of one or more neurons, or for an effector, sensor or body state. The nodes can be interpreted as shown in Table 1.

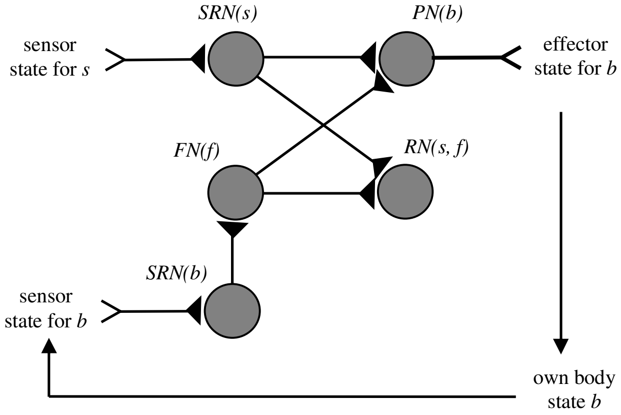


Figure 1: Neural network structure of the model with body loop

In the neural activation state of *RN(s, f)*, the experienced emotion *f* is related to the stimulus *s*, which triggers the emotion generation process. Note that the more this neuron is strongly related to *SRN(s)*, the more it may be considered to represent a level of awareness of what causes the feeling *f*; this may be related to what by Damasio (1999) is called a state of conscious feeling. This state that relates an emotion felt *f* to any triggering stimulus *s* can play an important role in the conscious attribution of the feeling to any stimulus *s*.

node nr	*denoted by*	*description*
0	*s*	stimulus; for example, another person's body state b'
1	*SS(s)*	sensor state for stimulus *s*
2	*SRN(s)*	sensory representation neuron for *s*
3	*PN(b)*	preparation neuron for own body state *b*
4	*ES(b)*	effector state for own body state *b*
5	*BS(b)*	own body state *b*
6	*SS(b)*	sensor state for own body state *b*
7	*SRN(b)*	sensory representation neuron for own body state *b*
8	*FN(f)*	neuron for feeling state *f*
9	*RN(s, f)*	neuron representing that *s* induces feeling *f*

Table 1 Overview of the nodes involved

According to the Simulation Theory perspective a neural model for emotion reading should essentially be based on a neural model to generate the own emotions as induced by any stimulus *s*. Indeed, the neural model introduced above can be specialised in a quite straightforward manner to enable emotion reading. The main step is that the stimulus *s* that triggers the emotional process, which until now was left open, is instantiated with the body state *b'* of another person, for example a facial expression of another person. Indeed, more and more evidence is available that (already from an age of 1 hour), as an example of the functioning of the mirror neuron system (Rizzolatti, 2005), sensing somebody else's facial expression leads (within about 300 milliseconds) to preparing for and showing the same facial expression

(Goldman and Sripada, 2004, pp. 129-130). Within the network in Figure 1 this leads (via activation of the sensory representation state *SRN(b')*) to activation of the preparation state *PN(b)* where *b* is the own body state corresponding to the other person's body state *b'*. This pattern shows how this preparation state *PN(b)* functions as a mirror neuron. Next, via the recursive body loop gradually higher and higher activation levels of the own feeling state *f* are generated.

To formally specify the neural model, the mathematical concepts listed in Table 2 are used.

concept	*description*
N	set of node numbers (as listed in Table 1); variables indicating elements of this set are *i, j, k*
N'	*N*\{*0*} the set of node numbers except the node for the stimulus *s*
$w_{ij}(t)$	strength of the connection from node *i* to node *j* at time *t*; this is taken *0* when no connection exists or when *i=j*
$y_i(t)$	activation level of node *i* at time *t*
$net_i(t)$	net input to node *i* at time *t*
g	function to determine activation level from net input
γ	change rate for activation level
η	learning rate for weights

Table 2 Mathematical concepts used

The function *g* can take different forms, varying from the identity function $g(v) = v$ for the linear case, to a discontinuous threshold (indicated by β) step function with $g(v) = 0$ for $v<\beta$ and $g(v) = 1$ for $v\geq\beta$, or a continuous logistic threshold function based on $1/(1+exp(-\alpha(v-\beta))$ with steepness α. For the connections between nodes of which at least one is not a neuron the connections have been made simple: weights *1* and *g* the identity function; so $w_{12} = w_{34} = w_{45} = w_{56} = w_{67} = 1$

The activation levels are determined for step size Δt for all $i \in N'$ as follows:

$$net_i(t) = \Sigma_{j\in N}\ w_{ji}(t)\, y_j(t)$$
$$\Delta y_i(t) = \gamma(g(net_i(t)) - y_i(t))\, \Delta t$$

Note that for step size $\Delta t = 1$ and change rate $\gamma = 1$, the latter difference equation can be rewritten to

$$y_i(t+1) = g(net_i(t))$$

which is a wellknown formula in the literature addressing simulation with neural models.

The model description in the form of a system of differential equations can be used for an analysis of equilibria that can occur. Here the external stimulus level for *s* is assumed constant. Moreover, it is assumed that $\gamma > 0$. In general putting $\Delta y_i(t) = 0$ provides the following set of equations for $i \in N'$:

$$y_i = g(\Sigma_{j\in N}\ w_{ji}\, y_j)$$

For the given network structure these equilibrium equations are:

$$y_1 = g(w_{01}\, y_0)$$
$$y_2 = g(w_{12}\, y_1)$$
$$y_4 = g(w_{34}\, y_3)$$
$$y_5 = g(w_{45}\, y_4)$$
$$y_6 = g(w_{56}\, y_5)$$
$$y_7 = g(w_{67}\, y_6)$$
$$y_8 = g(w_{78}\, y_7)$$
$$y_3 = g(w_{23}\, y_2 + w_{83}\, y_8)$$
$$y_9 = g(w_{29}\, y_2 + w_{89}\, y_8)$$

Taking into account that connections between nodes among which at least one is not a neuron have weight *1* and *g* the identity function, it follows that the equilibrium equations are:

$$y_2 = y_1 = y_0$$
$$y_7 = y_6 = y_5 = y_4 = y_3$$
$$y_8 = g(w_{78}\, y_7)$$
$$y_3 = g(w_{23}\, y_2 + w_{83}\, y_8)$$
$$y_9 = g(w_{29}\, y_2 + w_{89}\, y_8)$$

Example Simulations: Non-Adaptive Case

The numerical software environment Matlab has been used to obtain simulation traces for the model described above. An example simulation trace that results from this model with the function *g* the identity function is shown in Figure 2. Here, time is on the horizontal axis, and the activation levels of three of the neurons *SRN(s)*, *FN(f)*, and *RN(s,f)* are shown on the vertical axis. As shown in this picture, the sensory representation of a certain stimulus *s* quickly results in a feeling state *f*, and a representation that *s* induces *f*. When the stimulus s is not present anymore, the activations of *FN(f)* and *RN(s, f)* quickly decrease to *0*. The weight factors taken are: $w_{23} = w_{83} = w_{89} = 0.1$, $w_{78} = 0.5$ and $w_{29} = 0$. Moreover, $\gamma = 1$, and a logistic threshold function was used with threshold *0.1* and steepness *40*.

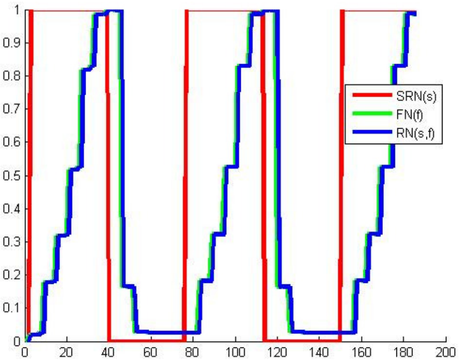


Figure 2: Example simulation for non-adaptive emotion reading

For the values taken in the simulation above, the equilibrium equations are:

$$y_2 = y_1 = y_0$$
$$y_7 = y_6 = y_5 = y_4 = y_3$$
$$y_8 = g(0.5\, y_7)$$
$$y_3 = g(0.1\, y_2 + 0.1\, y_8)$$
$$y_9 = g(0.1 y_8)$$

As the threshold was taken *0.1* it follows from the equations that for stimulus level $y_0 = 0$ all values for y_i are (almost) *0*, and for stimulus level $y_0 = 1$ that all values for y_i are *1*, which is also shown by the simulation in Figure 2.

An Adaptive Neural Emotion Reading Model

As a next step, the neural model for emotion reading is extended by a facility to strengthen the direct connection between the neuron *SRN(s)* for the sensory representation of the stimulus (the other person's face expression) and the neuron *RN(s, f)*. A strengthening of this connection over time creates a different emotion reading process that in principle can bypass the generation of the own feeling. The learning principle to achieve such an adaptation process is based on the Hebbian learning principle that connected neurons that are frequently activated simultaneously strengthen their connecting synapse e.g., (Hebb, 1949; Bi and Poo, 2001; Gerstner and Kistler, 2002; Wasserman, 1989). The change in strength for the connection w_{ij} between nodes $i, j \in N$ is determined (for step size Δt) as follows:

$$\Delta w_{ij}(t) = \eta\, y_i(t) y_j(t)(1 - w_{ij}(t))\, \Delta t$$

Here η is the learning rate. Note that this Hebbian learning rule is applied only to those pairs of nodes $i, j \in N$ for which a connection already exists.

Also for the adaptive case equilibrium equations can be found. Here it is assumed that $\gamma, \eta > 0$. In general putting both $\Delta y_i(t) = 0$ and $\Delta w_{ij}(t) = 0$ provides the following set of equations for $i, j \in N'$:

$$y_i = g(\Sigma_{j \in N}\ w_{ji}\, y_j)$$
$$y_i y_j(1 - w_{ij}) = 0$$

From the latter set of equations (second line) it immediately follows that for any pair $i, j \in N'$ it holds:

either $y_i = 0$
or $y_j = 0$
or $w_{ij} = 1$

In particular, when for an equilibrium state both y_i and y_j are nonzero, then $w_{ij} = 1$.

Example Simulations: Adaptive Case

Based on the neural model for adaptive emotion reading obtained in this way, a number of simulations have been performed; for an example, see Figure 3. As seen in this figure, the strength of the connection between *SRN(s)* and *RN(s, f)* (indicated by *b* which is in fact w_{29}) is initially *0* (i.e., initially, when observing the other person's face, the person does not impute feeling to this). However, during an adaptation phase of two trials, the connection strength goes up as soon as the person imputes feeling *f* to the target stimulus s (the observation of the other person's face), in accordance with the temporal relationship described above.

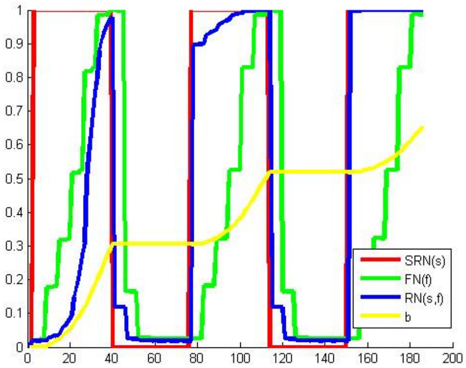


Figure 3: Example simulation for adaptive emotion reading

Note that, as in Figure 2, the activation values of other neurons gradually increase as the person observes the stimulus, following the recursive feedback loop discussed. These values sharply decrease as the person stops observing the stimulus as shown in Figure 3, e.g. from time point 40 to 76, from time point 112 to 148, and so on. Note that at these time points the strength of the connection between *SRN(s)* and *RN(s, f)* (indicated by *b*) remains stable. After the adaptation phase, and with the imputation sensitivity at high, the person imputes feeling *f* to the target stimulus directly after occurrence of the sensory representation of the stimulus, as shown in the third trial in Figure 3. Note here that even though the person has adapted to impute feeling *f* to the target directly after the stimulus, the other state property values continue to increase in the third trial as the person receives the stimulus; this is because the adaptation phase creates a connection between the sensory representation of the stimulus and emotion imputation without eliminating the recursive feedback loop altogether. Note that when a constant stimulus level *1* is taken, an equilibrium state is reached in which $b = 1$, and all y_i are *1*.

The learning rate η used in the simulation shown in Figure 3 is *0.02*. In Figure 4 a similar simulation is shown for a lower learning rate: *0.005*.

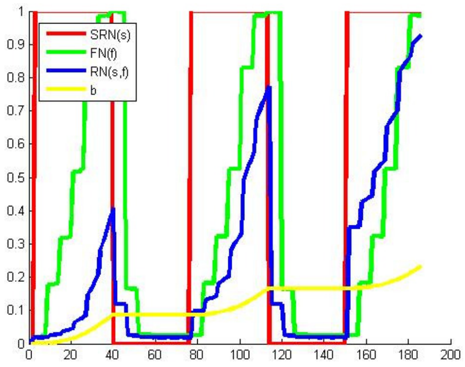


Figure 4: Adaptive emotion reading with lower learning rate

Discussion

In recent years, an increasing amount of neurological evidence is found that supports the 'Simulation Theory' perspective on emotion reading, e.g., (Rizzolatti, Fogassi, and Gallese, 2001; Wohlschlager and Bekkering, 2002; Kohler, Keysers, Umilta, Fogassi, Gallese, and Rizzolatti, 2002; Ferrari, Gallese, Rizzolatti, and Fogassi, 2003; Rizzolatti, 2004; Rizzolatti and Craighero, 2004; Iacoboni, 2005, 2008). That is, in order to recognise emotions of other persons, humans exploit observations of these other persons' body states as well as counterparts within their own body. The current paper introduces a numerical model to simulate this process. This model is based on the notions of (preparatory) mirror neurons and a recursive body loop (cf. Damasio, 1999, 2004): a converging positive feedback loop based on reciprocal causation between mirror neuron activations and neuron activations underlying emotions felt. In addition, this model was extended to an adaptive neural model based on Hebbian learning, where neurons that are frequently activated simultaneously strengthen their connecting synapse (cf. Hebb, 1949; Bi and Poo, 2001; Gerstner and Kistler, 2002; Wasserman, 1989). Based on this adaptive model, a direct connection between a sensed stimulus (for example, another person's face expression) and the emotion recognition can be strengthened.

The simulation model has been implemented in Matlab, in a generic manner. That is, the model basically consists of only 2 types of rules: one for propagation of activation levels between connected neurons, and one for strengthening of connections between neurons that are active simultaneously. These rules are then applied to all nodes in the network. To perform a particular simulation, only the initial activation levels and connection strengths have to be specified. Both for the non-adaptive and for the adaptive model, a number of simulations have been performed. These simulations indicated that the model is indeed sufficiently generic to simulate various patterns of adaptive emotion reading. An interesting question for further research is to what extent the model can simulate other neural processes as well. Another challenge for the future is to extend the model such that it can cope with multiple qualitatively different emotional stimuli (e.g., related to joy, anger, or fear), and their interaction.

Validation of the presented model is not trivial. At least, this paper has indicated that it is possible to integrate Damasio's idea of body loop with the notion of mirror neurons and Hebbian learning, and that the resulting patterns are very plausible according to the literature. In this sense the model has been validated positively. However, this is a relative validation, only with respect to the literature that forms the basis of the model. A more extensive empirical evaluation is left for future work.

By other approaches found in the literature, a specific emotion recognition process is often modelled in the form of a prespecified classification process of facial expressions in terms of a set of possible emotions; see, for example, (Cohen, Garg, and Huang, 2000; Malle, Moses, and Baldwin, 2001; Pantic and Rothkrantz, 1997, 2000). Although a model based on such a classification procedure is able to perform emotion recognition, the imputed emotions have no relationship to a person's own emotions. The neural model for emotion reading presented in the current paper uses a person's own feelings in the emotion reading process as also claimed by the Simulation Theory perspective, e.g., (Goldman, 2006; Goldman and Sripada, 2004). Besides, in the neural model presented here a direct classification is learnt by the adaptivity model based on a Hebbian learning rule. A remarkable issue here is that such a direct connection is faster (it may take place within hundreds of milliseconds) than a connection via a body loop (which usually takes seconds). This time difference implies that first the emotion is recognised without feeling the corresponding own emotion, but within seconds the corresponding own emotion is in a sense added to the recognition. When an as if body loop is used instead of a body loop, the time difference will be smaller, but still present. An interesting question is whether it is possible to design experiments that show this time difference as predicted by the neural model.

Some other computational models related to mirror neurons are available in literature; for instance: a genetic algorithm model which develops networks for imitation while yielding mirror neurons as a byproduct of the evolutionary process (Borenstein and Ruppin, 2005); the mirror neuron system (MNS) model that can learn to 'mirror' via self-observation of grasp actions (Oztop and Arbib, 2002); the mental state inference (MSI) model that builds on the forward model hypothesis of mirror neurons (Oztop, Wolpert, and Kawato, 2005), etc. A comprehensive review of these computational studies can be found in (Oztop, Kawato, and Arbib, 2006). All of the above listed computational models and many others available in the literature are targeted to imitation, whereas the neural model presented here specifically targets to interpret somebody else's emotions.

The approach adopted in the current paper has drawn some inspiration from the four models sketched (but not formalised) in (Goldman, 2006, pp. 124-132). The recursive body loop (or as if body loop) introduced here addresses the problems of model 1, as it can be viewed as an efficient and converging way of generating and testing hypotheses for the emotional states. Moreover, it solves the problems of models 2 and 3, as the causal chain from facial expression to emotional state is not a reverse simulation, but just the causal chain via the body state which is used for generating the own emotional feelings as well. Finally, compared to model 4, the models put forward here can be viewed as an efficient manner to obtain a mirroring process between the emotional state of the other person on the own emotional state, based on the machinery available for the own emotional states.

References

Bi, G.Q., and, M.M. (2001) Synaptic Modifications by Correlated Activity: Hebb's Postulate Revisited. Ann Rev Neurosci, vol. 24, pp. 139-166.

Borenstein, E., & Ruppin, E. (2005). The evolution of imitation and mirror neurons in adaptive agents. *Cognitive Systems Research,* 6(3).

Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J. (2007). A Language and Environment for Analysis of Dynamics by Simulation. *International Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464.

Bosse, T., Jonker, C.M., and Treur, J., (2008). Formalisation of Damasio's Theory of Emotion, Feeling and Core Consciousness. *Consciousness and Cognition Journal*, vol. 17, 2008, pp. 94–113. Preliminary, shorter version in: Fum, D., Del Missier, F., Stocco, A. (eds.), *Proc. of the 7th Int. Conference on Cognitive Modelling, ICCM'06*, 2006, pp. 68-73.

Bosse, T., Memon, Z.A., and Treur, J. (2008), Adaptive Estimation of Emotion Generation for an Ambient Agent Model. In: Aarts, E., et.al. (eds.), *Ambient Intelligence, Proceedings of the Second European Conference on Ambient Intelligence, AmI'08*. Lecture Notes in Computer Science, vol. 5355. Springer Verlag, 2008, pp. 141–156.

Bosse, T., Memon, Z.A., and Treur, J., (2009). An Adaptive Emotion Reading Model. In: *Proc. of the 31th Annual Conference of the Cognitive Science Society, CogSci'09*. Cognitive Science Society, Austin, TX, 2009, to appear.

Cohen, I., Garg, A., and Huang, T.S. (2000). Emotion recognition using multilevel HMM. In: *Proc. of the NIPS Works. on Affective Computing*, Colorado, 2000.

Damasio, A. (1999). *The Feeling of What Happens: Body, Emotion and the Making of Consciousness*. Harcourt Brace, 1999.

Damasio, A. (2004). *Looking for Spinoza*. Vintage books, London, 2004.

Ferrari PF, Gallese V, Rizzolatti G, Fogassi, L. 2003. Mirror neurons responding to the observation of ingestive and communicative mouth actions in the monkey ventral premotor cortex. *Eur. J. Neurosci.* 17:1703–14.

Gerstner, W., and Kistler, W.M. (2002). Mathematical formulations of Hebbian learning. Biol. Cybern., vol. 87, 2002, pp. 404-415.

Goldman, A.I. (2006). *Simulating Minds: the Philosophy, Psychology and Neuroscience of Mindreading*. Oxford University Press.

Goldman, A.I., and Sripada, C.S. (2004). Simulationist models of face-based emotion recognition. *Cognition*, vol. 94, pp. 193–213.

Hebb, D.O. (1949). *The Organization of Behaviour*. John Wiley & Sons, New York, 1949.

Iacoboni M. (2008). *Mirroring People*. New York: Farrar, Straus & Giroux

Iacoboni, M., (2005). Understanding others: imitation, language, empathy. In: Hurley, S. & Chater, N. (2005). (eds.). *Perspectives on imitation: from cognitive neuroscience to social science* vol. 1, MIT Press, pp. 77-100.

Kohler, E., Keysers, C., Umilta, M.A., Fogassi, L., Gallese, V., and Rizzolatti, G. 2002. Hearing sounds, understanding actions: action representation in mirror neurons. *Science* 297:846–48

Malle, B.F., Moses, L.J., Baldwin, D.A. (2001). *Intentions and Intentionality: Foundations of Social Cognition*. MIT Press.

Memon, Z.A., and Treur, J. (2008), Cognitive and Biological Agent Models for Emotion Reading. In: Jain, L. et al. (eds.), *Proceedings of the 8th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'08*. IEEE Computer Society Press, 2008, pp. 308-313.

Oztop, E., & Arbib, M.A. (2002). Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics,* 87(2), 116-140.

Oztop, E., Kawato, M. and Arbib M. (2006), Mirror neurons and imitation: a computationally guided review, Neural Networks 19 (3) (2006) 254-271.

Oztop, E., Wolpert, D., & Kawato, M. (2005). Mental state inference using visual control parameters. C*ognitive Brain Research,* 22(2), 129-151.

Pantic, M., and Rothkrantz, L.J.M. (1997). Automatic Recognition of Facial Expressions and Human Emotions. In: *Proceedings of ASCI'97 conference*, ASCI, Delft, pp. 196-202.

Pantic, M., and Rothkrantz, L.J.M. (2000), Expert System for Automatic Analysis of Facial Expressions, *Image and Vision Computing Journal*, vol. 18, pp. 881-905.

Rizzolatti, G. and Craighero, L. (2004) The mirror-neuron system. *Annu. Rev. Neurosci.* **27**, 169–92.

Rizzolatti G, Fogassi L, Gallese V (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. Nature Rev Neurosci 2:661–670.

Rizzolatti G. 2005. The mirror-neuron system and imitation. In: Hurley, S. & Chater, N. (2005). (eds.). *Perspectives on imitation: from cognitive neuroscience to social science*, vol. 1, MIT Press, pp. 55-76.

Wasserman, P.D. (1989). *Neural Computing: Theory and Practice*. Van Nostrand Reinhold, New York, 1989.

Wohlschlager A, Bekkering H. 2002. Is human imitation based on a mirror-neurone system? Some behavioural evidence. *Exp. Brain Res.* 143:335–41

Validation of an Agent Model for Human Work Pressure

Fiemke Both[1], Mark Hoogendoorn[1], S. Waqar Jaffry[1], Rianne van Lambalgen[1], Rogier Oorburg[2], Alexei Sharpanskykh[1], Jan Treur[1], and Michael de Vos[2]

[1]Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands
{fboth, swjaffry, rm.van.lambalgen, sharp, treur}@few.vu.nl

[2]Force Vision Lab, Barbara Strozzilaan 362a,
1083 HN Amsterdam, The Netherlands
{rogier, michael}@forcevisionlab.nl

Abstract

Human performance can seriously degrade under demanding tasks. To improve performance, agents can reason about the current state of the human, and give the most appropriate and effective support. To enable this, the agent needs a work pressure model, which should be valid, as the agent might otherwise give inappropriate advice and even worsen performance. This paper concerns the validation of an existing work pressure model. First, human experiments have been designed and conducted, whereby measurements related to the model have been performed. Next, this data has been used to obtain appropriate parameter settings for the work pressure model, describing the specific subject. Finally, the work pressure model, with the tailored parameter settings, has been used to predict human behavior to investigate predictive capabilities of the model. The results have been analyzed using formal verification.

Introduction

In demanding working circumstances the quality of the tasks performed by a human might be severely influenced (cf. Hancock *et al.*, 1995, Hanley, 1997). Especially when tasks are performed in a critical domain, such effects are highly undesired. To improve task performance in such situations, personal assistant agents (cf. Kozieok and Maes, 1993; Mitchell *et al.*, 1994; Maheswaran *et al.,* 2003) can be used to monitor the activities of the human, and intervene in case needed. Interventions could for example take the form of assigning (part of) the tasks to other humans, or give advice regarding the performance of the task.

One crucial element in the support given by a personal assistant agent is that it should be given in appropriate circumstances: the agent should have an awareness of the state of the human. In Bosse *et al.* (2008a) a dynamical model has been presented that describes the cognitive workload experienced by humans, given knowledge of the human's characteristics in combination with the tasks that need to be performed. The model is quantitative, based upon mostly qualitative theories from Psychology, but was not validated yet using human experiments. The primary focus of this paper is to develop and implement an approach for the validation of this human work pressure model. The validation has been performed by taking a number of steps. First of all, an experiment with 31 human subjects has been conducted. Hereby, the subjects were to play a game whereby they experience different amounts of workload. Each subject was given two conditions. Using the empirical data obtained from this experiment, parameter estimation techniques have been deployed to find appropriate parameter settings for the model to accurately describe the subject's behavior in one of the conditions. Thereafter, these settings have been used to predict the behavior of the subject in the other condition. Finally, properties that relate to the work pressure model have been verified against the empirical data as well.

This paper is organized as follows. First, the work pressure model is briefly explained. Thereafter, the setup of the experiment and the results of parameter estimation are shown. Next, the verification of properties against the empirical data, and finally the paper is concluded and future work is discussed.

Work pressure model

The Agent model for the Functional State (FS) of a human represents the dynamical state of a person when performing a certain task. States such as experienced pressure, motivation and exhaustion of the person are predicted, but also the performance quality and the amount of generated effort to the task..

The model is based on two different theories: 1) the cognitive energetic framework (Hockey, 1997), which states that effort regulation is based on human recourses and determines human performance in dynamic conditions; 2) The idea, that when performing sports, a person's generated power can continue on a *critical power* level without becoming more exhausted (Hill, 1993). In the FS model (cf. Figure 1) critical power is represented by the critical point: the amount of effort someone can generate without becoming more exhausted.

As input the FS model uses external factors (task demands and environment state) and personal factors (experience, cognitive abilities and personality profile), which are used to determine a person's dynamical state. In addition, it determines the relation of this state to the human's actions with respect to the task (e.g. performance quality), represented in the Task Execution State.

An example equation of the model is:

$$E(t+\Delta t) = E(t) + Pos(\eta \cdot (GE(t)-CP(t)) \cdot \Delta t) - \pi \cdot RE(t) \cdot \Delta t$$

Here Exhaustion (*E*) builds up or reduces over time. When the generated effort (*GE*) is above the critical point (*CP*), exhaustion increases, otherwise exhaustion decreases depending on the level of recovery effort (*RE*). Parameters *η* and *π* determine the amount of increase or decrease. The function *Pos(x)* in this formula is defined as the maximum of x and 0. For more details on the model, see (Bosse *et al.*, 2008).

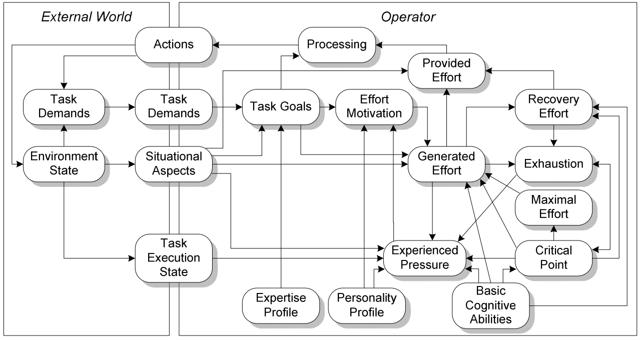


Figure 1. Agent Model for an Operator's Functional State

Experimental setup

First, an overview of the game and its participants is given. The main part of the experiment is a game which combines a shooting task and a calculation task. Thereafter, the procedure of the experiment is explained. A more detailed version can be found in Appendix A: http://www.few.vu.nl/~fboth/ICCM/appendix_A.pdf.
Finally, a description is given of how data from the experiment has been used as input for the work pressure model.

Game and Participants

In the experiment the main task is a shooting game where the goal is to get as many points as possible. Objects (friends and enemies) were falling down in different locations at different speeds. The purpose is to shoot the enemies before they hit the ground. Shooting at a missile is done by a mouse click at a specific location; the missile would then explode exactly at the location of the mouse click. The speed with which the missile reaches this location is 79.6 pixels per second. When an object is within a radius of 50 pixels of the explosion, the object is destroyed. The number of points a participant receives for hitting an enemy is proportional to the proximity of the explosion. When a participant shoots a friend or when an enemy reaches the bottom of the screen, points are lost. When a friendly object reaches the bottom of the screen points are gained. Next to each of the objects, a calculation is written on the screen. A correct calculation indicates that the object is friendly and should not be shot. An incorrect calculation indicates that the object is an enemy and should be shot before it reaches the bottom of the screen. For a demo of the shooting game, see http://www.forcevisionlab.nl/demo/missilecommand.swf.

In the study 31 persons participated (18 males, 13 females, of which 25 students). They ranged in age from 17 to 57 years with a mean age of 26 years. The experiment took approximately 1 hour for which participants received a voucher of 10 euro. In addition, there was a voucher of 100 euro for the one with the best score on the game.

Procedure

For the experiment a 2 factor within subjects design was used. Two different conditions within each participant were tested. In Bosse *et al.* (2008a), two scenarios were simulated using the model. Scenario 1 started with a low task level and continued with a high task level. Scenario 2 started with a high task level and continued with a low task level. Condition was counterbalanced over participants to correct for a possible order effect, such that participants with an odd number started with condition 2 and even numbered participants started with condition 1.

Participants started the experiment with filling out a personality questionnaire with questions from the NEO-PI-R and the NEO-FFI (Costa and McCrae, 1992); with these questions some aspects of each participant's personality were measured, to serve as input for the personality profile of the work pressure model. Neuroticism and extraversion were measured with the NEO-FFI. With the NEO-PI-R vulnerability (part of neuroticism) and ambition (part of conscientiousness) were measured.

After the questionnaire, participants performed three small tests each consisting of 30 trials which were equal between participants. These tests served as input for model validation (see the next subsection and Appendix A for the explanation thereof). Instructions for each test were shown on the screen. The first test was a simple choice Reaction Time test (choice-RT), where a square was presented either left or right from a fixation cross at the centre of the screen. Participants had to react with either the left arrow (when the square was presented left) or the right arrow (when the square was presented right). The second test was a task where calculations were presented. Again, participants had to choose whether the calculation was correct (left arrow) or incorrect (right arrow). The third small test (mouse-RT) was another Reaction Time task; here a circular target was presented somewhere on the screen. Participants had to react quickly and precisely by clicking with the mouse as close as possible to the centre.

After the three small tasks, participants practiced during 3 minutes for the experiment-game described in the previous subsection. The goal of the practice task was familiarize with the shooting and calculation tasks in the game. After practice the participants started the experiment-game with either condition 1 or condition 2, which both took 15 minutes.

From experiment data to work pressure model

In order to validate the model, data from the experiment was used to calculate the values of several concepts of the work pressure model, namely personality profile, basic cognitive abilities (BCA) and expertise profile, following theories from Psychology (Matthews & Deary, 1998; Plomin &

Spinath, 2002; Rose *et al*, 2002; Salgado, 1997). Hereby, several parameters are introduced that need to be estimated by the parameter estimation approach as well. Including this, the number of parameters that should be estimated is 27. For the precise mathematical equations used, see http://www.few.vu.nl/~fboth/ICCM/appendix_D.pdf.

Furthermore, from the experiment data the situational demands can be calculated. Although the scenarios were the same for all participants, the calculated task level could differ due to the performance quality. Therefore, Situational Demands were calculated per time step per participant. According to the model, situational demands and the expertise profile together contribute to task level.

$$TaskLevel = (1.5 - Exp)\cdot SitD \qquad (1)$$

In the experiment, performance quality was measured in terms of efficiency and effectiveness. Efficiency represented the number of missiles necessary to shoot an enemy. Effectiveness was dependent on how close to the object the missile exploded (explosion fraction) and whether an enemy or friend was shot. In case of an enemy being shot:

$$Effectiveness = (1 + explosion\_fraction)/2.0 \qquad (2)$$

Effectiveness was 0 when a friend was shot or an enemy landed. When a friend landed, effectiveness was 1. Using effectiveness and efficiency, the task execution state was calculated:

$$ObjTES = (0.25\cdot efficiency + 0.75\cdot effectiveness)\cdot 2 \qquad (3)$$

Estimation of parameters

This section presents the results of parameter estimation for the work pressure model using two different methods: a gradient-based approach and an approach based on probabilistic search.

Gradient-based parameter estimation

To perform parameter estimation, a method based on the maximum likelihood principle has been applied (Sorenson, 1980). In line with this principle a likelihood function of the measurement data and the unknown parameters is defined. This function is essentially the probability density function of the measurement data given the parameter values $p(z|\theta)$. Furthermore, it was assumed that the measurements contained noise which is zero-mean and has a Gaussian distribution. The measurement data were represented by the random, normally distributed variable z. Such an assumption is often made for dynamic systems in many areas. The parameter vector, which makes the likelihood function most probable to obtain the measurements z (…$\hat{\theta}_{ML}$ .. which maximizes the likelihood function) is called the maximum likelihood estimate; it is obtained by minimizing the error function:

$$E(\theta) = \frac{1}{2}\cdot\sum_{i=1}^{N}(z_i - y_i)^T \cdot R^{-1}\cdot(z_i - y_i) + \frac{N}{2}\cdot \ln|R| \qquad (5)$$

Here the measurements obtained are discrete time, N is the number of measurements, R is the measurement noise covariance matrix. The estimate of R is obtained as

$$\hat{R} = \frac{1}{N}\cdot\sum_{i=1}^{N}(z_i - \hat{y}_i)\cdot(z_i - \hat{y}_i)^T \qquad (6)$$

The maximum likelihood estimates are consistent, asymptotically unbiased and efficient (Sorenson, 1980).

The calculation of the maximum likelihood estimate is performed iteratively. The estimate value at the $(k+1)$ iteration is determined as:

$$\hat{\theta}_{ML}{}^{k+1} = \hat{\theta}_{ML}{}^{k} + [\nabla_\theta{}^2 E(\theta)]^{-1}\cdot[\nabla_\theta E(\theta)] \qquad (7)$$

Here the first gradient is defined as

$$\nabla_\theta E(\theta) = \sum_{i=1}^{N}\left[\frac{\partial y_i}{\partial\theta}\right]^T \cdot R^{-1}\cdot(z_i - y_i) \qquad (8)$$

For the work pressure model the expressions for the partial derivatives w.r.t. the parameters (i.e., sensitivity coefficients) have been obtained analytically (see Appendix B: http://www.few.vu.nl/~fboth/ICCM/appendix_B.pdf).

The analytical determination of the second gradient is more involved, therefore a Gauss-Newton numerical approximation has been used for it:

$$\nabla_\theta{}^2 E(\theta) = \sum_{i=1}^{N}\left[\frac{\partial y_i}{\partial\theta}\right]^T \cdot R^{-1}\cdot\left[\frac{\partial y_i}{\partial\theta}\right] \qquad (9)$$

Such an approximation does not cause a significant error in the parameter estimate. Furthermore, the use of the second gradient speeds up the convergence of the estimation process significantly.

The state values of the system were calculated by numerical integration of the model equations using the 4th order Runge-Kutta method, which has proven to be both accurate and stable. The estimation error is calculated in each iteration as root mean square error:

$$err = \sqrt{\sum_{i=1}^{N}\frac{(z_i - \hat{y}_i)^2}{N}} \qquad (10)$$

The parameter estimation procedure based on the maximum likelihood principle has been implemented using the following algorithm:

```
Algorithm: ML-PARAMETER-ESTIMATION
Input: Initial values of the parameters $\theta^1$, maximal number of iterations itmax; satisfactory error value err_sat; matrix of the input values U; matrix of the output values Z
Output: Maximum likelihood estimate $\theta_{ML}$
1  i=1
2  Until i ≤ itmax perform steps 3-7
3  Calculate the current state of the system using the model equations
4  Calculate the output root mean square error $err^i$ using (10).
5  if err ≤ err_sat, then $\theta_{ML} = \theta^i$; exit endif.
6  if i < itmax, then
     6a  Calculate the noise covariance matrix R using (6)
     6b  Calculate the sensitivity coefficients $\partial y/\partial\theta$
     6c  Calculate the first and second gradients using the formulae (8) and (9) respectively.
     6d  Calculate the parameter values for the next iteration $\theta^{i+1}$ using (7)
   endif
7    i = i+1
8    Find the minimum error $err^m$ in $\{err^i|\ i=1..itmax\}$; then $\theta_{ML} = \theta^m$; exit.
```

The algorithm was implemented in the Matlab 7 environment. The worst case complexity is estimated as O(NN·|θ|·M), where NN is the number of integration points, |θ| is the number of the estimated parameters, M is the number of outputs. The execution of an iteration took less than 2 sec on an average PC.

Simulated annealing

The Simulated Annealing method uses a probabilistic technique to find a parameter setting. In this method a random parameter setting is chosen as the best available parameter setting at the start. Then a displacement is introduced into these settings to generate a neighbor of the current parameter settings in the search space. If this neighbor is found more appropriate representation of the observed human behavior then it is marked as the best known parameter setting otherwise a new neighbor is selected to evaluate its appropriateness. The displacement in the parameter settings depends on the temperature, in case the temperature is higher, the steps will become larger. The temperature at a certain time point for the parameter settings is defined as follows

$$\textit{Temperature} = \textit{computational-budget-left} \cdot \textit{error} \quad (11)$$

Here the computational budget is the number of neighbors to be tested for better approximation. The displacement in the parameter for example γ was derived from following equations selecting any one at random.

$$\gamma=\gamma+\textit{Temperature} \cdot (1\text{-}\gamma) \cdot \textit{random\_no\_between}[0,1] \quad (12a)$$

$$\textit{or } \gamma = \gamma\text{-}\textit{Temperature} \cdot \gamma \cdot \textit{random\_no\_between}[0,1] \quad (12b)$$

The method is described as follows:

Algorithm: SA-PARAMETER-ESTIMATION

Input: Initial randomly selected values of the parameters θ^1, computational budget C; observed human behaviour B;

Output: Best estimate of parameter settings θ_{BE}

```
1   $\theta_{BE}=\theta^1$
2   while $C \geq 0$ perform steps 3-8
3   Choose a random parameter setting $\theta$ in neighbourhood of $\theta_{BE}$ using equation (11 and 12a, 12b).
4   Calculate the output root mean square error err for $\theta$ using (10).
5   Calculate the output root mean square error $err_{BE}$ for $\theta_{BE}$ using (10).
6   if $err \leq err_{BE}$, then $\theta_{BE} = \theta$; $err_{BE} = err$; endif.
7   Decrease C;
8   Temperature = C * $err_{BE}$;
9   output $\theta_{BE}$.
```

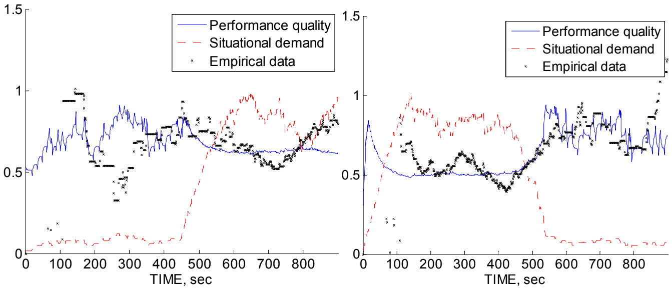


Figure 2. Empirical data and the estimated output performance quality for subject 37 for condition1 (left) and condition 2 (right)

In figure 2 performance quality for subject 37 is shown for computational budget 10000 and 900 observed human behavior. Here it should be noted that graph represents the curve generated with parameter settings producing minimum root mean square error found till the end of computational budget. The algorithm has been implemented in C++ and applied to the work pressure model. If C is computational budget, then the worst case complexity of the method can be expressed as O(C·B), where B is the number of observed behaviors. Here it could be observed that computational complexity of this method is independent number of parameter.

Results of the estimation

The gradient-based and simulated annealing methods have been applied for the estimation of 30 parameters of the work pressure model (see Appendix C: http://www.few.vu.nl/~fboth/ICCM/appendix_C.pdf). The estimation has been performed for 31 subjects, for both experimental conditions. The initial setting of the parameters has been taken from Bosse *et al.* (2008a). This setting is grounded partially in the psychological literature; furthermore it ensures the desired properties of the modeled system. Figure 2 illustrates the empirical data and the estimated output performance quality for subject 37 for both conditions.

The estimation by both methods showed similar behavioral patterns in the output of the model. However, the gradient-based method has a better precision in comparison to the simulated annealing. The root mean square errors calculated in both parameter estimation methods are given in Table 1. To evaluate the quality of estimation also other measures have been used. In particular, the Cramer-Rao bounds provide a useful measure of relative accuracy of the estimated parameters (Sorenson, 1980).

Table 1. Root mean square errors of estimation by the gradient-based (GB) and simulated annealing (SA) methods for all subjects in both experimental conditions

Error range		< 0.1	[0.1, 0.25)	[0.25, 0.4)	> 0.4
Subjects in condition 1	GB	21	11-20, 22, 24-41	-	-
	SA		40	11, 12, 22, 24-26, 30, 32-39, 41	13-18, 20, 21, 28, 29, 31
Subjects in condition 2	GB	12, 15, 18, 20, 21, 23, 27, 30	11, 13, 14, 16, 17, 19, 22, 24-26, 28, 32-41	29, 31	-
	SA	32	17, 26, 30, 31, 34. 35, 37, 40	12, 27, 38, 41	11, 13-16, 18-23, 25, 28, 29, 33, 36, 39

This measure sets a lower bound on the standard deviation of the estimators:

$$\sigma_\theta \geq \sqrt{I^{-1}(\theta)} \qquad (13)$$

Here $I(\theta)$ is the information matrix:

$$(I(\theta))_{ij} = E\left| \frac{\partial^2 \log p(z \mid \theta)}{\partial\theta_i \partial\theta_j} \right| \qquad (14)$$

For efficient estimation the equality holds. Furthermore, for the maximum likelihood method, $I(\theta) = \nabla^2_\theta E(\theta)$, which also needs to be calculated for (9); thus no additional computation effort for the evaluation of this measure is required. Using this measure at least 57% (70% in the best case) of the estimated parameters have been identified as accurate for all subjects in both conditions (relative standard deviation (rsd) ≤ 5%). Other parameters, although less accurate (5% < rsd < 40%) still have a degree of confidence.

Another useful criterion for judging the quality of the estimates is the correlation coefficients among the estimates calculated as:

$$c_{\theta_i\theta_j} = \frac{(I(\theta)^{-1})_{ij}}{\sqrt{(I(\theta)^{-1})_{ii} \cdot (I(\theta)^{-1})_{jj}}} \qquad (15)$$

Only one significant correlation between the parameters A and ϕ has been identified.

The precision of the parameter estimation is essential for prediction of the system dynamics using the model. To examine predictive capabilities of a model, cross-validation is often used. In the cross-validation of the work pressure model the empirical data of the condition 2 have been used for the parameter estimation, whereas the data of the condition 1 were used for validation of the model with the parameter estimates obtained from the condition 1.

The prediction quality was determined by comparing the root mean square errors for both conditions. For most of the subjects (84%) in the GB estimation, prediction errors (Table 2) differ from the estimation errors (Table 1, subjects in condition 1) insignificantly (less than 10%). Furthermore, also cross-validation was performed, in which data from one of the settings were used for parameter estimation and data from the other setting were used for validation (Figure 3).

Table 2. Prediction errors of estimation by the GB and SA methods for all subjects in condition 1 using the estimated parameters from condition 2

Error range	< 0.1	[0.1, 0.25)	[0.25, 0.4)	> 0.4
GB	21	12-20, 22, 24-30, 34-40	11, 31, 32, 41	33
SA	-	17, 26, 31, 32, 37, 40	12, 13, 22, 25, 28, 30, 34, 35, 38, 41	11, 14-16, 18-21, 29, 33, 39

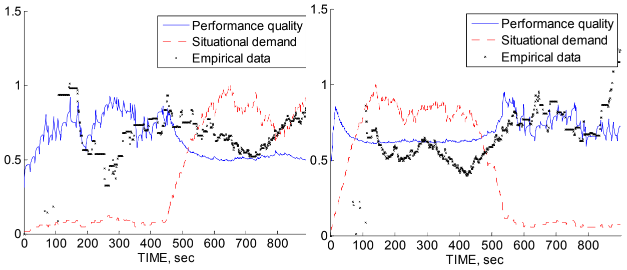


Figure 3. Predicted dynamics for subject 37 in condition 1 using the estimated parameters from condition 2 (left) and in setting 2 using the parameters from setting 1 (right)..

Verification of Properties

This section focuses on logical verification, another approach which has been used to validate the model. The idea is that properties are identified that are entailed by the work pressure model, and these properties are verified against the empirical data that has been obtained. In order to conduct such an automated verification, the properties have been specified in a language called TTL (for Temporal Trace Language, cf. Bosse *et al.*, 2008b) that features a dedicated editor and an automated checker. This predicate logical temporal language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to *states* of the world, *time points* and *traces*, i.e. trajectories of states over time. In addition, *dynamic properties* are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate |=, i.e., state(γ, t) |= p denotes that state property p holds in trace γ at time t. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as ¬, ∧, ∨, ⇒, ∀, ∃. For more details on TTL, see (Bosse et al., 2008a).

Three main properties have been identified that follow from the work pressure model. The first property specifies that performance quality decreases in case a task level in a certain range is experienced:

P1(min_level, max_level, d, x)

*If at time point t1 the task level is tl and the performance quality pq, and tl is in the range [min_level max_level], and until t1+d the task level does not cross these boundaries, then there exists a time point t2> t1 at which the performance quality is at most x * pq.*

```
P1(min_level, max_level, d, x) ≡
∀γ:TRACE, t1:TIME, pq1:REAL
  [ state(γ, t1) |= has_value(performance_quality, pq1) &
  ∀tl:REAL, t':TIME ≥ t1 & t' ≤ t1 + d
     [state(γ, t') |= has_value(task_level, tl) ⇒
      [ tl ≤ max_level & tl ≥ min_ level ] ]
  ⇒ ∃t2:TIME > t1, pq2:REAL
      [state(γ, t2) |= has_value(performance_quality,pq2) &pq2 ≤ x * pq1]
```

This property has been verified using the following values: min_level is set to 20% above BCA, max_level is set to the highest task level encountered in the experiment, the duration d is set to 60 time steps (i.e. a minute real time), and x is set to 1 (i.e. performance quality should never go up, but can remain the same). These settings follow the model: in case a task level above BCA is experienced, the human becomes exhausted, and the quality can no longer go up. Results show that this property is satisfied in **60%** of the empirical traces.

The second property concerns the opposite: in cases where there is a task level between certain boundaries, the performance quality should be at least as high as before the period (note that the formal form has been omitted for the sake of brevity):

P2(min_level, max_level, d, x)

*If at time point t1 the task level is tl and the performance quality pq, and tl is in the range [min_level, max_level], and until t+d the task level does not cross these boundaries, then there exists a time point t2> t1 at which the performance quality is at least x * pq.*

Using the following settings: max_level at 20% below BCA, min_level is set to 0 and d and x the same as for the previous property, this property is satisfied in **45%** of the cases. In case a task level is experienced which is somewhat below the highest task level that can be handled without exhaustion building up (i.e. the BCA), then the performance will get better, or at least stay the same (as there is no exhaustion).

The final property which has been verified concerns performance quality being higher for cases whereby there is a lower task level:

P3(low_level, high_level)

In case the task level at a time point t1 is tl1, and at a time point t2 the task level is tl2, and tl1 > high_level and tl2 < low_level, then there exists a time point t' > t1 and there exists a time point t'' > t2 such that the performance quality at time point t' is lower than the performance quality at time point t''.

Using a low_level of 20% below BCA, and a high_level of 20% above the cognitive abilities, this property is satisfied in **60.7%** of the cases. The property complies with the model, because a task level beyond BCA results in exhaustion leading to a worsened performance, which is not the case for a task level far below BCA. In total, **25.0%** of the cases comply with properties P1, P2, and P3.

Discussion and conclusions

To reason about the human behavior and support possibilities personal assistant agents often use (cognitive) models. To ensure that support is provided by agents in a timely and knowledgeable manner, such models should be accurate and validated. This paper contributes an approach to validate the work pressure model. In the following the performed validation steps of the approach are discussed.

The experience with the experiment was that the participants were very motivated to perform well on the main task. This was not only due to the reward; they were also enthusiastic about the game itself. In order to keep the learning effect to a minimum and to maintain the participants' concentration, every participant performed only two sessions of the 15 minute game. However, precision of parameter estimation will increase when measurements of more within-subject conditions are taken.

The results obtained for the parameter estimation are satisfactory. However, a number of parameters (35% in average) were evaluated as less accurate, and, therefore, less reliable. Partially this can be explained by a large overall number of parameters being estimated. Most of the less precise parameters have a weak relation to the measured output (e.g., noise sensitivity) Furthermore, since the empirical data were collected based on irregular events (i.e., actions of humans), some intervals contained the amount of information insufficient for estimation. Despite this, as shown in the paper, the models with estimated parameters demonstrated good predictive capabilities in the cross-validation, which is a strong indicator of the model validity.

The trends as predicted by the model have also been verified against the empirical material. The results show that a reasonable percentage of the traces satisfy each of these individual properties. The combination of all three properties is however only satisfied in 25% of the cases, which can mainly be attributed to the aforementioned collection based on irregular events, making the data obtained more prone to sudden changes.

The topic of model validation received much attention in the areas of Psychology and Social Science. In particular, a validation approach from (Yilmaz, 2006) distinguishes the validation phases similar to the ones considered in the paper (e.g., conceptual and operational validation); however, the precise elaboration of the phases is focused largely on social processes, which are not relevant for our work. Furthermore, examples of model validation are found in psychology, e.g. on the subject of visual attention (Parkhurst et al., 2002), however often no parameter estimation is involved.

In the future research the considered parameter estimation methods will be extended for the case of real-time estimation, which accounts for human learning. Furthermore, a personal assistant agent will be implemented that is able to monitor and balance work pressure of the human in a timely and knowledgeable manner.

References

Bosse, T., Both, F., Lambalgen, R. van,, & Treur, J. (2008). An Agent Model for a Human's Functional State and Performance. In: Jain, L. *et al.* (eds.), *Proceedings of International Conference IAT'08*. (pp. 302-307). IEEE Computer Society Press.

Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., & Treur, J. (2008). Specification and Verification of Dynamics in Agent Models. *International Journal of Cooperative Information Systems*, *18 (1)*, 167-193.

P.T. Costa Jr., & R.R. McCrae. (1992). *Revised NEO Personality Inventory (NEO-PI-R) and the NEO Five-Factor Inventory (NEO-FFI) professional manual* (Psychological Assessment Resources). Odessa, FL.

Hancock, P.A., Williams, G., Manning, C.P., & Miyake, S. (1995). Influence of task demand characteristics on workload and performance. *The International Journal of Aviation Psychology, 5(1)*, 63-86.

Hill, D.W. (1993). The critical power concept. *Sports Medicine*, vol.16, pp. 237-254.

Hockey, G.R.J. (1997). Compensatory control in the regulation of human performance under stress and high workload: a cognitive-energetical framework. *Biological Psychology 45*, 73-93.

Kozierok, R., & Maes, P. (1993). A Learning Interface Agent for Scheduling Meetings. *Proceedings of the 1st International Conference on Intelligent User Interfaces* (pp. 81-88).

Maheswaran, R., Tambe, M., Varakantham, P., & Myers, K. (2003). Adjustable autonomy challenges in personal assistant agents: A position paper. *Proceedings of the AAMAS'03 Workshop on Agents and Comp. Autonomy* (pp.187–194).

Matthews, G., & Deary, I.J. (1998). *Personality traits*. Cambridge, UK: Cambridge University Press.

Mitchell, T., Caruana, R., Freitag, D., McDermott, J., & Zabowski, D. (1994). Experience with a Learning Personal Assistant. *Communication of the ACM 37(7)*, 81-91.

Parkhurst, D., Law, K., & Niebur, E. (2002). Modeling the role of salience in the allocation of overt visual attention. *Vision Research 42(1)*, 107-123.

Plomin, R., & Spinath, F.M. Genetics and general cognitive ability. *Trends in Cognitive Science 6(4)*, 369-176.

Rose, C.L., Murphy, L.B., Byard, L., & Nikzad, K. (2002). The role of the Big Five personality factors in vigilance performance and workload. *European Journal of Personality 16*, 185-200.

Salgado, J.F. (1997). The five factor model of personality and job performance in the European community. *Journal of Applied Psychology 82* (1): 30–43.

Sorenson, H.W. (1980). *Parameter estimation: principles and problems*. Marcel Dekker, Inc., New York.

Yilmaz, L. (2006). Validation and verification of social processes within agent-based computational organization models. *Computational and Mathematical Organization Theory*, *12*, 283-312.

The Influence of Spreading Activation on Memory Retrieval in Sequential Diagnostic Reasoning

Udo Böhm (udo.boehm@s2006.tu-chemnitz.de)
Katja Mehlhorn (katja.mehlhorn@phil.tu-chemnitz.de)
Chemnitz University of Technology, Department of Psychology,
Wilhelm-Raabe-Str. 43, 09107 Chemnitz, Germany

Abstract

A crucial aspect of diagnostic reasoning is the integration of sequentially incoming information into a consistent mental representation. While research stresses the importance of working memory in such a task, it is not clear how the information represented in working memory can guide the retrieval of associated information from long-term memory. Factors that might influence this retrieval are the amount of information currently in the focus of attention (Lovett, Daily & Reder, 2000) and the time since the information first became available (Wang, Johnson & Zhang, 2006). By comparing the results of different ACT-R models to human data from a sequential diagnostic reasoning task, we show that these factors do not necessarily influence the retrieval. Our findings rather suggest that in a task where information has to be actively maintained in working memory, each piece of this information has the same potential to activate associated knowledge from long-term memory, independent from the amount of information and the time since it entered working memory.

Keywords: information integration; diagnostic reasoning; spreading activation; working memory; ACT-R

Introduction

Generating and evaluating explanations for data extracted from the environment is a key component of many everyday tasks like medical or technical diagnosis and social attribution. This kind of reasoning is often called diagnostic or abductive reasoning (Josephson & Josephson, 1994; Johnson & Krems, 2001). For example, in medical diagnostic reasoning a physician needs to find the best explanation for the set of symptoms displayed by a patient. In such a task, information (e.g., the patients' symptoms) often becomes available step by step. The reasoner needs to integrate this information into a consistent mental representation that is updated every time a new piece of information becomes available. To find an explanation for the observed information, associated information (e.g., potential explanations for a set of symptoms) needs to be retrieved from memory.

Working memory has been proposed to play a crucial role in such a task. It is needed to keep track of the subsequently gathered information (Baumann, 2001) and it might hold possible explanations for this information retrieved from the reasoners long-term memory (Baumann, 2001; Thomas, Dougherty, Sprenger & Harbison, 2008). However, it is not clear how the information is represented in working memory over the course of the task and how that influences the retrieval of associated information. The goal of this paper is to develop a better understanding of how information in working memory guides the retrieval of associated knowledge from long-term memory in a sequential diagnostic task.

To achieve this, we implement different assumptions about the retrieval in ACT-R models and compare the model data to human data from a diagnostic reasoning experiment (Bauman, Mehlhorn & Bocklisch, 2007). Before we turn to describing the models, results and the related theories in detail, we want to point out that abduction in general and diagnosis in particular are complex tasks. In this paper we focus on memory retrieval, as it is a key aspect of these tasks. However, one should keep in mind that the models are a simplification of the task, as they ignore more deliberate processes (as e.g. described by. Johnson & Krems, 2001)

Theories

Human memory might be understood as a set of elements, each of which is assigned a specific activation value. In this conception, a subset of the elements being activated above some specific threshold constitutes working memory (e.g., Just and Carpenter, 1992). In diagnostic reasoning, observations (e.g., the symptoms presented by a patient) and their possible explanations (e.g., diseases causing these symptoms) are held in memory. Given such a knowledge structure, observations can serve as a cue for the retrieval of associated knowledge. That is, information in the focus of attention (e.g., the symptoms presented by a patient) initiates a spreading activation process that activates associated information in long-term memory. Although this assumption has been made by various researchers (e.g., Arocha & Patel, 1995; Bauman et al., 2007; Thomas et al., 2008), the nature of this spreading activation process is not yet fully understood.

It has been argued that the total amount of activation that can be spread from working memory is limited and will be equally divided among the elements that spread activation (Lovett et al., 2000). Thus, the amount of activation spread by each single piece of information will depend on the amount of information that is currently held in the focus of attention. It has also been argued that information in working memory is subject to decay (e.g., Wang et al., 2006). That means that the activation spread from a specific piece of information in working memory to associated knowledge in long-term memory depends on the time since the information became available.

As noted above, in diagnostic reasoning the reasoner needs to find an explanation for information observed from the environment. As new information often only becomes available over time, the amount of information in working memory (i.e. the number of observations that need to be explained) as well as the 'age' of information in working memory (i.e. the time since the observation was made) varies. Therefore, sequential diagnostic reasoning is a field especially suited to test assumptions about information representation in working memory and its effect on retrieval from long-term memory.

To test the different possibilities, we designed different cognitive models using ACT-R. In its current implementation, ACT-R's declarative memory system consists of chunks (facts like *Influenza can cause cough and fever*) that represent declarative knowledge. Access to these memory elements depends on their activation (Anderson, 2007; Lovett et al., 2000). For each chunk, this activation is computed as the sum of its base-level activation and the associative activation from the current context (i.e. spreading activation). The base-level reflects the chunk's previous usefulness in terms of the number of times it was used and the time that has elapsed since. The associative activation reflects a chunk's usefulness in the current context and is computed as the product of the activation spread to it from some specific source (see below) and the strength with which it is related to that source (Anderson, 2007).

The source that provides the activation to be spread is information about the current problem or task. This information is represented in one of ACT-R's modules, the imaginal module. This module holds a mental representation of the problem currently in the focus of attention (Anderson, 2007). In a sequential diagnostic reasoning task, it is assumed that the imaginal module thus holds the information about all the data gathered so far. This information can then spread activation to associated knowledge held in declarative memory. To test the nature of the representation of information in working memory we implemented different modes of this spreading activation process in four ACT-R models.

The first model addresses the question if the amount of information in the focus of attention should influence spreading activation. To test this, we used the standard implementation of spreading activation in ACT-R. In this implementation, the total amount of spreading activation is assumed to be equally divided among the information stored in the source chunk (Lovett et al., 2000). Thus, the activation spread by each single piece of information depends on the amount of information in the focus of attention. The more slots the source chunk contains, the less activation can be spread by each single slot.

The second and the third model address the question whether information in working memory is subject to decay. In the second model, we use an equation for decaying activation proposed in a constraint satisfaction model (UECHO) by Wang et al. (2006). It assumes spreading activation to decay in curvilinear, negatively accelerated manner. Thus, information in working memory increasingly loses its impact over time. To test if decay needs to be negatively accelerated as proposed by Wang et al., or if a more simple assumption of decay would be sufficient, we implemented a third model using a linear decay function. In this model, information in working memory loses its ability to spread activation linearly over time.

For being able to better access the explanatory power of the above models, we implemented a fourth model that serves as a control model. This model is most parsimonious, as it assumes a constant amount of activation spread by each piece of information in working memory. Thus, in this model, spreading activation neither depends on the amount of information held in working memory, nor on the time since the information became available.

Experiment

Human data was obtained in an experiment using an artificial diagnosis task (see also Baumann et al., 2007). Participants were told to imagine they are a doctor in a chemical plant and had to diagnose which chemical their patient had been in contact with. Therefore, they learned a knowledge structure consisting of nine different chemicals grouped into three categories. Chemicals were named with single letters and each chemical caused three to four symptoms (Table 1). Each symptom could be associated with two, three or six chemicals. Participants acquired this knowledge in an extensive training session, where they had to solve various tasks until reaching proficient performance.

In two subsequent experimental sessions, participants then worked on 340 diagnostic reasoning trials. In each of these trials, symptoms belonging to a chemical were presented sequentially on the screen. At the end of each trial, participants were asked for their diagnosis (see Figure 1 for a sample trial). As each symptom had several possible causes, only the combination of symptoms in a trial allowed for unambiguously identifying the correct diagnosis. With the number of observed symptoms, the number of plausible diagnoses could be narrowed down, leaving the correct diagnosis (consistent to all symptoms) at the end of the trial.

To track the activation of different explanations during the course of this reasoning task, a probe reaction task was used. After one of the symptoms in each trial, a single letter was shown. This could either be the name of one of the chemicals or not. Once the letter was presented on the screen, participants were to indicate as fast as possible whether it was a chemical or not. The idea of this probe reaction task is based on the idea of lexical decision tasks (e.g., Meyer & Schvaneveldt, 1971) according to which participants should respond faster to a probe that is activated higher in memory than to a probe of low activation. Using this measure, it was possible to monitor the activation of explanations over the course of the sequential reasoning task with as little impact on the task itself as possible.

Table 1. Summary of the material participants had to learn (original material in German).

Group	Chemical	Symptoms
Landin	B	cough, short breath, headache, eye inflammation
	T	cough, short breath, headache, itching
	W	cough, eye inflammation, itching
Amid	Q	skin irritation, redness, headache, eye inflammation
	M	skin irritation, redness, headache, itching
	G	skin irritation, eye inflammation, itching
Fenton	K	diarrhea, vomiting, headache, eye inflammation
	H	diarrhea, vomiting, headache, itching
	P	diarrhea, eye inflammation, itching

Three different types of explanations were tracked in the experiment. First, the probed explanations could be an element of the current explanation (that is they were consistent to all symptoms observed so far). These probes are termed *'relevant'*. Second, the probed explanation could be an explanation that was never considered during the current trial. These probes were termed *'irrelevant'*. Third, the probed explanation could have been considered relevant until some evidence inconsistent with that explanation forced participants to reject it. These probes are called *'rejected'*. Rejected probes additionally varied with respect to the time since their rejection. They could be probed directly after rejection (*just rejected*); one symptom after rejection (*rejected 1 symptom* ago); or two symptoms after rejection (*rejected 2 symptoms ago*).

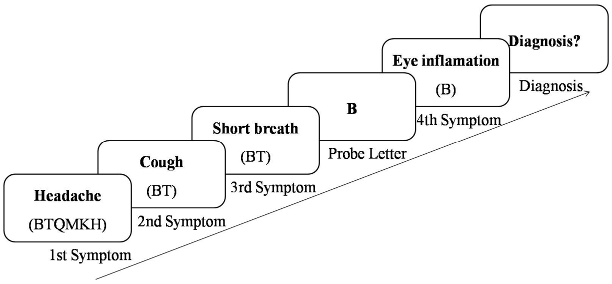


Figure 1. Sample trial from Baumann et al. (2007) with B as a relevant probe. (Letters in parentheses represent relevant explanations).

Models

Four ACT-R models using different implementations of spreading activation from working memory were designed to test the assumptions presented above. Because ACT-R's memory system is dependent on patterns of retrieval time, the temporal order of events was modeled as closely as possible to the actual experiment. Thus, the models went through the same trials as human participants. After one symptom of each trial a probe was presented and the models had to indicate whether it was a chemical or not by typing 'Y' for Yes and 'N' for No respectively. At the end of each trial the models typed the letter representing the diagnosis. This was accomplished using ACT-R's perceptual and motor modules that allow for modeling time to process visual stimuli and performing key strokes.

As participants had received extensive training on the task prior to the experiment, the base levels of the chunks representing symptoms and probes or diagnosis respectively were all set to the same high level.

To implement the integration of the sequentially presented symptoms we assumed one chunk to be placed in the imaginal module at the beginning of each trial. Over the course of the trial, the slots of this chunk were successively filled with the symptoms seen thus far. As noted above, we assumed the imaginal module to be the source of spreading activation, thus, only information stored in this module could spread activation to associated concepts in declarative memory.

To solve the probe task, the model had to retrieve the explanation-chunk representing the probe letter. Due to spreading activation from the observed symptoms stored in the imaginal module, explanations associated to these symptoms received more spreading activation and could therefore be retrieved faster. Thus, as in human participants, the time to respond to a probe could be used as a measure for the activation of explanations in memory. As soon as the model was asked for the final diagnosis, it attempted to retrieve an explanation-chunk from memory. As the explanation most consistent to all observed symptoms obtained the highest spreading activation, this explanation was the one most likely to be retrieved.

To model the different assumptions concerning the nature of activation processes in working memory, we varied the implementation of spreading activation from the imaginal module between the different models as described in the following.

Model 1. In the first model, the amount of activation spread by each symptom depends on the number of symptoms observed so far. The imaginal module (holding the observed symptoms) can spread a certain amount of maximum activation that is equally divided among the symptoms:

$$W_j = W/n \qquad (1)$$

with W_j being the spreading activation associated with the j^{th} symptom, W being the total amount of activation for the module and n being the number of symptoms hold by the module. This is the standard solution implemented in ACT-R. Thus, after the first symptom is presented, there is only one chunk in the imaginal module that can spread activation and thus, has a full spreading activation (set to 1). The more symptoms placed in the module over the course of the trial, the less activation is spread by each of these symptoms.

Model 2. For the second model, we implemented a function that assigned pre-specified amounts of activation to be spread to the slots of the source chunk. The values associated with the slots were computed using a formula proposed for the decay of information in a constraint satisfaction model (Wang et al., 2006; see also Mehlhorn & Jahn, 2009) that assumed a non-linear negatively accelerated decay:

$$W_j = W_{j-1}(1-d)^{\sqrt{t}} \qquad (2)$$

where W_j is the spreading activation associated with the j^th^ symptom, d denotes a decay parameter that was set to 0.4 and t denotes the time that has elapsed since the trial started. Thus, the most recent symptom always spreads a full amount of activation (set to 1). Over the course of the task, symptoms spread less activation the longer they are kept in the imaginal module.

Model 3. The third model also used a function assigning pre-specified decaying amounts of activation. However, in this model we implemented a linear instead of a negatively accelerated decay. To make sure that not the total amount of the decay, but only the slope of the decay function would influence the outcome, we used equal values as in Model 2 for the most recent and the oldest symptom:

$$W_j = W_1-(j-1)((W_1-W_4)/3) \qquad (3)$$

with W_j again being the spreading activation associated with the j^{th} symptom and W_1 and W_4 being the spreading activation values for the most recent and the latest symptom as computed by formula (2). Thus, in this model the activation spread by symptoms decays away in a linear manner over time.

Model 4. A constant amount of activation associated with each slot of the source chunk was implemented in the fourth model. Thus, W_j was set to a fixed value of 0.16 that had shown to provide a good fit to the human data. Thus, in this model, activation spread by a piece of information in the imaginal module neither depends on the amount of information in the module, nor on the time since the information first entered the module.

Results

All four models were compared to the results produced by human participants on four dependent measures, namely the accuracy that was reached in the diagnosis and the probe task and the average reaction times for correct responses in these two tasks.

Diagnosis Task. Table 2 shows the mean accuracies and the mean reaction times for the diagnosis task. All models were able to solve the diagnosis task reaching very good to perfect performance. Inspecting the reaction times for correct diagnoses reveals that all models produced about the same reaction times as human participants.

Table 2. Mean accuracies and mean reaction times by models and human participants in the diagnosis task.

	Mean accuracy (%)	Mean RT (ms) - correct diagnoses
Participants	96.1	608.09
Model 1	100	606.87
Model 2	98.4	571.21
Model 3	99.1	555.13
Model 4	100	658.31

Probe Task. To analyze the accuracy of the probe task, for human data as well as for the models' data, only trials with correct final diagnoses were used. To analyze reaction times to probes, trials on which either the diagnosis or the probe response was wrong, were excluded. This was done because for human participants it remains unclear what caused the wrong diagnosis or the wrong probe response. For example, a participant might have missed a symptom and thus reached a wrong conclusion, implying that the activation measured in the probe task is not the activation of the target letter but rather that of another, possibly irrelevant one.

Human participants responded correctly to the probes in 93.1% of the trials whereas all models reached 100% accuracy. Reaction times for the different probe types are illustrated in Figure 2. For all probe types, Model 4 fits the human data best. The other models deviated more from the human data, which is not only evident in overall faster reaction times, but also in the less well fitting patterns. The different fits are reflected by the R^2 between participants' data and the modeling results as well as the RMSSDs; being $R^2 = .35$ and RMSSD = 2.75 for Model 1, $R^2 = .37$ and RMSSD = 3.00 for Model 2, and $R^2 = .44$ and RMSSD = 3.58 for Model 3, whereas Model 4 reached a R^2 of .80 and a RMSSD of .85.

As can be seen in Figure 2, for *relevant* probes, participants' reaction times decreased the closer the probe was presented to the end of the trial. Model 4 produced a pattern close to the participants' data. In all other models, reaction times were too fast at the beginning of the trials and did not change substantially during the trials, indicating that the earlier symptoms were overweighed. It is notable that none of the models fit the positive acceleration (that is, a sudden drop in reaction times from symptom 3 to symptom 4) of the participants' data.

For *irrelevant* probes, participants' response times decreased slowly over the trial. The models' reaction times decreased as well, but except for Model 4, this decrease was much faster than for the participants. Again, the slopes of the curves differed between all models and the participant data. Participants reacted increasingly faster towards the end of the trial, while the models' reactions decreased asymptotically toward some value.

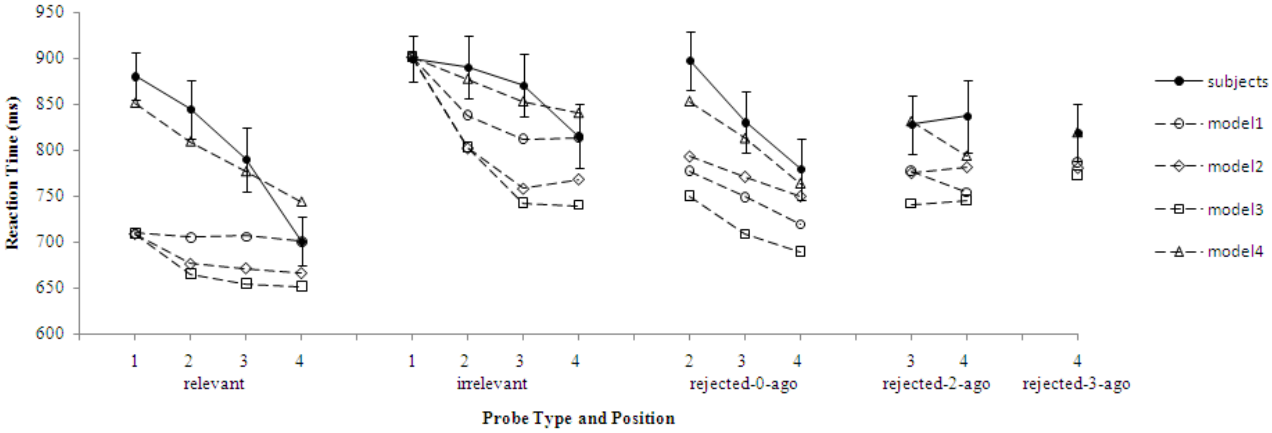


Figure 2. Models' and participants' reaction times for different probe types at different probe positions.

For the *just rejected* probes, all models' patterns matched the pattern provided by humans fairly well. However, the models' responses were too fast. Again, the fourth model's reaction times lie closest to the human data. For the probes rejected one symptom ago, all models' patterns again roughly matched the ones provided by human participants; except that Model 4 showed a slight decrease instead of an increase in the reaction times. Despite the difference in the slope, Model 4 again produced reaction times closest to the human data. Also for probes rejected two symptoms ago, the best fit to human data was provided by Model 4.

Discussion

In this paper, we explored the influence of the implementation of different spreading activation processes in working memory during a diagnosis task. In the task, sequentially presented information needed to be integrated to find an explanation most consistent to all pieces of information. We compared the data provided by four ACT-R models that utilized different patterns of spreading activation to human data on several dependent measures. The analysis of diagnostic performance and the probe accuracy was important to show that all models were able to solve the task. However, the most interesting dependent variable is the probe reaction time. It not only provides a measure for how strongly different types of explanations are activated by the observed symptoms, but also how this activation changes over time.

As the results show, neither the standard implementation of ACT-R (Model 1), assuming the amount of spreading activation in the focus of attention to depend on the amount of information held in the source chunk nor models assuming the spreading activation of information in working memory to decay away with time (Models 2 and 3), could account for the patterns found in human data. Varying the pattern of the decay function from a negatively accelerated decay in Model 2 to a linear decay in Model 3 also had no substantial effect on the model fit. Concluding, none of the models assigning varying activation-values to the information held in working memory were able to fit the data.

Contrary to these models, our fourth model provided a pattern very close to the one provided by human participants. This model assumed the amount of spreading activation associated with each piece of information in working memory to be constant. Before discussing possible implications of this finding, we would like to address several potentially critical aspects of our approach.

One could argue that the bad fit of the Models 1, 2 and 3 might only be due to the high base levels assigned to the diagnosis chunks, thus causing the reaction times to be too short. To rule out this possibility, we also implemented the three models with lower base-levels. However, this did not improve the models' fit, because it did not affect the pattern of the response times, but only the absolute level.

Another possible source of criticism might be the different amount of total spreading activation that was used for the different models. That is, for example in Model 2, the sum of all activations assigned to the different slots of the chunk in the imaginal module was 1.56, whereas the total spreading activation in Model 1 was 1. To rule out possible criticism related to this point, we also implemented all four models in a way such that the total spreading activation was constant across the models. This, again, did not change much about the general data pattern.

Conclusions

Our results have several interesting implications. First, they question the implementation of spreading activation currently used in ACT-R. Second, they question the assumption of decay in working memory as proposed in some constraint satisfaction models. Why could those theoretical assumptions not be confirmed by our data? Does the amount of information in working memory really have no impact on how much activation can be spread by each piece of information? And is information in working memory really not subject to decay?

We would answer both questions with no. The results do neither implicate that there is no overall limit to the amount of activation spread from working memory nor that there is no decay. In our task, participants had to maintain a relatively small amount of information in working memory (up to four symptoms). This lies within the general range of working memory capacity (cf. Cowan, 2000). Thus, our results do not question that the total amount of activation spread from the focus of attention is related to working memory capacity (e.g., Lovett et al., 2000). Rather, this spreading activation might be assigned to the information in the focus of attention in a different way. That is, until the total capacity of working memory is reached, each piece of information seems to spread the same amount of activation.

Moreover, only information that is not currently held in the focus of attention might be subject to processes of decay. That means, as soon as some piece of information becomes irrelevant to the current task or as the amount of information in the focus exceeds its limited capacity, this information might decay away. However, in our task, the information neither became irrelevant nor did it exceed the capacity of working memory during the whole reasoning process. When new symptoms are observed, the reasoner needs to integrate them with earlier symptoms to find an explanation consistent to all symptoms. Therefore, the older symptoms need to be actively maintained, and thus they do not decay.

An interesting question for further research would be to take a closer look at what happens when the amount of information to be actively maintained during the task exceeds working memory capacity. As several authors suggest, in such cases the least activated information would be dropped from working memory (e.g., Thomas et al., 2008; Chuderski, Stettner & Orzechowski, 2006). Thus, this information should no longer be able to spread activation to associated information in long-term memory but instead it should become subject to decay.

Concluding, our results shed some light on the representation of information in working memory during a sequential diagnostic reasoning task. They suggest that in such a task, each piece of this information has the same potential to activate associated knowledge from working memory. It will be an interesting question for further research to determine in how far this finding can be generalized from diagnostic reasoning to other tasks that require information to be actively maintained in working memory.

Acknowledgements

We thank Jelmer Borst, Niels Taatgen, Christian Lebiere and four anonymous reviewers for their helpful comments on the models and on an earlier version of this paper.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.

Arocha, J. F., & Patel, V. L. (1995). Construction-integration theory and clinical reasoning. In C. A. Weaver, III, S. Mannes & C. R. Fletcher (Eds.), *Discourse comprehension: Essays in honor of Walter Kintsch.* (pp. 359-381). Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc.

Baumann, M. (2001). *Die Funktion des Arbeitsgedächtnisses beim abduktiven Schließen: Experimente zur Verfügbarkeit der mentalen Repräsentation erklärter und nicht erklärter Beobachtungen*. Doctoral dissertation, Chemnitz University of Technology, Germany.

Baumann, M.., Mehlhorn, K., & Bocklisch, F. (2007). The activation of hypotheses during abductive reasoning. *Proceedings of the 29th Annual Cognitive Science Society* (pp. 803-808).

Chuderski, A., Stettner, Z., & Orzechowski, J. (2006). Modeling individual differences in working memory search task. *Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 74-79).

Cowan, N. (2000). The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behavioral and Brain Sciences, 24*, 87-185.

Johnson, T. R. & Krems, J. F. (2001) Use of current explanation in multicausal abductive reasoning. *Cognitive Science, 25*, 903-939.

Josephson, J. & Josephson, S. G. (1994). *Abductive inference: Computation, philosophy, technology.* Cambridge University Press.

Just, M A. & Carpenter, M. A. (1992). A capacity theory of comprehension: individual differences in working memory. *Psychological Review, 99*, 122-149.

Lovett, M. C., Daily, L. Z. & Reder, L. M. (2000). A source activation theory of working memory: cross-task predictions of performance in ACT-R. *Cognitive Systems Research, 1*, 99-118.

Mehlhorn, K. & Jahn, G. (2009). Modeling sequential information integration with parallel constraint satisfaction. To appear in: *Proceedings of the 31st Annual Cognitive Science Society*.

Meyer, D.E., & Schvaneveldt, R.W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology, 90,* 227-234.

Thomas, R. P., Dougherty, M. R., Sprenger, A. M. & Harbison, J. I. (2008). Diagnostic hypothesis generation and human judgment. *Psychological Review, 115*, 155-185.

Wang, H., Johnson, T. R., & Zhang, J. (2006). The order effect in human abductive reasoning: an empirical and computational study. *Journal of Experimental & Theoretical Artificial Intelligence. 18(2)*, 215–247

Predicting Interest: Another Use for Latent Semantic Analysis

Thomas J. Connolly (connot3@rpi.edu)
Vladislav D. Veksler (vekslv@rpi.edu)
Wayne D. Gray (grayw@rpi.edu)
Department of Cognitive Science, Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180 USA

Abstract

Latent Semantic Analysis (LSA) is a statistical technique for extracting semantic information from text corpora. LSA has been used with success to automatically grade student essays (Intelligent Essay Scoring), model human language learning, and model language comprehension. We examine how LSA may help to predict a reader's interest in a selection of news articles, based on their reported interest for other articles. The initial results are encouraging. LSA (using default corpus and setup) can closely match human preferences, with RMSE values as low as 2.09 (human ratings being on a scale of 1-10). Additionally, an Adapting Measure (best parameters for each individual) produced significantly better results, RMSE = 1.79.

Keywords: Adapting Measure; Latent Semantic Analysis; LSA; human interest prediction; predicting ratings; news articles

Introduction

The ability to accurately predict user preferences is valuable to any system that strives to deliver meaningful data based on a single query. It allows improved accuracy in the results delivered and by extension, superior service to a system's users. One example of this is the Netflix's Cinematch algorithm, which uses linear statistical models to provide their users with estimates of how much they will enjoy a certain movie given their prior movie preferences (Netflix, 1997). Latent Semantic Analysis (LSA; Landauer & Dumais, 1997), a technique designed to find relations between bodies of text, may offer a suitable alternative for rating text documents. LSA has been employed in a similar manner to predict grades for student essays (Foltz, Laham, & Landauer, 1999), making it worth exploring its worth as a model for user preferences. This paper examines LSA's capability to predict user preferences for news articles and outlines an experiment designed and used to this end. The problem space was defined by 3 factors. First, we examine the effect of using different amounts of the articles' content (title only versus title + content) on prediction accuracy. Second, we examine how different methods for predicting interest compare (e.g. average rating of 3 closest articles, weighted average of 9 closest ratings, etc.). Lastly, we evaluated nomothetic (one size fits all) and idiographic (tailored to individuals) approaches to predict user preferences.

Background

Latent Semantic Analysis Latent Semantic Analysis (LSA; Landauer & Dumais, 1997) is a statistical technique for extracting semantic information from text corpora. It is a powerful technique that has been used with success for automatically grading student essays (Landauer & Dumais, 1997), to model human language learning (Landauer & Dumais, 1997), to model language comprehension (Lemaire, Denhiere, Bellissens, & Jhean-Iarose, 2006), and more.

Although this paper focuses on LSA, other techniques for modeling the human semantic space may be appropriate (Blei & Lafferty, 2006; Blei, Ng, & Jordan, 2003; Griffiths & Steyvers, 2004; Lindsey, Stipicevic, Veksler, & Gray, 2008; Matveeva, Levow, Garahat, & Royer, 2005; Veksler, Govostes, & Gray, 2008), and will be assessed in future work. Moreover, we examine only one of many possible LSA spaces, based on the work of Landauer & Dumais (1997), constructed based on the TASA corpus (Zeno, Ivens, Millard, & Duvvuri, 1995).

The TASA corpus contains a body of text which represents a collection or reading material that a college freshman should be familiar with (Zeno, Ivens, Millard, & Duvvuri, 1995). The LSA-TASA space has been used frequently as a model of human semantics (e.g. Griffiths, Steyvers, & Tenenbaum, 2007; Landauer & Dumais, 1997; Veksler, Govostes, and Gray, 2008;), and is an appropriate first model to use for current research. However, corpus selection makes a great difference for any semantic modeling (Lindsey, Veksler, Grintsvayg, and Gray, 2007), and other corpora will be employed in future research to further improve predictions of human ratings.

Intelligent Essay Scoring Intelligent Essay Scoring, in particular the Intelligent Essay Assessor (IEA) is relevant to current work. Intelligent Essay Assessment, in short, utilizes LSA to grade student essays by comparing them with essays of known quality based on the degree of conceptual relevance and the amount of relevant content. When put to the test, Results indicated that this technique varied from a human grader as much as two human graders varied from each other. This shows that the Intelligent Essay Assessor performed almost identically to human graders, showing a great deal of support for LSA as a measure of semantic similarity (Foltz, Laham, & Landauer, 1999). The proposed model differs from IEA in that it does not work with a predetermined set of ranked bodies of text, opting instead to learn the user's ranking system and attempting to emulate it.

Theory

LSA represents semantics as a multidimensional vector-space. A given body of text can be represented in this semantic space by a vector. The relatedness between any two such vectors can then be measured based on the angle between them. The greater the angle between two vectors, the greater the difference semantically between the two concepts represented by said vectors.

We believe that humans assign utility values to semantic concepts, and that these values can be measured and utilized to model human interest. The assumption is that the closer two articles are in semantic space, the closer their interest values should be for any given participant. Figure 1 helps to illustrate this idea further, where similar semantic topics have similar interest values for a sample human participant. Thus, the interest value for any new vector drawn in this semantic space may be predicted based on which existing vectors it is closest to.

The experiment described below tests this prediction. The idea is that by comparing the articles having known human interest values with a new article having none yet assigned, we can predict the utility value of the new article. By taking the average interest value of the n most closely related LSA vectors, we can infer a value for the unknown article. Giving more weight to vectors that are more closely related to the unvalued article's vector may increase the prediction's accuracy (this may compensate for cases where the known semantic interest space is sparse, and only a small number of articles have a high relatedness to the unrated article).

Modeling

In the experiment we explore the problem space mentioned in the introduction based on its three defining factors.

Content Size

The experiment inspected the effect content size held over the accuracy of the predictions. When considering news articles, we used the title text versus the full article text to examine this. We wanted to know if the titles' of the articles alone would give enough information for rating predictions. The assumption was that the article titles would give a fair indication of the article's representation in a semantic space (as is the case much of the time for human readers). On the other hand it may be better to base predictions on the full set of data, in this case article content.

Measures

Several measures were used to predict utility values to each user's articles. Averages of the n closest related articles to the one being rated were used, with $n = 3,5,7,9,11,15,25,33,100,299$. Weighted averages and double weighted averages of the same amounts were also calculated in averages of the n closest vectors. The weighted average was used to determine a predicted interest for article a as follows:

$$WA(a) = \frac{\sum_{k=1}^{n} LSA(a,x_n) \times Rating(x_n)}{\sum_{k=1}^{n} LSA(a,x_n)} \tag{1}$$

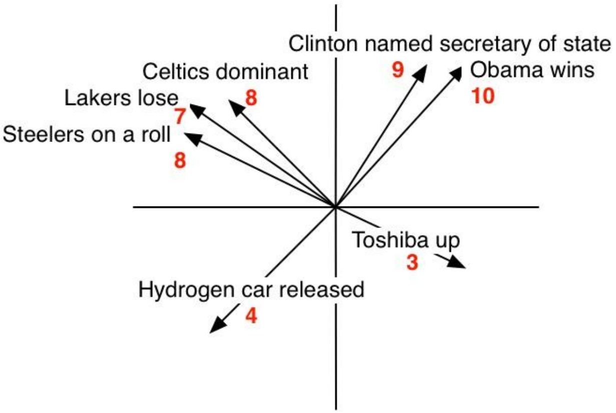


Figure 1: An example of a semantic space displaying article vectors, their titles, and the utility value assigned to them by a fictitious human participant.

where x_k is the nth most closely related article and $LSA(a,x_k)$ is the measure of the relatedness between a and x_k . The Double weighted average is calculated almost the same except that the measure of the relatedness between a and x_k is squared like so:

$$WA(a) = \frac{\sum_{k=1}^{n} LSA(a,x_n)^2 \times Rating(x_n)}{\sum_{k=1}^{n} LSA(a,x_n)^2} \tag{2}$$

Each measure was tested using the LSA vectors for only the article's title as well as the article's title and content. There is no comparison between articles rated by different users. Also, the averages were rounded, so that an integer value was assigned as the predicted rating. For each user, the root mean squared error (RMSE) of the predicted utility values from the user defined utility values was calculated as an indication of overall performance.

Nomothetic Versus Idiographic

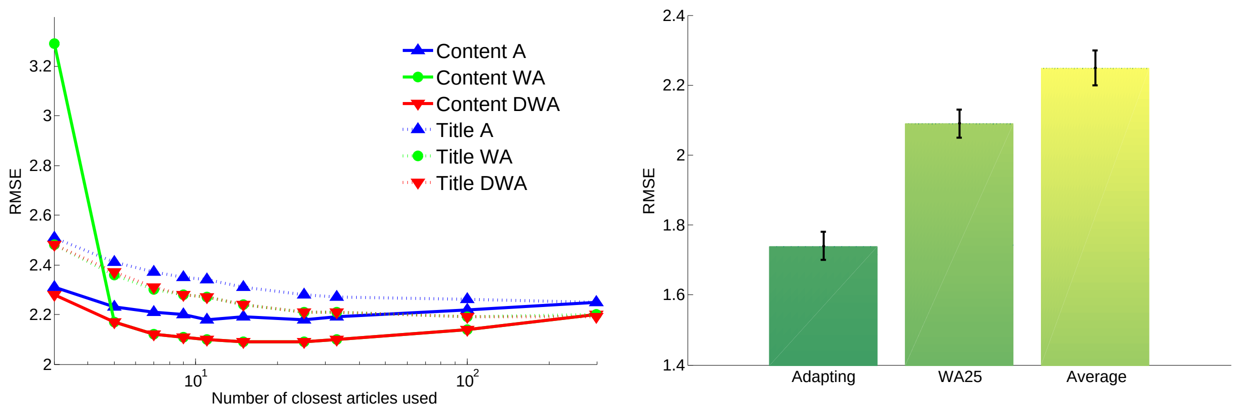
Finally the prediction accuracy between two distinct approaches was measured and compared. The Nomothetic approach simply used one content size and one measure with every user. This static approach was applied to every combination of content size and measure. The idiographic approach involved an Adapting Measure, which tried every combination of content size and measure to predict a given user's utility values, and chose the most accurate on a user by user basis.

Experiment

The primary purpose of this experiment was to measure the accuracy of LSA in predicting a user's interest in regards to news articles based off that user's previous ratings.

Procedure and Design

Participants 200 undergraduate students of RPI participated for course-credit. Twelve of the participants failed to finish the experiment, and their data was subsequently removed from any further analysis.



(a)

(b)

Figure 2: **(a)** Average RMSE values for Average (A), Weighted Average (WA), and Double-Weighted Average (DWA) measures for predicting interest values, using either the article Title, or its Content to create the LSA vector. **(b)** A comparison of the average RMSE values of an Adapting Measure, the nomothetic measure with the lowest RMSE (weighted average of 25 articles), and the Average Interest Heuristic (average of 299 articles).

Design The experiment employed a single-group design with no between-subject variables. 400 news articles were taken from Reuters.com and classified into 20 categories based on their content. The categories used were: Sports, Entertainment, US News, Environment, Health News, Lifestyle-Health, Politics, International, Business News, Deals, Private Equity, Mergers & Acquisitions, Science, Internet, Lifestyle-Technology, Technology, Lifestyle-Travel, Oddly Enough, Lifestyle-Living, and Lifestyle-Autos. Each category contained 20 articles. These articles were pulled from the Reuters RSS feed the week before the experiment was run so as to offer the most recent articles possible to the Participants for grading. The experiment itself was designed as a web application and the Participants were instructed to complete it at home, allowing a greater number of participants to contribute.

Procedure Before the online experiment began, the participants were instructed to provide their name, gender, age, and major. Participants were then instructed to rate 300 articles, chosen randomly from the aforementioned set of 400 articles. Each article was to be rated on a scale of 1 to 10 with 1 signifying indifference and 10 signifying that an article perfectly matches the participants interest. The participants were also told that they did not have to read the articles if they could gauge their interest by the title alone. The experiment appeared in the browser as a list of 10 article titles hyperlinked to their source with 10 radio buttons underneath each title labeled 1-10 to allow the participants to submit their ratings with ease. Once the participants rated 10 articles, they would be able to click a button at the bottom of the screen which would reload the page with 10 new articles. The experiment would not allow the participants to move on to the next page of 10 articles without first rating the 10 that were currently displayed After the 300 articles were rated, the participants were then asked to complete a questionnaire that gave us valuable feedback in regards to the experiment's procedure. Pilot participants were able to finish the experiment in less than one hour.

Results and Analysis

Each measure's performance (i.e. how accurately they predicted the participants' ratings) is displayed as RMSE values in Figure 2 and Table 1.

	Content			**Title**		
n	**A**	**WA**	**DWA**	**A**	**WA**	**DWA**
3	2.31	3.29	2.28	2.51	2.48	2.48
5	2.23	2.17	2.17	2.41	2.36	2.37
7	2.21	2.12	2.12	2.37	2.30	2.31
9	2.20	2.11	2.11	2.35	2.28	2.28
11	2.18	2.10	2.10	2.34	2.27	2.27
15	2.19	2.09	2.09	2.31	2.24	2.24
25	2.18	**2.09**	2.09	2.28	2.21	2.21
33	2.19	2.10	2.10	2.27	2.21	2.21
100	2.22	2.14	2.14	2.26	2.19	2.19
299	**2.25**	2.20	2.20	2.25	2.20	2.19

Table 1: RMSE values from the graph in Figure 2a.

It appears that using the actual content of the article to fill the semantic space is superior to using just the title text. This is most likely due to the greatly increased amount of text used to create the article vectors. Larger bodies of text allow for stronger similarity between the articles' content, and therefore better results. Focusing purely on the content based information, it is evident that there is no benefit to using double weighted averaging, as it offers almost identical results to just using the weighted averages. The best overall measures seen here are the weighted averages of 15 and 25 articles (WA15 and WA25). Given this information we can say that the best nomothetic measures are weighted averages of somewhere between 15 and 25 of the most closely related articles.

Also worth noting is the performance increase from using the average rating of all 299 articles. Using the participant's average interest value as a heuristic for approximating their interest in any one article, results in an average RMSE of 2.25. This Average Interest Heuristic may be used as a performance baseline. The average difference in RMSE values between WA25 and the Average Interest Heuristic is .16. This is a dramatic difference, considering that real-world rating prediction algorithms are competitive to the RMSE values of .001. Consider, for example, the Netflix Prize contest where RMSE improvements in the thousandths place are the difference between being in the top 5 and being in the top 26. (Netflix, 1997).

Although WA25 produces the lowest average RMSE, greater accuracy can be achieved by using an Adapting Measure. By choosing the best measure for each participant, performance is increased. In other words, whereas WA25 may be the best rating predictor for one participant, a simple WA15 may be more appropriate for another. The average RMSE value for using the Adapting Measure is 1.74. A repeated measures ANOVA revealed significant differences between the Adapting Measure (M=1.74, SE=.04), WA25 (M=2.09, SE=.04), and the Average Interest Heuristic (M=2.25, SE=.05), F(2, 557) = 38.272, $p < .001$.

Conclusions

The experiment determined that LSA warrants further study as a model of predicting human interest. Initial results for predicting participants' interests in news articles (using the default LSA corpus and setup) were very positive, resulting in RMSE values as low as 2.09 using a nomothetic method. The idiographic method resulted in significantly better performance still, RMSE = 1.74. A greater degree of article content seems to lead to more informative LSA vectors, and better rating predictions. Lastly, we have narrowed down the list of measures for further examination to Weighted Averaging of 15 to 25 closest articles, disregarding Averaging and Double-Weighted Averaging methods of rating estimation. With further study and experimentation we believe that this impressive level of accuracy can be improved to an even greater precision.

Future experiments will involve rating predictions for more diverse text (e.g. comics, books, scientific papers). Modifications to the current model will be explored, using alternative (more modern) training corpora for LSA, and different modeling techniques.

Acknowledgements

The work was supported, in part, by grant N000140710033 to Wayne Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer. We thank Robert Lindsey for his assistance.

References

Balabanovic, M. (1998). Exploring versus exploiting when learning user models for text representation. User Modeling and User-Adapted Interaction, 8, 71-102.

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. The Journal of Machine Learning Research, 3, 993-1022.

Blei, D., & Lafferty, J. (2006). Correlated topic models. In Advances in neural information processing systems 18.

Foltz, P., Laham, D. & Landauer, T. (1999). Automated Essay Scoring: Applications to Educational Technology. In B. Collis & R. Oliver (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 1999 (pp. 939-944). Chesapeake, VA: AACE.

Golder, S. and Huberman, B. A. 2005 The structure of collaborative tagging systems. Technical report, In-formation Dynamics Lab, HP Labs.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. Proceedings of the National Academy of Sciences, 101, 52285235.

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in semantic representation. Manuscript submitted for publication.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240.

Lemaire, B., Denhiere, G., Bellissens, C., & Jhean-Iarose, S.(2006). A computational model for simulating text comprehension. Behavior Research Methods, 38(4), 628-637.

Lindsey, R., Veksler, V. D., Grintsvayg, A., & Gray, W. D. (2007). Effects of Corpus Selection on Semantic Relatedness. 8th International Conference of Cognitive Modeling, ICCM2007, Ann Arbor, MI.

Lindsey, R., Stipicevic, M., Veksler, V.D., & Gray, W. D. (2008). BLOSSOM: Best path Length On a Semantic Self-Organizing Map. 30th Annual Meeting of the Cognitive Science Society.

Matveeva, I., Levow, G., Farahat, A., & Royer, C. (2005).Term representation with generalized latent semantic analysis. Presented at the 2005 Conference on Recent Advances in Natural Language Processing.

Netflix. (1997-2009). Netflix Prize: View Leaderboard. Retrieved April 8, 2009, from http://www.netflixprize.com/leaderboard.

Netflix. (1997-2009). Netflix Prize: View FAQ. Retrieved April 12, 2009, from http://www.netflixprize.com/faq.

Paul Heymann, Georgia Koutrika, and Hector Garcia Molina, Can Social Bookmarking Improve Web Search?. Resource Shelf, (2008).

Veksler, V. D., Govostes, R. Z., & Gray, W. D. (2008). Defining the Dimensions of the Human Semantic Space. Proceedings of the 30th Annual Meeting of the Cognitive Science Society, CogSci08.

Bayesian model comparison and distinguishability

Julien Diard (julien.diard@upmf-grenoble.fr)
Laboratoire de Psychologie et NeuroCognition CNRS-UPMF
Grenoble, France

Abstract

This paper focuses on Bayesian modeling applied to the experimental methodology. More precisely, we consider Bayesian model comparison and selection, and the *distinguishability* of models, that is, the ability to discriminate between alternative theoretical explanations of experimental data. We argue that this last concept should be central, but is difficult to manipulate with existing model comparison approaches. Therefore, we propose a preliminary extension of the Bayesian model selection method that incorporates model distinguishability, and illustrate it on an example of modeling the planning of arm movements in humans.

Keywords: Bayesian modeling; model selection; distinguishability; arm movement; trajectory planning; human control.

Notation

δ	a single data point
Δ	a set of data points
x, y	coordinates for data points
m_i	a single model (the *i*-th)
M	a set of models
θ	a single parameter value
Θ	a set of parameter values
D	measure of distinguishability

Introduction

In probabilistic modeling, models are usually encoded by a term that describes the probability of an experimental datum δ, given the model M_i: $P(\delta \mid M_i)$.

When the purpose is to select a model out of several alternatives, given some observed data points, the $P(\delta \mid M_i)$ term is usually hierarchically encapsulated in a higer-level model, which relates several models M, several possible parameter values for these models Θ, and several data points Δ:

$$P(M\ \Delta\ \Theta) = P(M)P(\Theta \mid M)P(\Delta \mid \Theta\ M)\ . \tag{1}$$

This leads to a variety of interesting model selection techniques, like the Maximum Likelihood Estimator (MLE), the Maximum A Posteriori estimator (MAP), various least squares based estimators, or algorithms using the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), or, more generally, the Bayesian Model Selection method (BMS). We refer the interested reader to previous reviews of these techniques (Myung & Pitt, 2004; Hélie, 2005).

All of these methods, at their core, aim at selecting a model out of a class of models, in order to maximize the fit measure, or some compound of the fit and model complexity. One of the possible extensions is, instead of selecting one single model, to consider the whole distributions over models $P(M \mid \Delta)$ in order to gain a better understanding of the relation between the best model and the next best models. The issues here are legitimate: how is the best model winning over the rest? Is it only marginally better?

However, some further questions, that are relevant in terms of scientific methodology, cannot easily be treated on the basis of Eq. (1) alone. Indeed, it is a very simple structure, which places at the heart of the analysis the fit of a single datum δ_i to a single model m_j, in the term $P(\delta_i \mid [M = m_j])$.

For instance, a couple of questions, that are crucial for the scientific methodology, are: "are m_i and m_j predicting different results?", and "where should the next experiment investigate in order to clarify whether m_i or m_j is the best model?". In other words, the central issue here is the *distinguishability* of models m_i and m_j (Berthier, Diard, Pronzato, & Walter, 1996) [1], in particular with respect to the space of experimental data. Instead of caring about the particular fit, or lack thereof of a model, the concern is about the relative fits of available models; are models with relatively close fits able to being discriminated, or not? The two above questions could then be translated mathematically by $P(\text{distinguishable} \mid [M_1 = m_1]\ [M_2 = m_2])$ and $P(x_{T+1} \mid [\text{distinguishable} = 1])$ (using an informal notation for the moment).

However, it appears that Eq. (1) is too limited to allow for an easy formulation of the inclusion of the distinguishability of models. The fit and experimental adequacy of a model, in science, is a complex concept; capturing this rich and difficult concept in a single number that would form the basis of an absolute ranking might be a red herring. Indeed, even the composition of the notion of fit and generalization into a single measure has proven a challenging task for a wide variety of modeling approaches, even though the two concepts are related. Therefore, we propose to pursue an alternate route, developing explicit mathematical formulations of the measure of interest, so as to allow their principled manipulation, using Bayesian inference.

Therefore, we propose to extend here the hierarchical model of Eq. (1) so as to incorporate the notion of distinguishability of models. The central component is to augment the model fit term with a model comparison term.

In the remainder of this paper, we develop the theoretical distinguishability analysis Bayesian model, illustrate it on a hypothetical example, and finally apply it to a scientifically relevant area, the modeling of the planning of arm movements in humans.

Bayesian model distinguishability

Let m_1 and m_2 be two models under consideration. Consider a data space where x are inputs, and models m_1 and m_2 predict outputs y_1 and y_2, respectively, according to the term

[1] This is not a self reference, despite the homonymy of the second author.

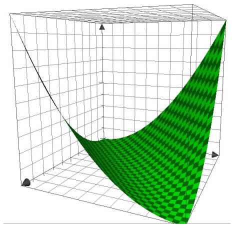


Figure 1: $P(D=1 \mid x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2)$ plotted against $P(M = M_1\ \Theta = \theta_1 \mid x\ y)$ and $P(M = M_2\ \Theta = \theta_2 \mid x\ y)$.

$P(y \mid x\ \theta\ M)$, which is the general likelihood term. Let D be a probabilistic variable, that is binary: it can take values 1 if the models are distinct, and 0 if the models are not distinct. Therefore, we are interested in $P(D \mid x\ y\ \theta_1\ M_1\ \theta_2\ M_2)$: what is the probability that the models are distinguishable given an experimental point x, y and two models?

We define the hierarchical model of model comparison, our alternative to Eq. (1) for the purpose of manipulating distinguishability, as follows:

$$\begin{aligned} & P(D\ x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2) \\ = \quad & P(M_1\ M_2)P(\Theta_1 \mid M_1)P(\Theta_2 \mid M_2) \\ & P(x)P(y \mid x\ M_1\ M_2\ \Theta_1\ \Theta_2) \\ & P(D \mid x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2) \end{aligned}$$

We call the term $P(D \mid x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2)$ the *a posteriori distinguishability*, because it is the distinguishability of M_1 and M_2 with respect to some already observed data point x, y, as opposed to $P(D \mid M_1\ M_2\ \Theta_1\ \Theta_2)$, which is the *a priori distinguishability* of M_1 and M_2, irrespective of any data point. The latter will be obtained via Bayesian inference from the former, as shown below.

a posteriori distinguishability

Model We define $P(D \mid x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2)$ as follows:

$$P(D=1 \mid x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2) = \sqrt{(P(M=M_1\ \Theta=\theta_1 \mid x\ y) - P(M=M_2\ \Theta=\theta_2 \mid x\ y))^2}\ .$$

It is illustrated Fig. 1 [2]

This measure of distance between model recognition given an experimental data has some interesting properties; for instance, the probability that $D=1$ is 0 if and only if the probabilities $P(M=M_1\ \Theta=\theta_1 \mid x\ y)$ and $P(M=M_2\ \Theta=\theta_2 \mid x\ y)$ are equal. The probability that $D=1$ is 1 if and only if either one of $P(M=M_1\ \Theta=\theta_1 \mid x\ y)$ and $P(M=M_2\ \Theta=\theta_2 \mid x\ y)$ is 1 and the other is 0.

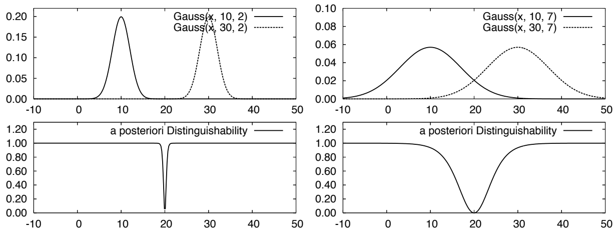


Figure 2: Left: The models are clearly distinguishable. Right: In case of higher standard deviation, the indistinguishability gap between the predictions is wider.

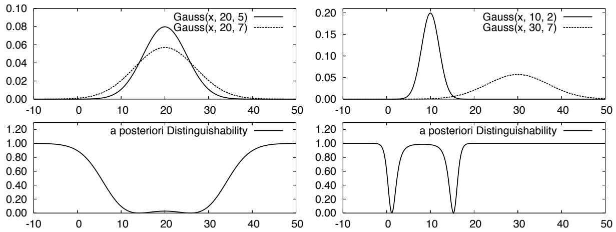


Figure 3: Left: Hardly distinct models (gaussian distributions with same means and close standard deviations). Right: Models with different means and different standard deviations. Note the asymmetry of the bump of distinguishability around the left gaussian model, which corresponds to the cases where it is recognized as the correct model.

Another property of our measure of distinguishability is to be noted: it is symmetric with respect to M_1 and M_2, contrary to previous approaches (Navarro, Pitt, & Myung, 2004).

Finally, since D is a binary variable, we easily turn our distinguishability into a probability measure by defining $P(D = 0 \mid .) = 1 - P(D = 1 \mid .)$: *a posteriori* distinguishability integrates to one. Therefore, the probability distributions over D can fully be described by a single number. By convention, in the remainder of the paper, we only focus on $P(D = 1 \mid .)$.

Example We give a straightforward example, in order to illustrate the a posteriori distinguishability of models.

We define two models m_1 and m_2, of the same family of models, both being defined by Gaussian probability distributions over some arbitrary unit. We set x to some arbitrary value for the moment, in order to have a mono-dimensional data point, over y. Finally, we only consider two possible sets of parameters θ_1 and θ_2. In the following figures, we show how $P(D \mid x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2)$ evolves as a function of y, in different cases for θ_1 and θ_2.

The first example is when the two models clearly predict different outcomes over y. The Gaussian probability distributions for m_1 and m_2 are centered on values μ_1 and μ_2 that are far apart, in the sense that $\mu_1 - \mu_2 \gg \sigma_1$ and $\mu_1 - \mu_2 \gg \sigma_2$. This case is shown Fig. 2 (left). It can be seen that the distinguishability measure is very high over the whole space, except for the data points that fall right between the two mean

[2] Alternative definitions, based on other L_p norms, do exist and have been explored experimentally. For instance, we also used the absolute distance of the difference (the L_1 norm). However, these alternate definitions do not appear to yield dramatically different results. The issue of the *distinguishability of distinguishability measures* is a topic for further research.

predictions μ_1 and μ_2. When the certainty in these predictions gets lower, the indistinguishability gap between the predictions is wider (Fig. 2, right).

Another example concerns the opposite case and is shown Fig. 3 (left): the models are hardly distinct, except if data points fall very far from the predicted means (the flatter of the two models is recognized). Finally, we show Fig. 3 (right) the general case of varying means and standard deviations.

a priori distinguishability

A priori distinguishability is the analysis of models and their prediction, without reference to any actual experimental data point. In this paper, we have chosen to separate the data space in two components, x and y, which have different practical interpretations. x is the input data, that is to say, the part of the experimental point which is decided by the experimenter. On the other hand, y is the output data, that is to say, the measure which is made in experimental condition x.

For instance, if studying free falling objects, x might be weights, and y the time that it takes for an object of weight x to fall from the top of the tower of Pisa. When studying human memory, x might be the time of presentation of a stimulus to a participant, and y the number of features of that stimulus which are correctly recalled by the participant.

Having these two components in the data space opens two variants for *a priori* distinguishability. Firstly, it can be the distinguishability of models M_1 and M_2 for a given experimental condition x, without knowing y:

$$P(D \mid x\, M_1\, M_2\, \Theta_1\, \Theta_2)\ ,$$

which we refer to as the *a priori* distinguishability proper. It can also be the distinguishability of models M_1 and M_2 for all experimental conditions x and possible outcomes y:

$$P(D \mid M_1\, M_2\, \Theta_1\, \Theta_2)\ ,$$

which we refer to as the *overall a priori* distinguishability.

Both can be obtained from a posteriori distinguishability by Bayesian inference from the hierarchical model of model comparison $P(D\ x\ y\ M_1\ M_2\ \Theta_1\ \Theta_2)$. Indeed, assuming uniform probability distributions over discrete x and y variables [3]:

$$\begin{aligned} P(D \mid x\, M_1\, M_2\, \Theta_1\, \Theta_2) &\propto \sum_{y} P(D \mid x\, y\, M_1\, M_2\, \Theta_1\, \Theta_2)\ , \\ P(D \mid M_1\, M_2\, \Theta_1\, \Theta_2) &\propto \sum_{x,y} P(D \mid x\, y\, M_1\, M_2\, \Theta_1\, \Theta_2)\ . \end{aligned}$$

Fig. 4 shows the *a priori* distinguishability of two models that are based on Gaussian probability distributions, with means that are linear in x and standard deviations that are constant, and equal between the two models:

$$\begin{aligned} P(y \mid x\, M = M_1) &= \mathbf{G}_{\mu=\mathbf{a_1 x + b_1}, \sigma=\mathbf{k}}(\mathbf{y}) \\ P(y \mid x\, M = M_2) &= \mathbf{G}_{\mu=\mathbf{a_2 x + b_2}, \sigma=\mathbf{k}}(\mathbf{y}) \end{aligned}$$

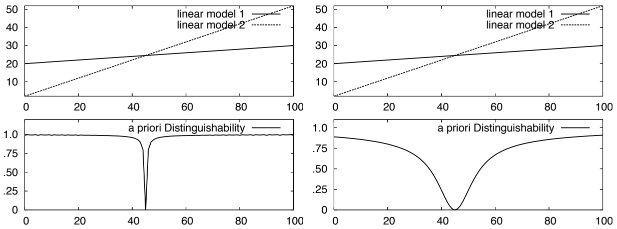


Figure 4: A priori distinguishability. Left: each model is linear with some normal noise of small standard deviation ($\sigma = 1.0$). As a result, models are easily distinguished in most of the space. Right: the noise is increased around the linear predictions ($\sigma = 5.0$): in the whole middle portion of the space, models are hardly distinguishable.

The left panel of Fig. 4 shows the case where the standard deviation k is small for the two models, so that they are highly distinguishable for almost all input data x, except where the means get close, because the linear functions $a_1 x + b_1$ and $a_2 x + b_2$ cross. The right panel shows a similar case, where the standard deviations $\sigma = k$ are larger, so that the region where models are less distinguishable is larger.

The computation of the overall *a priori* distinguishability is not detailed here, but it is trivial that it yields a higher distinguishability for the two models on the left panel than for the two models of the right panel of Fig. 4.

Full scale example: human arm control and planning strategies

Having illustrated the distinguishability model on a few mono-dimensional examples in previous sections, we now turn to a more complex scenario. We study here the planning and execution of movements for a two degree-of-freedom arm.

Human arm geometric model and notation

We consider a simple model of the right human arm, using two segments of same unitary length and two joints, α_1 the shoulder angle, and α_2 the elbow angle. This arm is constrained to move in the horizontal plane, and its endpoint (wrist) position is described by its x, y coordinates in this plane.

We only consider a limited range for possible arm configurations, that include biologically relevant positions: α_1, the shoulder angle, ranges from $-\pi/6$ (arm extended behind the subject) to $5\pi/6$ (arm crossing, in front of the chest). The elbow goes from $\alpha_2 = 0$ when the arm is outstretched, to a maximum value which is function of the shoulder position: when the arm is away from the chest, we assume the elbow can bend totally ($\alpha_2 = \pi$), while when the arm is close to the chest, this restricts the elbow angle to decrease linearly with α_1, so that when α_1 is maximum, α_2 only goes up to $\pi/2$.

The shoulder position is set at the origin $(x, y) = (0, 0)$. The set of possible angular joint configurations defines a

[3] An interesting case would be to consider when y is assumed to be distributed according to the average prediction given by all considered models: $P(y) = \sum_{x,M,\Theta} P(y \mid x\ M\ \Theta)$. We will not develop this further here.

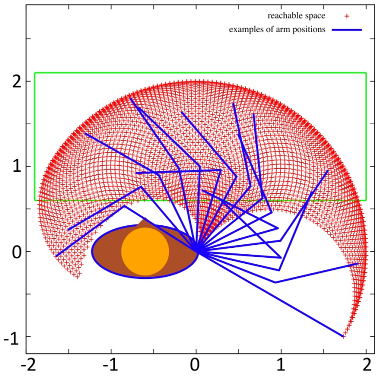


Figure 5: Overhead view of the horizontal plane, the reachable workspace (red or grey crosses), and some examples of possible arm configurations (blue or solid dark segments). The green square (light grey) delimits the integration space for computing the a priori distinguishability (see main text).

workspace of possible reachable positions for the endpoint, which is shown Figure 5. Each of these endpoint positions can be described either in the joint space by a pair of angular coordinates α_1, α_2, or in a Cartesian reference frame by the pair of x, y endpoint coordinates in the workspace.

Models of trajectory formation: interpolation in the joint space or in the workspace?

Movements in this two dimensional space are defined between a start position S and an end position E. Trajectories between these points are assumed to take a unitary interval of time; in other words, each trajectory is indexed by a time variable τ that goes between 0 and 1, with $\alpha_1(0), \alpha_2(0)$ (or $x(0), y(0)$) being the start position S and $\alpha_1(1), \alpha_2(1)$ (or $x(1), y(1)$) being the end position E.

There are two main hypotheses concerning the planning of movements in this context: movements might be planned in the articulatory or joint space (intrinsic planning), or they might be planned in the Cartesian workspace (extrinsic planning) (Palluel-Germain, Boy, Orliaguet, & Coello, 2006). We further assume, for these two alternatives, that the planning process is a simple linear interpolation (Hollerbach & Atkeson, 1987)

Bayesian models M_{int} and M_{ext}

Here, we define the two probabilistic models we consider.

The first model, M_{int} assumes that movements are planned in the intrinsic reference frame. In other words, given start joint angular values $S = (\alpha_1(0), \alpha_2(0))$ and end joint angular values $E = (\alpha_1(1), \alpha_2(1))$, the trajectory to be followed is chosen so that, for all time index $\tau \in [0,1]$, the joint values $\alpha_1(\tau), \alpha_2(\tau)$ are interpolated linearly between the start and end positions.

The start and end positions of movements constitute the x "input" experimental condition of our data space. The chosen and planned trajectory is the output of this experimental point, what the model is predicting; in order to simplify the computational analysis, we choose to summarize the whole planned trajectory by a single point along this trajectory, the one at time $\tau = 1/2$. Furthermore, we assume this point is observed in the Cartesian space x, y. In other words, the "output" data point, y, is the endpoint position $x_{int}(1,2), y_{int}(1/2)$ reached at time $\tau = 1/2$ along the trajectory planned in intrinsic coordinates. Around this predicted position, we will assume some noise, normally distributed, using a two-dimensional gaussian probability distribution with mean μ_{int} and diagonal covariance matrix S:

$$\mu_{int} = \left[\begin{array}{c} x(1/2) \\ y(1/2) \end{array} \right], \quad S = \left[\begin{array}{cc} \sigma & 0 \\ 0 & \sigma \end{array} \right].$$

Finally, M_{int} has no internal parameter Θ_{int}, which simplifies the notation.

We can now make M_{int} formal in the Bayesian programming notation:

$$\begin{aligned} & P(y \mid x\, M = M_{int}) \\ & \quad = \quad P(x\, y \mid \alpha_1(0)\, \alpha_2(0)\ \ \alpha_1(1)\, \alpha_2(1)\, M = M_{int}) \\ & \quad = \quad \mathbf{G}_{\mu,S}(x,y) \end{aligned}$$

The second model, M_{ext}, on the other hand, assumes that movements are planned in the extrinsic reference frame, that is to say, directly the Cartesian workspace. In other words, given start joint angular values $S = (\alpha_1(0), \alpha_2(0))$ and end joint angular values $E = (\alpha_1(1), \alpha_2(1))$, these are first converted into Cartesian start and end positions $x(0), y(0)$, $x(1), y(1)$. Then the straight segment, in the workspace, that connects these two points is the predicted trajectory. Trivially, the predicted middle point at time $\tau = 1/2$ is the geometric middle of the segment (assuming a symmetric velocity profile).

As previously, we assume some normally distributed noise around the middle of the segment $x_{ext}(1/2), y_{ext}(1/2)$:

$$\begin{aligned} & P(y \mid x\, M = M_{ext}) \\ & \quad = \quad P(x\, y \mid \alpha_1(0)\, \alpha_2(0)\ \ \alpha_1(1)\, \alpha_2(1)\, M = M_{ext}) \\ & \quad = \quad \mathbf{G}_{\mu_{ext},S}(x,y) \end{aligned}$$

We show Fig. 6 some examples of trajectories predicted by the intrinsic planning model M_{int} and the extrinsic planning model M_{ext}, and the predicted points for the middles of these trajectories, at $\tau = 1/2$. A special case can be seen where the trajectories are superposed: this is when the line that passes by the start point S and end point E also passes through the shoulder position $(0,0)$. As previously demonstrated, in this case of radial movements, the predicted trajectories are straight segments both in the intrinsic and extrinsic models (Hollerbach & Atkeson, 1987).

Distinguishability of intrinsic and extrinsic interpolation models

Having defined the two intrinsic and extrinsic models in the Bayesian framework, by the terms $P(y \mid x\ M = M_{int})$ and

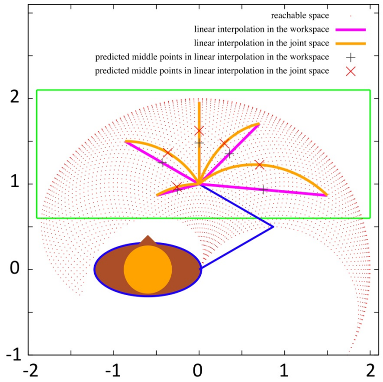


Figure 6: Some trajectories from the same starting position S: $\alpha_1(0) = \pi/6, \alpha_2(0) = 4\pi/6$, predicted by the intrinsic planning model M_{int} (curved trajectories, in orange or light grey) and the extrinsic planning model M_{ext} (straight segments, in purple or dark grey).

$P(y \mid x\ M = M_{ext})$, we can then encapsulate them in the Bayesian metamodel of distinguishability. This allows to compute, for each possible movement to be performed in the workspace, the *a priori* distinguishability between M_{int} and M_{ext}.

More precisely, we restrict the considered movements to those that can be performed by both strategies. Indeed, because the reachable space is not convex (see Fig. 5 or Fig. 6) some movements do have solutions in the intrinsic model, but not in the extrinsic model. In other words, for some pairs of start and end positions, the segment between them lies outside of the reachable space. For instance, this is the case for trajectories with the arm fully outstretched at the starting position. For this reason, we restrict our analysis for a convex subregion of the reachable space (the green rectangle of Fig. 5), and only compute the distinguishability of models for movements where both the start and end positions are inside it.

For a given movement, defined by a start position $S = (\alpha_1(0), \alpha_2(0))$ and an end position $E = (\alpha_1(1), \alpha_2(1))$, we compute the *a priori* distinguishability of models M_{int} and M_{ext}, by integrating over all possible data points. Here again, we only consider possible data points that fall inside the green rectangle of Fig. 5.

Therefore, we obtain, for all possible movements, the probability values $P(D = 1 \mid \alpha_1(0)\ \alpha_2(0)\ \alpha_1(1)\ \alpha_2(1)\ M_1 = M_{int}\ M_2 = M_{ext})$.

Result analysis However, since all possible movements define a four dimensional space, this distinguishability measure cannot easily be visualized and interpreted as is. Some selections and projections to lower dimensional spaces is required, for the distinguishability measure to be plotted. We will firstly present results for a given start position (for all possible end positions), and secondly, aggregated results for all possible pairs of start and end positions.

We further define three projections, to analyse the results.

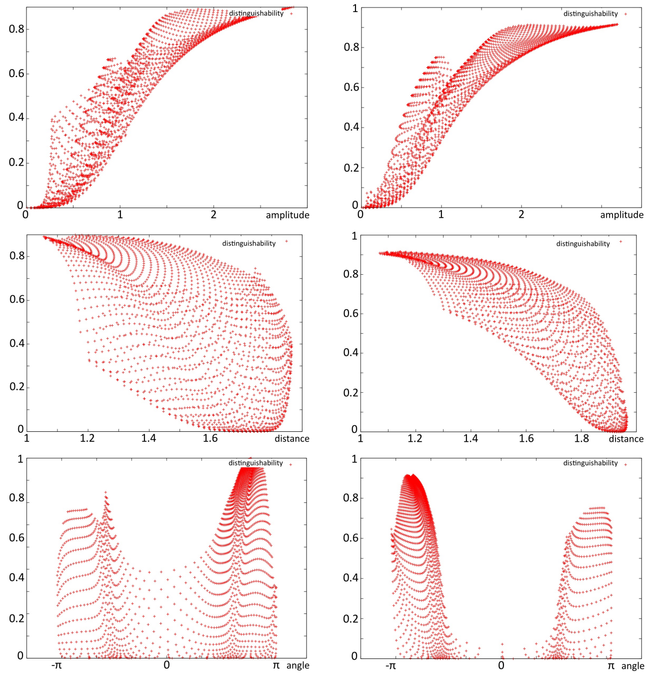


Figure 7: Top row: distinguishability of models plotted against the amplitude of a movement. Middle row: distinguishability of models plotted against the distance of a movement. Bottom row: distinguishability of models plotted against the angle difference with respect to radial lines. Left column: distinguishability of models for all possible movements starting from $S = (\pi/6, \pi/3)$. Right column: distinguishability of models for all possible movements starting from $S = (5\pi/8, \pi/6)$.

We will group movements according to their amplitude, their distance to the shoulder, and their angle difference with respect to radial lines.

The amplitude of a given movement from start position S to end position E is simply defined as the Cartesian distance between S and E in the workspace.

The distance of a given movement from start position S to end position E we define as the distance to origin of the point at $\tau = 1/2$ predicted by M_{ext}. In other words, we consider the distance between the shoulder position and the middle of the segment between S and E in the workspace: some movements are performed very near the body, some movements are performed near the outside limits of the workspace.

Finally, the angle difference with respect to radial lines, for a given movement from start position S to end position E, we define as the angle difference between the segment SE and the segment from shoulder position $(0,0)$ to S. This measure allows to see whether a given movement is purely radial (going straight away from or to the shoulder), or if it is a circular movement (tangent to some circle centered on the shoulder).

We show Fig. 7 the distinguishability analysis for three different starting positions. Fig. 8 shows the aggregate results for all possible starting positions.

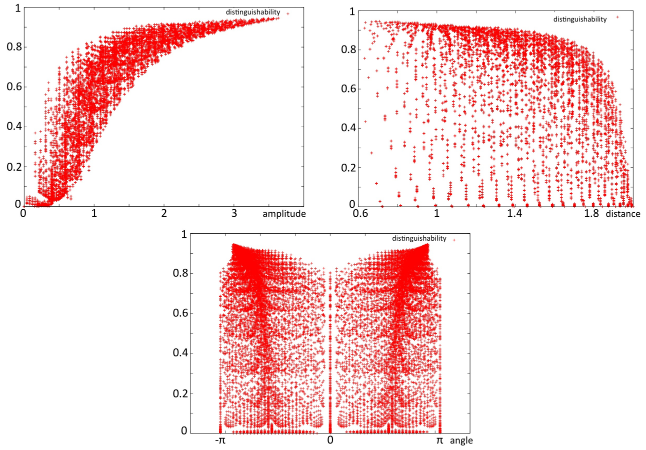


Figure 8: Distinguishability of M_{int} and M_{ext} for all possible movements. Top left plot: distinguishability plotted against the amplitude of movements. Top right plot: distinguishability plotted against the distance of movements. Bottom plot: distinguishability plotted against the angle difference with respect to radial lines.

Result interpretation From these results, some conclusions can be drawn.

The most prominent feature is that the models M_{int} and M_{ext} appear to be most distinguishable when movements are large. Indeed, the distinguishability of models is high for movements of large amplitude, and gets to 0 when the movements are very small. This is confirmed easily by intuition: for large movements, the geometry of the arm has the most impact on the curve predicted by M_{int}. In other words, we see here the effect of the direct kinematic transform.

A second feature is that, for all possible movements, the distinguishability of models does not seem to be dependent of the distance of the performed movement. However, an exception is to be noted: when movements are performed near the outer boundary of the reachable space, the models become hardly distinguishable: their distinguishability dips to 0. This is a confirmation of a fact that was already demonstrated mathematically (Hollerbach & Atkeson, 1987). This was a very important finding, as it allowed to cast doubt on the discrimination power of a previous experiment, where participants had to perform movements bringing them to that outer boundary (Soechting & Lacquaniti, 1981).

A final feature we wish to analyze concerns the angle of movements with respect to radial lines. Contrary to the previous case, this finding contradicts, or rather refines, previous mathematical developments. Indeed, it was shown previously that purely radial movements render the intrinsic and extrinsic planning models not distinguishable (Hollerbach & Atkeson, 1987). Indeed, in this case, both models predict that the trajectories performed are straight (radial) segments. We also confirmed this in one of the example trajectories shown Fig. 6. However, this indistinguishability is purely spatial: when considering the time profile of trajectories, they become distinguishable. This is shown by the bottom plot of Fig. 8: while it is true that radial movements entail, overall, a slightly lower distinguishability of models, there are radial movements where M_{int} and M_{ext} are still distinguishable. This can also be demonstrated by isolating these trajectories, and analyzing them. And indeed, even for radial movements, the positions predicted at time $\tau = 1/2$ are different for the two models. This was also shown on the example radial trajectory of Fig. 6.

Conclusion

In this paper, we developed an original Bayesian metamodel that integrates the notion of distinguishability of models. It allows to manipulate this concept using Bayesian inference, to compute a posteriori distinguishability of given models, but also their a priori and overall a priori distinguishability. We illustrated our model on an example about the planning of arm movements in humans, and showed how it could be used to analyse the space of all possible experimental points. For instance, it was shown that radial movements are indistinguishable spatially, are distinguishable in the temporal domain, and finally, that movements of large amplitude could be used to better discriminate between the alternative models of intrinsic and extrinsic planning.

Further theoretical developments include using the distinguishability metamodel to draw experimental conditions, given that we look for discriminating power, using Bayesian inference to compute a term of the form $P(x \mid M_1\ \theta_1\ M_2\ \theta_2\ D = 1)$.

Acknowledgment

This work has been supported by the BACS European project (FP6-IST-027140).

References

Berthier, F., Diard, J.-P., Pronzato, L., & Walter, E. (1996). Identifiability and distinguishability concepts in electrochemistry. *Automatica*, *32*(7), 973–984.

Hélie, S. (2005). An introduction to model selection: Tools and algorithms. *Tutorials in Quantitative Methods for Psychology*, *1*(1), 1–10.

Hollerbach, J., & Atkeson, C. G. (1987). Deducing planning variables from experimental arm trajectories: Pitfalls and possibilities. *Biological Cybernetics*, *56*, 279–292.

Myung, I. J., & Pitt, M. A. (2004). Model comparison methods. *Methods in Enzymology*, *383*, 351–366.

Navarro, D. J., Pitt, M. A., & Myung, I. J. (2004). Assessing the distinguishability of models and the informativeness of data. *Cognitive Psychology*, *49*, 47-84.

Palluel-Germain, R., Boy, F., Orliaguet, J.-P., & Coello, Y. (2006). Influence of visual constraints in the trajectory formation of grasping movements. *Neuroscience Letters*(401), 97–102.

Soechting, J., & Lacquaniti, F. (1981). Invariant characteristics of a pointing movement in man. *Journal of Neuroscience*, *1*, 710–720.

Biomimetic Bayesian models of navigation: How are environment geometry-based and landmark-based strategies articulated in humans?

Julien Diard (Julien.Diard@upmf-grenoble.fr)
Laboratoire de Psychologie et NeuroCognition CNRS-UPMF
Grenoble, France

Panagiota Panagiotaki and Alain Berthoz
Laboratoire de la Physiologie de la Perception et de l'Action, Collège de France–CNRS,
Paris France

Abstract

We propose a computational model of human navigation, which encompasses both geometry-based and landmark-based navigation strategies. This model is based on a study of human cognitive strategies during a path memorization task in a Virtual Reality (VR) experiment. Participants were asked to memorize predefined paths in a large-scale virtual city (COSMOpoliS©). Our computational model qualitatively reproduces the results of this experiment. This model uses the Bayesian formalism, and focuses on the interplay between the elementary cognitive strategies hypothesized above. It offers an original interpretation of the way these strategies might be articulated, departing from the classical hierarchical structure. This novel view might be fruitful for robotic models from a biomimetic perspective, where managing representations of large-scale and complex environments is still a challenge.

Keywords: Bayesian modeling; human navigation; navigation strategies; landmark-based navigation; path integration.

We discuss here the results of an experiment, in which we have explored the existence of elementary cognitive strategies used for spatial encoding in humans. We have found that, while navigation mainly relies on landmark recognition and encoding for path memorization, the sudden disappearance of these brings in light a back-up mechanism strategy enabling the participants to navigate, although with less accuracy, using geometrical cues alone. These are the first evidence of equivalent components between humans and animals in this context.

In models of navigation, these observations of "back-up" mechanisms usually lead to modeling independent subsystems of navigation, and portraying them as hierarchically articulated. We believe this view of independent subsystems being hierarchically articulated to be simplistic, as it merely pushes back the problem of understanding how different sources of information are integrated in the central nervous system.

In this paper, we propose a probabilistic model that tackles this problem in an original manner. We develop a model of navigation, which, although composed of a single component, can mimic both landmark-based and geometry-based navigation strategies. Bayesian inference is the principle, which enables this single representation of the environment to give rise to several navigation strategies. The overall behavior of our model is dictated by the availability of sensory cues. When there are no uncertainties about the sensed landmarks, our model performs as landmark based navigation. On the other hand, when landmarks are not sensed, the model performs as geometry based navigation. In the following, the term "navigator" will refer to a simulated, imaginary participant that would navigate in our virtual city according to our mathematical model.

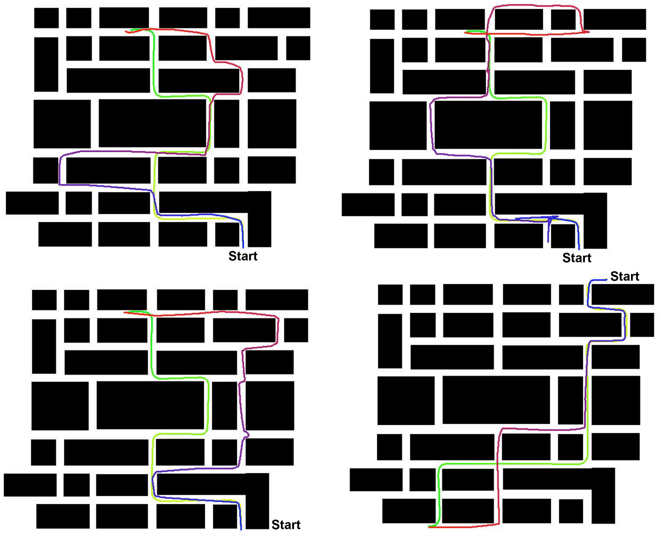


Figure 1: Top-view of the virtual city and archetypal errors observed in the condition where landmarks are removed between memorization and reproduction. In light gray (green), the learned paths. In black (blue to red), the reproduced paths by the participant. For example, in the top-right panel, note how the central building was passed from the right in the learned path, and from the left in the participant's reproduction.

The model qualitatively reproduces patterns of errors we observed in the COSMOpoliS© experiment. In this experiment, humans participants were immersed in a VR city using a VR helmet. They could navigate using a joystick for forward translations, and turn their body in the real world for virtual rotations (with a magnetic tracker set on the VR helmet). Participants were presented a movie of a trajectory, and were asked to memorize it (memorization phase). After seeing the movie twice, they were set in the starting end of the path and asked to reach its end, actively (reproduction phase). Landmarks (posters on walls, lampposts, etc.) were disposed in the city. Experimental conditions were defined by the availability of the landmarks in both the memorization and reproduction phase. We focus here in two conditions: in the Landmark condition, landmarks were available both in memorization and reproduction. In the Probe Trial condition, landmarks were removed between memorization and reproduction.

The data showed that all participants were able to successfully reach a goal in a virtual city in the Landmark condition. Data also showed that the goal was also reached when landmarks were removed between memorization and reproduction (Probe Trial). However, in that case, patterns of error could be observed quite frequently in the paths that participants made in order to reach the goal (see Fig. 1). Participants quite commonly reached the goal using a variant of the memorized path, passing buildings from the wrong side, for instance. Surprisingly, very few participants were actually conscious of these discrepancies.

The rest of this paper is structured as follows. Firstly, we briefly review the related work on hierarchical modeling of human, animal or robotic navigation. We then present the Bayesian model we developed in order to have our simulated navigator reproduce this pattern of error: we first introduce our simplifying assumptions, then define the model and describe its simulation. Finally, we discuss the interpretation of our model as was defined, as well as of the relevance of our simplifying assumptions. The paper concludes on a discussion on the way our assumptions could be relaxed, yielding perspectives on the future work.

Related modeling works

Both life sciences and robotics have made the modeling of navigation capabilities of autonomous entities a crucial point of research, and a wide variety of models already exists. We focus here on hierarchical models of navigation.

In the domain of mobile robotics, modeling the environment that a robot has to face, usually in the form of a map, is a crucial problem. The most promising approaches rely on the probability calculus, thanks to its capacity for handling incomplete models and uncertain information. These approaches include – but are far from limited to – Kalman Filters, Markov Localization models, (Partially and Fully) Observable Markov Decision Processes (POMDP and MDP), and Hidden Markov Models (see (Diard, Bessière, & Mazer, 2003) for a general introduction).

In this domain of probabilistic modeling for robotics, hierarchical solutions are currently flourishing. The more active domain in this regard is decision theoretic planning: one can find variants of MDPs that select automatically the partition of the statespace (see for instance (Hauskrecht, Meuleau, Kaelbling, Dean, & Boutilier, 1998)). Another class of approaches that rely on deterministic notions is based on the extraction of a graph from a probabilistic model, like for example a Markov Localization model (Thrun, 1998), or a MDP (Lane & Kaelbling, 2002).

However, the main philosophy used by these hierarchical approaches is to try to extract, from a very complex but intractable model, a hierarchy of smaller models Automatically selecting the right decomposition is of course a very difficult problem. Moreover, even obtaining in the first place the initial, complex model, is still a difficult challenge in the general case.

From a bio-mimetic perspective, it appears obvious that a global, complex, large-scale model is not the starting point of the acquisition of representations of space (B. J. Kuipers, 2000). Therefore, some robotic approaches, integrating insights from biology, rather start from low-level behaviors and representations, and then try to combine them so as to obtain large-scale representations (Diard & Bessière, 2008; B. J. Kuipers, 2000; B. Kuipers, Modayil, Beeson, MacMahon, & Savelli, 2004; Victorino & Rives, 2004). Indeed, the study of navigation capabilities in life sciences assumes right from the start of its analysis that navigation is hierarchical in nature, as can be easily assessed experimentally (Voicu, 2003).

The hierarchies of models proposed in some of these works (Trullier, Wiener, Berthoz, & Meyer, 1997; Franz & Mallot, 2000; B. J. Kuipers, 2000; B. Kuipers et al., 2004) have several aspects: they are hierarchies of increasing navigation skills, but also of increasing scale of the represented environment, of increasing time scale of the associated movements, and of increasing complexity of representations. This last aspect means that global topologic representations, which are simple, come at a lower level than global metric representations, which are arguably more complex to build and manipulate. This ordering stems from the general observation that animals that are able to use shortcuts and detours between two arbitrary encoded places (skills that require global metric models) are rather complex animals, like mammalians. These skills seem to be mostly absent from simpler animals, like invertebrates.

Works by Jacobs and Schenk go a step further, by proposing the Parallel Map Theory (PMT) (Jacobs & Schenk, 2003), in which a study of phylogenetically equivalent neuroanatomical areas across different species helps hypothesize common hierarchies of representations of space. In other words, they propose a model of how the different layers in the above theories might be implemented in the central nervous system.

Finally, Wang and Spelke (Wang & Spelke, 2002) assume three subsystems, two of which being a path integration (PI) and a view dependent place recognition system. These two, in the context of the current paper, can be seen as analogous of what we will denote as the environment geometry-based and landmark-based navigation systems, respectively.

However, to the best of our knowledge, the question of how different subsystems of a hierarchy of models can exchange information in a principled manner is still an open issue. In other words, most existing models of animal navigation describe hierarchies by identifying individual layers, but do not tackle the problem of how these layers are linked. They usually assume that a supervisor subsystem is responsible for selecting the interaction between individual components, but rarely describe the way this supervisor could work, or even discuss its plausibility (*e.g.* the reference frame selection subsystem of Redish & Touretzky (Redish & Touretzky, 1997)).

Our model precisely proposes an original articulation between a representation of a memorized path and resulting strategies of navigation.

Model

In this section, we develop a Bayesian model, which qualitatively reproduces the observed patterns of errors (see Fig. 1). Being preliminary, our model requires several simplifying assumptions that we describe first. We then describe how, given these assumptions, this single model is defined and used in order to simulate the navigator in the virtual city in both the Landmark and Probe Trial conditions. We finally discuss the similarity between the simulation and experimental data.

Simplifying assumptions

Our model requires two major assumptions: the first concerns the identification of orientations by the navigator; the second concerns the identification of landmarks.

Firstly, we assume that the navigator uses a global reference frame for orientations. This means that an estimate of the navigator's bearing with respect to some origin is available at every moment. Given this estimate, the navigator knows which direction it is currently going. This helps it classify elementary displacements according to the direction followed. This implies a separation between the estimation of orientations and the estimation of positions. Neuroanatomically, such a separation appears to be plausible: estimations of orientations might be grounded in head-direction cells (Stackman & Taube, 1997; Taube, 1998); estimations of positions might be grounded in place cells (Redish & Touretzky, 1997). However, to the best of our knowledge, such a separation is rarely present in robotic models, where, usually, the pose x, y, θ of the robot is considered, with similar mathematical treatment for position x, y and orientation θ.

In order to be used, this orientation reference frame does not need global sensory cues. Indeed, instead of being based on some external cue, the origin could be based on the starting orientation of the navigator (Berthoz et al., 1999).

We further assume that this global reference in orientation does not drift during the navigation of the path. Indeed, in COSMOpoliS© and in our simulation, all angles between streets are 90° angles, thus reducing risks of disorientation (drifting of the orientation reference frame). With these assumptions, in our model, we only need four possible orientations for the global reference frame. In the following, we denote "up" the starting direction, "down" the opposed direction, and "left" and "right" the two remaining directions.

Secondly, we assume that landmarks in the virtual city are all unique and easily recognizable. We assume they are placed at the intersections or decision points, as it has been shown that the relevance of a landmark to solving navigation tasks is explicitly encoded in the central nervous system (Janzen & Turennout, 2004). We further assume that landmarks can be used to recognize all intersections in the city without errors. These assumptions allow the model to include certainties (probabilities of 1) about the landmark and their recognition, when they are available.

Model definition

We now define a two-variable model.

The first variable, denoted L_t, is the location at time t, *i.e.* the intersection the navigator is in, as defined by the landmark appearing at this intersection. Assuming n different landmarks and intersections in the virtual city, $l_1, l_2, \ldots, l_n$, we thus define: $L_t = \{l_1, l_2, \ldots, l_n\}$. The second variable, denoted A, is the direction that should be followed at intersection l_t. According to our assumptions concerning the global orientation reference frame, we define A by $A = \{\text{up}, \text{left}, \text{down}, \text{right}\}$.

We thus define the joint distribution:

$$P(A, L_t) = P(L_t) P(A \mid L_t),$$

by applying Bayes rule. The first term, $P(L_t)$, is the likelihood to be in some intersection. We define this term by a uniform probability distribution: $P(L_t = l_t) = 1/n$. The second term, $P(A \mid L_t)$, represents probability distributions over directions to follow, given the identity of the intersection the navigator currently is at. We define this term by Conditional Probability Tables (CPT). We assume the navigator identifies these CPTs during the path memorization phase of the experiment. In other words, during path presentation, the navigator counts the number of times it went "up", "down", "left" and "right", and builds the CPTs that reflect these frequencies. The CPTs follow Laplace succession law distributions, which are similar to histograms, except that probabilities for unobserved cases are never zero [1].

There is one such CPT for each landmark seen along the memorized path. As we have assumed all landmarks to be unique, and assuming that the paths never pass twice in the same intersection (which is the case in the COSMOpoliS© experiment), these learned distributions are all of the same type: the probability is close to 1 for the actual direction followed along the path, and close to 0 for the three directions not followed.

[1] Laplace succession law probability distributions merge a prior distribution with observed data. The formula is $P(A = i \mid L_t = l_t) = \frac{n_i + w}{N + kw}$, with n_i the number of times a particular case i has been observed, N the total number of observations, k the size of the domain of the variable, and w a parameter which tunes the speed at which the initial uniform distribution is modified as the observations are collected. A Laplace succession law converges toward a histogram when the number of observations N is large. Assessing a biologically plausible weight w is an open question (out of the scope of this paper and experiment).

Model usage

Having learned the CPTs during the path presentations, the model is now fully defined. The joint distribution $P(A, L_t)$ is available to the navigator, and we describe here how it can be used to drive the navigator during the path reproduction, in the Landmark and Probe Trial conditions.

In both cases, the navigator must decide, at each intersection, the direction to follow in order to accurately reproduce the path it memorized. In the Landmark condition, intersections can be identified during the learning phase as well as during the reproduction phase. Therefore, when arriving at an intersection, the value l_t of the current intersection is available, and can be used to select the relevant probability distribution over actions $P(A \mid L_t = l_t)$. Once this distribution is selected, choosing the action with the highest probability value will lead the navigator along the memorized path, without errors.

Alternatively, the navigator can draw at random according to the memorized probability distribution. In this case, errors in the reproduction could occur, their frequency depending on the parameter w chosen for learning CPTs.

In the Probe Trial condition, landmarks are not available in the city anymore in the reproduction. Therefore, when arriving at an intersection, it is not possible for the navigator to know the value of l_t. However, using Bayesian inference, $P(A)$ can be computed: it is the probability distribution over actions to take at each intersection, without knowing the intersection identity. The computation is as follows:

$$P(A) = \sum_{L_t} P(A, L_t) \propto \sum_{L_t} P(A \mid L_t).$$

This computation yields the best estimate available to the navigator in order to choose what direction to go at each intersection during the reproduction. Drawing at random according to $P(A)$ allows the navigator to reproduce the memorized path in the best manner, given the absence of visual cues.

Model simulation

We have simulated the model in an idealized version of the VR city, abstracting ourselves from issues related to the small scale of the VR city COSMOpoliS©. We call this ideal, simulated city EQUApoliS. EQUApoliS is a regular, infinite grid of square blocks, with simulated unique landmarks at each intersection.

We defined a path to be learned and reproduced (narrow black path in Fig. 2). When the navigator memorizes this path, it learns by observation CPTs, one for each intersection. One such CPT is shown Fig. 3.

In the Landmark condition, the memorized path is accurately reproduced. Indeed, recall that the CPTs in this condition can be read so as to provide, at each intersection, a probability distribution over actions that clearly encodes the direction that was followed during path memorization. Therefore the navigator is driven along a path which is an exact reproduction of the memorized path.

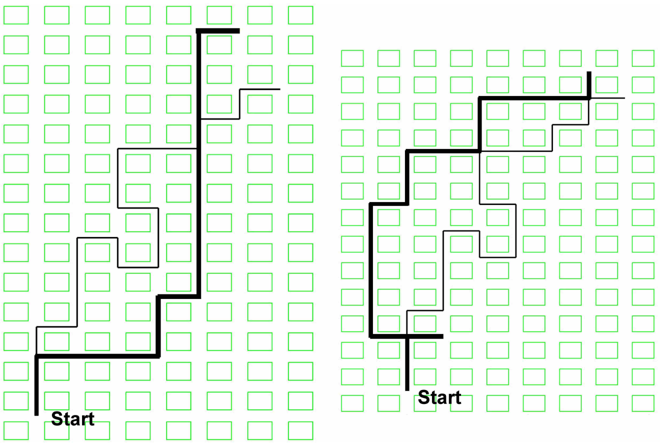


Figure 2: Typical trajectories obtained by the simulated navigator in the Probe Trial condition. In narrow black, the memorized path, in bold black, the simulated reproduced path.

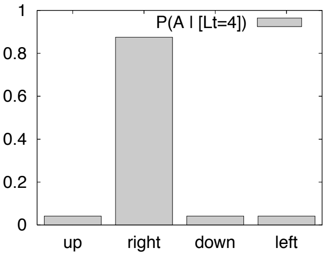


Figure 3: The CPT learned for the term $P(A \mid L_t = 4)$: this is the probability distribution for the fourth intersection of the memorized path (see Fig. 2). At this intersection, the navigator observed twice a movement to the right. This yields a high probability value of going right (0.875, assuming w is set to 0.1).

In the Probe Trial condition, however, landmarks are not available anymore, and $P(A)$ must be computed. In this example, this leads to the probability distribution shown Fig. 4. As the path contains 12 moves in the same direction as the initial orientation (we note this direction "up"), 7 moves to the "right", 1 move "down", and 1 move to the "left", the computation for the term $P(A)$ in the Probe Trial gives the probability distribution shown. At each intersection, we draw at random according to $P(A)$, until 21 displacements have been made: Fig. 2 shows typical trajectories obtained in this manner (bold trajectories).

Discussion

Interpretation of the proposed model

The simulation results illustrate that the proposed computation of the $P(A)$ distribution can be interpreted as a "path integration" component, both in an intuitive sense and in a mathematical sense.

In an intuitive sense, the probability distribution over ac-

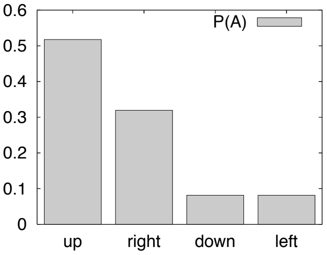


Figure 4: The CPT computed for the term $P(A)$ in the Probe Trial simulation.

tions drives the navigator toward the goal in the correct general direction. Indeed, $P(A)$ as computed can be interpreted as an estimation of the angle from the starting orientation to the goal. For instance, the distribution shown Fig. 4 encodes the knowledge that, to reach the goal, the navigator must mainly go "up" and "right", and it also encodes the relative proportions of these elementary displacements. $P(A)$, seen is this manner, not only encodes an estimate of the bearing of the goal, it also encodes the accuracy or reliability of that estimate, by the spread or uncertainty of the obtained probability distribution.

Let us now recall the mathematical sense of a path integrator. Let $\vec{p}(t)$ be the path, *i.e.* the sequence of elementary displacement vectors at time $t, 0 \leq t \leq T$. The vector $\vec{V}(t)$ representing the global displacement from the initial time 0 to time T is then given by:

$$\vec{V}(t) = \int_0^T \vec{p}(t).$$

In our formulation, time is not continuous, but discretized using events which are the passage at intersections. This explains the use of a discrete summation over intersections L_t instead of a continuous integral over time. Moreover, we assume the elementary displacements are not known deterministically, as in $\vec{p}(t)$, but are encoded using the probability distributions $P(A \mid L_t)$. Therefore, the equation $P(A) \propto \sum_{L_t} P(A \mid L_t)$ can be interpreted as a Bayesian, discrete version of a path integration mechanism.

The simulation shows that our model qualitatively reproduces the patterns of errors made by participants in the Probe Trial. Indeed, in the simulated Probe Trial path reproduction (Fig. 2), we observe that, even though the navigator is driven in the general direction of the goal, the order in which the elementary displacements were performed in the learned path are completely forgotten. This is a direct consequence of the way the probability distribution $P(A)$ is computed. In $\sum_{L_t} P(A \mid L_t)$, the summation can exactly be interpreted as an aggregation of all observed displacements. In other words, the sequencing of displacements, which is present in $P(A \mid L_t)$, is not present anymore in $P(A)$.

The model structure proposes an original hypothesis concerning the interplay between the landmark-based cognitive strategy and the path integration strategy for spatial navigation. Whereas, in the literature, they are commonly pictured as independent mechanisms hierarchically articulated by a main system / back-up system relationship, in our model, there is only one navigation system. When all sensory information are available, this system corresponds to the landmark-based navigation; when some sensory inputs are missing, the same system can operate in a degraded mode, and then exhibits properties of a path integration mechanism.

Relaxing our assumptions: towards experimental predictions and new protocols

We now discuss the relevance of the simplifying assumptions required by our model, which leads us to its possible extensions and the experimental predictions it can provide.

We have assumed, in the model, that all landmarks could be identified with no errors. In a real world navigation scenario, it is of course highly improbable that visual landmarks are never ambiguous. In the COSMOpoliS© experiment, landmarks were unique along the trajectory. However, the study of the way places and intersections are identified is a complete domain of investigation in itself. The goal is to distinguish the intersection identity L_t from the perceived sensory cues at that intersection $P_1, \ldots, P_k$. For instance, landmarks are not the only cues that can be used to identify intersections, as configurations of landmarks could play a role (Mallot & Gillner, 2000), and geometrical configurations of the intersection itself (T-shaped, X-shaped) is probably also encoded (Stankiewicz & Kalia, 2007). In our model, we have assumed that the intersection identity L_t to be readily available; in practice, it could be estimated according to $P(L_t \mid P_1, \ldots, P_k)$. Determining the perceptual components $P_1, \ldots, P_k$ and the structure of this perceptual model is subject of future work.

Another major simplification in our model is the lack of temporal dependency between intersections. Indeed, it is highly probable that pairs $\langle L_t, L_{t+1} \rangle$ of landmarks perceived in sequence, or even higher order sequences $\langle L_t, \ldots, L_{t+m} \rangle$ are used for memorizing paths. Sequences of actions might also serve as large-scale cues for memorizing the paths. This could be incorporated in *m*-order Markov models of the form $P(L_t, \ldots, L_{t+m}, A_t, \ldots, A_{t+m})$. It might be interesting to use future experimental data in order to estimate *m*, *i.e.*, the length of the sequences of sensory and motor cues used for path memorizing.

Finally, we wish to discuss the way we generate simulated paths with the model. Indeed, so far, we have assumed the navigation could use probability distributions over actions, and draw at random, at each intersection, directions to follow. In the current simulation, no memory whatsoever is included in this process. In other words, our simulated navigator would not be able to know if it was "unlucky" in its progress, and was deviating away from the memorized orientation to the goal. However, it appears obvious that human navigators would update their estimation of the orientation to the goal as they progress towards it. Mathematically, it would be straightforward to enrich our model to reproduce such a mechanism. Unfortunately, the current experimental data would not enable us to determine the biological plausibility of any such mathematical development.

Conclusion

We have presented a preliminary model of large-scale human navigation in a virtual city. This model successfully qualitatively reproduces patterns of errors that were observed in human participants. In the Landmark condition, where all visual cues are present, both the participants and the simulated navigator accurately reproduce the learned path. In the Probe Trial condition, where the visual cues needed to recognize the current position are missing, both the participants and the simulated navigator are still able to reach the goal, but both do so using variants of the learned paths.

The proposed model is based on Bayesian modeling. A single probability distribution encodes the learned path. It encodes properties of the learned path, and can be used to generate different strategies according to the availability of cues. In the Landmark condition, the probability distribution can be read directly, and the navigator performs as if using a landmark-based navigation strategy. Whereas, in the Probe Trial condition, the probability distribution can be used to generate the best estimate about actions to perform, thanks to Bayesian inference, and the navigation then performs as if using a geometry-based navigation strategy. Having a single model, which is the basis of several navigation strategies, departs from the classical view where each strategy is independently encoded and which requires an arbitrator for hierarchically articulating them.

This could provide novel insights into the cognitive mechanisms involved in human navigation and space representation, and hopefully, could be transferred to biomimetic robotic architectures, where managing hierarchical representations of complex environments is still a challenge.

Acknowledgments

This work has been supported by the BIBA and BACS European projects (FP5-IST-2001-32115, FP6-IST-027140).

References

Berthoz, A., Amorim, M., Glasauer, S., Grasso, R., Takei, Y., & Viaud-Delmon, I. (1999). Dissociation between distance and direction during locomotor navigation. In R. Golledge (Ed.), *Wayfinding behaviour* (pp. 328–348). John Hopkins University Press.

Diard, J., & Bessière, P. (2008). Bayesian maps: probabilistic and hierarchical models for mobile robot navigation. In P. Bessière, C. Laugier, & R. Siegwart (Eds.), *Probabilistic reasoning and decision making in sensory-motor systems* (Vol. 46, pp. 153–176). Springer-Verlag.

Diard, J., Bessière, P., & Mazer, E. (2003). A survey of probabilistic models, using the bayesian programming methodology as a unifying framework. In *The second int'l conf on computational intelligence, robotics and autonomous systems (CIRAS 2003).* Singapore.

Franz, M., & Mallot, H. (2000). Biomimetic robot navigation. *Robotics and Autonomous Systems*, *30*, 133–153.

Hauskrecht, M., Meuleau, N., Kaelbling, L. P., Dean, T., & Boutilier, C. (1998). Hierarchical solution of Markov decision processes using macro-actions. In G. F. Cooper & S. Moral (Eds.), *Proc. of the 14th conf on uncertainty in artificial intelligence (UAI-98)* (pp. 220–229).

Jacobs, L. F., & Schenk, F. (2003). Unpacking the cognitive map: the parallel map theory of hippocampal function. *Psychological Review*, *110*(2), 285–315.

Janzen, G., & Turennout, M. van. (2004). Selective neural representation of objects relevant for navigation. *Nature Neuroscience*(6), 673–677.

Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., & Savelli, F. (2004). Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *Proc. of the ieee int'l conf on robotics and automation (ICRA04)* (pp. 4845–4851).

Kuipers, B. J. (2000). The spatial semantic hierarchy. *Artificial Intelligence*, *119*(1–2), 191–233.

Lane, T., & Kaelbling, L. P. (2002). Nearly deterministic abstractions of markov decision processes. In *Eighteenth national conf on artificial intelligence (AAAI-2002).*

Mallot, H., & Gillner, S. (2000). Route navigating without place recognition: what is recognised in recognition-triggered responses? *Perception*, *29*, 43–55.

Redish, A. D., & Touretzky, D. S. (1997). Cognitive maps beyond the hippocampus. *Hippocampus*, *7*(1), 15–35.

Stackman, R., & Taube, J. (1997). Firing properties of head direction cells in the rat anterior thalamic nucleus: dependence on vestibular input. *Journal of Neuroscience*, *17*, 4349–4358.

Stankiewicz, B. J., & Kalia, A. A. (2007). Acquisition of structural versus object landmark knowledge. *Journal of Experimental Psychology: Human Perception and Performance*, *33*(2), 378–390.

Taube, J. (1998). Head direction cells and the neurophysiological basis for a sense of direction. *Progress in Neurobiology*, *55*, 225–256.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, *99*(1), 21–71.

Trullier, O., Wiener, S., Berthoz, A., & Meyer, J.-A. (1997). Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, *51*, 483–544.

Victorino, A. C., & Rives, P. (2004). An hybrid representation well-adapted to the exploration of large scale indoors environments. In *Proc. of the ieee int'l conf on robotics and automation (ICRA04)* (pp. 2930–2935).

Voicu, H. (2003). Hierarchical cognitive maps. *Neural Networks*, *16*, 569–576.

Wang, R. F., & Spelke, E. S. (2002). Human spatial representation: insights from animals. *TRENDS in Cognitive Science*, *6*(9), 376–382.

Large Declarative Memories in ACT-R

Scott Douglass (Scott.Douglass@mesa.afmc.af.mil)
Jerry Ball (Jerry.Ball@mesa.afmc.af.mil)
Human Effectiveness Directorate, 711th Human Performance Wing
Air Force Research Laboratory
6030 S. Kent Street, Mesa, AZ 85212

Stuart Rodgers (stu@agstechnet.com)
AGS TechNet
Dayton, OH

Abstract

The development of large-scale cognitive models introduces significant computational challenges. Large declarative memories are a case in point. It is not computationally feasible to load a large declarative memory into the process space available for execution of a cognitive model. Fortunately, computer science provides us with relational databases to support access to large external stores of information from within an executing process. This paper motivates and describes the interfacing of the ACT-R cognitive architecture with a relational database to support large declarative memories within ACT-R models.

Keywords: large-scale cognitive modeling; large declarative memory, ACT-R cognitive architecture, relational database.

The Need for Large Declarative Memories

The typical cognitive model models a specific laboratory task with modest declarative memory (DM) requirements. The DM of such models can be loaded into the process space of the model and executed efficiently. The ACT-R cognitive architecture (Anderson, 2007; Anderson et al., 2004) comes with efficient data storage and access mechanisms for managing modest size declarative memories within the process space of the model. However, the development of cognitive models of complex tasks requires more substantial DMs. At some point, the size of declarative memory becomes too large to be loaded into the executing process as a whole and external data storage and access mechanisms are needed.

Researchers in the Air Force Research Laboratory, Human Effectiveness Directorate, Cognitive Models and Agents Branch (AFRL/RHAC) in collaboration with the Cognitive Engineering Research Institute (CERI), AGS TechNet and L3 Communications are engaged in a project to develop a Synthetic Teammate (Ball et al., 2009) capable of functioning as the Air Vehicle Operator (AVO), or pilot, in a 3-person team task simulation of an Uninhabited Aerial Vehicle (UAV) performing reconnaissance missions (Cooke & Shope, 2005). All the major components of the Synthetic AVO are being developed within the ACT-R cognitive architecture, including language comprehension (Ball, Heiberg & Silber, 2007), language generation, dialog and situation models, and task behavior (Myers, to appear). The use of ACT-R reflects our commitment to develop a cognitively plausible, yet functional synthetic agent. We believe that adhering to well-established cognitive constraints may actually facilitate the development of functional agents by pushing development in cognitively plausible directions which are more likely to be successful in modeling complex human behavior than non-cognitively plausible alternatives.

The Synthetic AVO must communicate with the planning officer, who plans the route, and the payload operator, who takes pictures of targets, in order to accomplish a 40 minute reconnaissance mission involving more than 12 waypoints, many of which are targets. As a result, language comprehension is an important component of the larger Synthetic AVO model. Further, the range of vocabulary and grammatical constructions used by the teammates to communicate with the AVO is extensive and unpredictable. An analysis of spoken transcripts between all human teams participating in several earlier studies identified a total of 2500 unique words in 19K spoken utterances and an analysis of text chat transcripts in a recent study identified 1700 unique words in 5500 text chat messages. Overall, we expect the model to require a vocabulary of 10-15K words and multi-word units to be capable of adequately modeling human communicative behavior on this complex task. By itself, the language comprehension component is pushing the scale of DM beyond the capacity of the existing ACT-R data storage and access mechanisms.

To support the projected vocabulary, we are pursuing the integration of a large subset of the WordNet lexicon (Fellbaum, 1998; Miller, 1998) into the model.

Table 1: Summary of the SGP parameters introduced by the persistent-DM module.

Parameter Name	**Description of Parameter Behavior**
PDM-active	Enable/disable the use of persistent DM
PDM-add-DM-serializes	Determines if add-dm produces DB chunks or not
PDM-resets-clear-DB	Enable/disable the clearing of the persistent DM during model resets
PDM-DB-name	Name of PostgreSQL database containing the persistent DM of interest
PDM-user	Username required by the PostgreSQL DBMS for DB access
PDM-passwd	Password required by the PostgreSQL DBMS
PDM-DBMS-hostname	DBMS hostname provided as a machine name or IP address

To accomplish this, we are leveraging the WN-Lexical interface to ACT-R developed by Bruno Emond (2005). The WN-Lexical interface provides a capability to load all of WordNet into ACT-R's DM at once. However, on our hardware the model exhausts the memory capacity after just 30% of WordNet is loaded.

Although we do not envision using the entire WordNet lexicon in our language comprehension model, we do expect to use a large enough subset for it to be problematic for the existing ACT-R data storage and access capabilities. To support the integration of a large subset of the WordNet lexicon into the language comprehension model, an external data storage system is needed. WN-Lexical provides a capability to load individual words into DM as needed, retaining unused words in an external store. However, this capability is not tightly integrated with ACT-R's declarative memory module and cannot take advantage of ACT-R DM mechanisms like spreading activation. This is problematic since there is a high level of lexical ambiguity in the WordNet lexicon (e.g. the word "dog" has eight senses in WordNet) and spreading activation is a key mechanism for dealing with such ambiguity. Ideally, **the external data storage capability should be transparent from the perspective of ACT-R and DM**—i.e. whether the model is accessing a word from an internal or external data store should not affect the behavior of the model. The next section describes just such a capability.

Persistent DM for ACT-R

Current Declarative Module

The chunks constituting declarative memory in ACT-R 6 are stored internally in a single data structure. When a retrieval request is executed by the ACT-R declarative module, a process carried out by the module *essentially* uses constraints in the retrieval request and computed activations to identify which chunk matching the constraints (if any) should be accessed from the data structure and placed into the retrieval buffer. This process is simple and effective when the number of chunks in the data structure remains below a certain threshold. As the number of chunks in declarative memory increases, the process slows and eventually breaks down.

An ACT-R user wanting to model cognitive processes dependent on declarative memories larger than a critical threshold therefore requires new data storage and access mechanisms to support DM chunk storage and retrieval. Fortunately, the modular nature of ACT-R and the software design of ACT-R 6 greatly facilitate the development and deployment of alternative chunk storage and retrieval mechanisms.

New SQL Functionality

To meet large DM requirements, we've developed a chunk storage and retrieval capability in ACT-R 6 based on PostgreSQL, a powerful, open source object-relational database management system (DBMS). This "*persistent-DM*" module outsources chunk storage to an industrial-strength external DBMS while leaving ACT-R's retrieval calculus untouched. The persistent-DM module (defined in a single file) modifies the behavior of ACT-R's declarative module by: (1) introducing seven control parameters; (2) providing programmatic support for managing the interaction between ACT-R and the PostgreSQL DBMS; (3) extending the retrieval process; and (4) modifying the comparison of chunk slots. Table 1 describes the seven control parameters.

The persistent-DM module's parameters allow the ACT-R modeler to easily control the behavior of the module. For example, toggling the *PDM-active* parameter from T (on) to NIL (off) disables use of PostgreSQL and returns the chunk storage/retrieval behavior of ACT-R back to its default. Setting *PDM-add-DM-serializes* and *PDM-resets-clear-DB* to T (yes) when persistent DM is enabled allows a modeler to populate the persistent DM with chunks explicitly defined in a model. Lastly, setting *PDM-add-DM-serializes* and *PDM-resets-clear-DB* to NIL (no) when persistent DM is enabled allows a modeler to make a DM that persists across model runs available, without having to comment out parts of the model. The persistent-DM module provides the ACT-R modeler with programmatic support for the definition and management of external PostgreSQL databases. A modeler using the persistent-DM module can programmatically:

– Generate and use generic SQL queries to interact with persistent external knowledge bases.
– Serialize (write) and de-serialize (read) ACT-R chunks in massive knowledge bases.
– Use transactions, rollbacks and commits to protect, undo, and save changes to declarative memory.

Most importantly, a modeler using the persistent-DM module can transparently:

– Employ PostgreSQL DBMS-based alternatives to the default chunk addition, removal, and merging processes in ACT-R 6. These alternatives don't change the calculus underlying ACT-R's declarative module, they just change the way retrieval requests are used to determine the subset of chunks from declarative memory that will participate in the calculation of activation during the retrieval process.
– Use retrieval constraints based on regular expressions.

To use the persistent-DM module, an ACT-R modeler needs to: (1) install the PostgreSQL DBMS on a computer (the computer can be the modeler's workstation or a dedicated remote server); (2) install a common-lisp library supporting interaction with PostgreSQL; (3) drop the persistent-DM module definition file into the ACT-R 6 "modules" directory; (4) activate the module by adding something like the following to the SGP section of a model.

```
(sgp :pdm-db-name "model-v5-DM"
     :pdm-user "Scott"
     :pdm-passwd “Open_Seseme"
     :pdm-resets-clear-db T
     :pdm-add-dm-serializes T
     :pdm-active T
...
```

Figure 1: Activating and configuring the persistent-DM module in an ACT-R model.

The activation of the persistent-DM module has no impact on model behavior. However, wall clock performance of the ACT-R simulator is impacted. Chunk serialization and de-serialization processes depend on non-trivial information exchanged with the PostgreSQL DBMS and using persistent-DM when models have small declarative memories exacts a fixed and relatively high cost. If the cost of using persistent-DM remains essentially fixed, then persistent-DM will eventually outperform ACT-R's default declarative memory system when models have large enough declarative memories. To find out where this tipping point is, and to better understand when we should and shouldn't use the persistent-DM module, we conducted a comparative analysis of default and persistent-DM.

Computational Efficiency of Retrievals from Different Size Declarative Memories

When the number of chunks maintained by an ACT-R model remains low, keeping them on-hand in an internal data structure facilitates optimal simulator performance. Under these circumstances, the cost of forming an external query, dispatching the query to a DBMS, and interpreting the return from a DBMS exceeds the cost of comparing candidate chunks to retrieval constraints. When the number of chunks maintained by an ACT-R model exceeds a certain value, keeping them on-hand in an internal data structure exceeds lisp/machine memory limits and the framework crashes. Under these circumstances, modeling can only proceed if a DBMS is used. Between these two extremes is a decision space in which the benefits of using persistent-DM gradually exceed the costs. To start exploring the nature of this decision space, a controlled evaluation of the performance of the persistent-DM was conducted. During this evaluation, three factors were systematically varied:

– *Type of DM*: default or persistent
– *Size of DM*: ~1K, ~5K, ~10K, ~20K, ~80K or ~240K chunks
– *Retrieval Constraints*: 1, 2, 3 or 4 slot/value requirements

Each of the differently sized DMs was defined by a separate ASCII file containing a single call to ACT-R's "add-dm" command. Under conditions where the type of DM being evaluated was default, these files were loaded into ACT-R and "add-dm" added chunks to the internal chunk table. Under conditions where the type of DM being evaluated was persistent, PostgreSQL databases containing these same chunks were connected to by the persistent-DM module.

```
aardwolf-noun-pos
  ISA noun
   parent  "none"
   token  "type"
   type  noun
   super-type  noun
   subtype  noun
   form  nil
   word  aardwolf-word
   gram-form  common-sing
   animate  animate
```

Figure 2: ACT-R chunk specification of a noun describing an aardwolf part-of-speech.

Chunks used in the evaluation represented nouns. The ACT-R specification of the noun chunk type consisted of nine slots (see Figure 2).

Regardless which type of DM was being evaluated, retrieval requests intended to recover 10 randomly chosen chunks from each differently sized DM were executed to assess wall-clock retrieval times. Figure 3 shows example retrieval requests based on 1 and 4 retrieval constraints.

```
+retrieval>
  ISA noun
   parent  "none“
...
+retrieval>
  ISA noun
   parent  "none"
   super-type  noun
   word  aardwolf-word
   gram-form  common-sing
```

Figure 3: Example ACT-R retrieval requests based on 1 and 4 retrieval constraints.

Additional retrieval constraints lead to more specific retrievals but require additional slot/value comparisons. Evaluations of ACT-R's retrieval process lead us to believe that the use of additional slot/value constraints in more constrained retrieval requests would impose a time cost when ACT-R's default retrieval mechanisms are employed. We incorporated the retrieval constraints factor into the evaluation in order to systematically assess the actual costs (if any) of employing greater retrieval constraints. Due to database indexing and SQL query optimizations; additional constraints shouldn't impose similar time costs. The incorporation of the retrieval constraints factor into the evaluation allowed us to directly assess the efficiency (or lack of) of SQL queries based on composed constraints.

During the evaluation, 2 performance measures were recorded:

1. *Setup-time*: The amount of time it took to make chunks in the differently sized DMs available to ACT-R's retrieval process.
2. *Retrieval-time*: The amount of time it took to actually retrieve a chunk matching the retrieval constraints.

Table 2 lists the average setup times we measured in the evaluation of default and persistent DM. Times in the table clearly show that loading a declarative memory specification into ACT-R through default methods requires an increasing amounts of time when declarative memory size increases. The details of the relationship between the size of DM and set-up time, while interesting, do not contribute to the point that as declarative memory grows, a load-time problem appears. Consequently, they won't be discussed further. The failure of ACT-R to load 240,000 chunks into default declarative memory provides us with an initial estimate of the number of chunks—at least as complex as our noun chunk type—beyond which ACT-R becomes unstable on our hardware. Lastly, the cost of connecting to an external PostgreSQL DBMS was found to be relatively constant.

Table 2: Summary of setup times (in msec)

	~1K	~5K	~10K	~20K	~80K	~240K
default	98	375	981	2828	103395	NA
persistent	82	90	90	94	86	86

Ten retrieval times were recorded under all 6x4 combinations of DM size and retrieval constraints. These measures were analyzed using a repeated measures ANOVA. Since default DM was unable to accommodate 240,000 chunks, retrieval times are missing in Figure 4. The persistent DM retrieval times under these circumstances were not included in the repeated measures ANOVA. All main effects and interactions were found to be highly significant. The significant Size of DM X Retrieval Constraints X Type of DM interaction (F(12,108) = 3.682, p<0.001) is illustrated in Figure 4. The figure shows that while wall-clock retrieval times are uninfluenced by the size of declarative memory when persistent DM is used, they are significantly influenced by the size of declarative memory when default DM is used. When declarative memory contains more than approximately 80,000 chunks, the benefits of keeping chunks in an internal data structure are lost. 80,000 chunks seem to be the point at which things decidedly favor persistent-DM; at least given the complexity of our noun chunk type. Using additional retrieval constraints imposes no additional time costs on persistent DM. A clear, and eventually significant, relationship between constraints and retrieval time can be seen in default DM.

While this simple evaluation reveals some of the capabilities of persistent-DM, much work needs to be done. For example, activations were not computed in the comparative study. When sub-symbolic chunk properties are calculated and maintained in ACT-R's declarative module, symbolic properties of candidate chunks such as fan and type inheritance can easily lead to a dramatic need to obtain properties of non-candidate chunks. In order to begin to understand the costs and benefits of persistent-DM, the calculation of activation was inhibited. We are planning follow-up evaluations that will systematically control fan and chunk type inheritance in order to further explore the capabilities of the persistent-DM module.

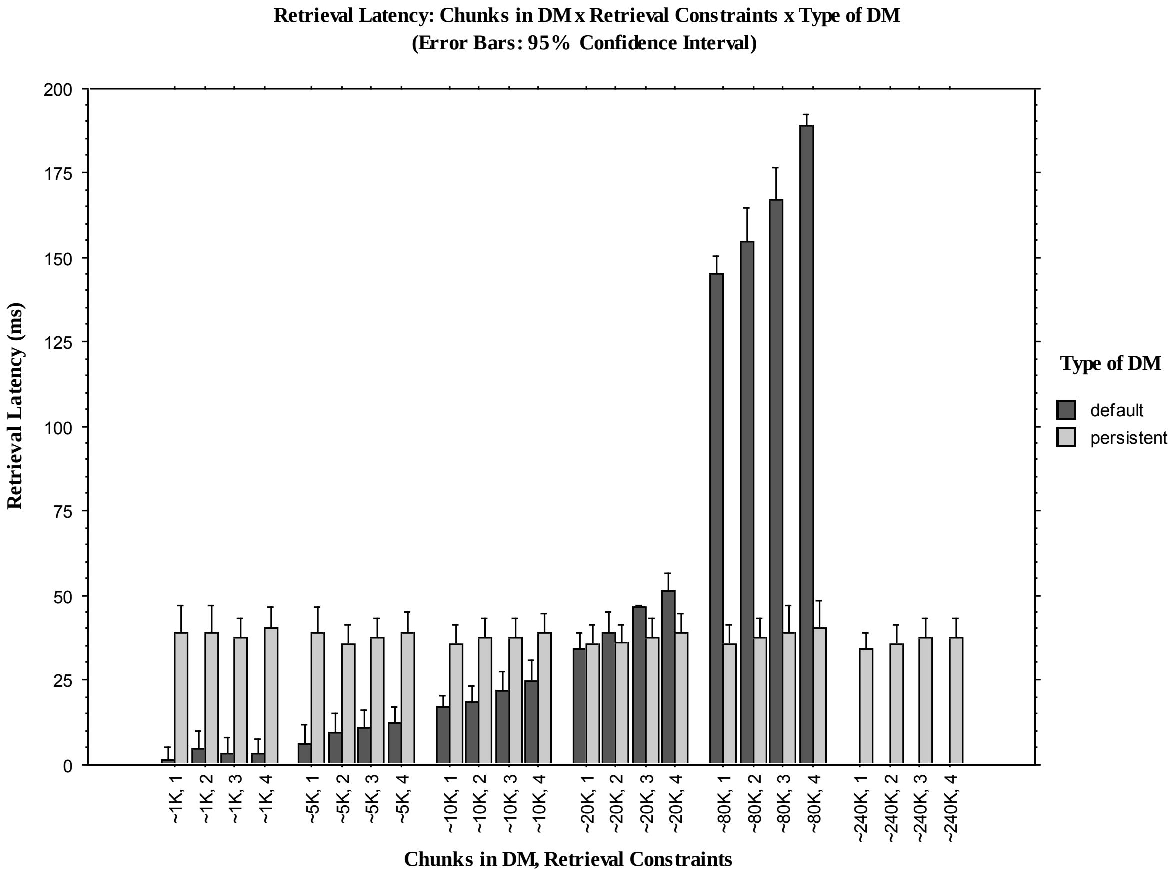


Figure 4: Summary of wall-clock retrieval times.

Conclusions

We have integrated a relational database with the ACT-R cognitive architecture to support the creation of large, externalized, persistent declarative memories whose behavior matches that of existing internal declarative memories. We are using this capability in the development of a complex model of a Synthetic AVO capable of communicating with human teammates in performance of a reconnaissance task. More generally, such a capability is needed in the development of complex cognitive models with significant declarative memory requirements and this capability aligns with our research focus on large-scale cognitive modeling (cf. Douglass & Luginbuhl, 2008).

Future Directions

The interface to the external database is currently functional and outperforms ACT-R's internal data storage and access mechanisms on large declarative memories under a range of conditions as demonstrated in the previous section. However, we believe there is room for significant improvement and optimization of the performance of the interface. In particular, the retrieval of a word from DM actually involves a chain of retrievals which includes all the elements in the fan list of the word. In the worst case, this chain could consume much of declarative memory, bringing the system to its knees. Besides needing to retrieve the fan list for a word in order to compute activations, if the retrieval template is highly unconstrained, many DM chunks will match and the computations may exceed process internal resource capacities. We bumped into this problem early on in a version of the model which only used spreading activation based soft constraints on word retrieval. If the retrieval template contains no hard constraints on the form of the word to be retrieved, relying exclusively on spreading activation to bias the retrieval, then ACT-R must compute the activation of every word in DM to determine which word to retrieve. This is computationally explosive and was unworkable with a mental lexicon of just 2500 words. We were forced to reinstate a whole word hard constraint on retrievals, with a fallback to a first letter hard constraint and spreading activation based soft-constraint on matching words if the whole word retrieval fails. Currently, these retrievals are executed by different productions. However, we are exploring the possibility of using the regular expression capability provided by the persistent DM module (a slot name preceded by "~"

invokes this regular expression matching capability) to conditionally retrieve the whole word first and if that fails then retrieve the word with highest activation matching the first letter, all within a single retrieval production. Given a conditional retrieval capability, a whole word match would terminate the retrieval, returning the matched word, and the less constrained and more computationally expensive first letter match with spreading activation over matching words would not occur.

We are also planning on using the regular expression matching capability to support retrieval of perceptual units at multiple levels of representation. The perceptual module of ACT-R currently divides the linguistic input into word units using a function called chop-string. This function relies on spaces and punctuation to delimit words. For example, the input “he went.” would be divided into “he” “went” and “.”. However, sometimes words contain punctuation and shouldn’t be divided—for example “etc.” and “didn’t”. And sometimes words can have a space as in “ad hoc” and “a priori”. In the case of “didn’t”, the chop-string function returns “didn” “’” and “t” and it takes three attention fixations and several productions per “word” to process this input. Given the rapidity with which humans process language during reading—approximately 225 msec per space delimited word during silent reading (Rayner, 1998)—this treatment of “didn’t” is unlikely to be cognitively plausible. To bring the language comprehension model into closer alignment with reading results, what is needed is a capability to recognize the largest unit in DM which matches the input, often matching multi-word units in a single attention fixation. To achieve this we are implementing a capability to do retrievals at multiple levels using a disjunction of perceptual units derived from the input. For example, “didn’t” will lead to an attempt to retrieve either “didn’t” or “didn” within a single retrieval specification, “John’s” (as in “John’s book”) will lead to an attempt to retrieve either “John’s” or “John”, “etc.” will lead to an attempt to retrieve either “etc.” or “etc”, “went.” will lead to an attempt to retrieve either “went.” or “went”, “a priori” will lead to an attempt to retrieve either “a priori” or “a” and “because of” will lead to an attempt to retrieve either “because of” or “because”.

Finally, longer term we are contemplating pushing ACT-R’s activation computation into the database—transparently from the perspective of ACT-R and DM. This would avoid the need to retrieve large numbers of chunks from external DM in order to compute their activations within the ACT-R process.

References

Anderson, J. R. (2007). *How Can the Human Mind occur in the Physical Universe?* NY: Oxford.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S, Lebiere, C, and Qin, Y. (2004). An Integrated Theory of the Mind. *Psychological Review* 111, (4). 1036-1060

Ball, J., Heiberg, A. & Silber, R. (2007). Toward a Large-Scale Model of Language Comprehension in ACT-R 6. *Proceedings of the 8th International Conference on Cognitive Modeling*, pp. 163-168. Edited by R. Lewis, T. Polk & J. Laird.

Ball, J, Myers, C., Hieberg, A., Cooke, N., Matessa, M. & Freiman, M. (2009). The Synthetic Teammate Project. Paper presented at the *Behavior Representation in Modeling and Simulation Conference*.

Cooke, N. & Shope, S. (2005). Synthetic Task Environments for Teams: CERTT’s UAV-STE. *Handbook on Human Factors and Ergonomics Methods*. 46-1-46-6. Boca Raton, FL: CLC Press, LLC

Douglass, S. & Luginbuhl, D. (co-chairs) (2008). Cognitive Modeling and Software Engineering Workshop.

Emond. B. (2006). WN-LEXICAL: An ACT-R module built from the WordNet lexical database. In *Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 359-360). Trieste, Italy.

Fellbaum, C. (ed) (1998). *WordNet: An electronic lexical database. Cambridge*. MA : MIT Press.

Miller, G.A. (1998). Foreword, In C. Fellbaum, (ed), WordNet: An electronic lexical database. Cambridge, MA: MIT Press

Myers, C. (to appear). An Account of Reuse and Integration. *Proceedings of the 9th International Conference on Cognitive Modeling*

Rayner, K. (1998). Eye Movements in Reading and Information Processing: 20 Years of Research. *Psychological Bulletin*, 124(3), 372-422.

An Instance-Based Learning Model of Stimulus-Response Compatibility Effects in Mixed Location-Relevant and Location-Irrelevant Tasks

Varun Dutt (varundutt@cmu.edu)[1]
Motonori Yamaguchi (myamaguc@psych.purdue.edu)[2]
Cleotilde Gonzalez (coty@cmu.edu)[1]
Robert W. Proctor (proctor@psych.purdue.edu)[2]

[1] Dynamic Decision Making Laboratory
Social and Decision Sciences Department
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[2] Department of Psychological Sciences
Purdue University, 703 Third Street, West Lafayette, IN 47907-2081 USA

Abstract

This paper presents a cognitive model of stimulus-response compatibility (SRC) effects for a situation in which location-relevant and location-irrelevant tasks are intermixed within a single trial block. We provide a computational explanation of the cognitive processing involved in the mixed-task condition. The model is based on the Instance-Based Learning Theory, developed originally to explain decision making in dynamic tasks, and the ACT-R theory of cognition. The comparison of the model's outputs to human data demonstrates high similarity, and the model offers an explanation for sequential modulations of the SRC/Simon effects observed when compatible and incompatible trials repeat or switch. Several possibilities to apply this model to novel tasks are discussed.

Keywords: Instance-Based Learning; ACT-R; Simon effect; Stimulus-response compatibility; Situation; Decision; Utility; Experience.

Introduction

In everyday activities, there are numerous situations where one is required to perform multiple tasks concurrently or in a sequence. The nature of performance is often altered in such a task condition, compared to that for a single task performed in isolation. Thus, studies of multi-task performance have been of major interest to basic and applied researchers. The main aim of the present research is to develop a computational model of human performance in a multi-task condition, in which task performance is known to be different from that for a single-task condition.

In choice-reaction tasks, responses are faster and more accurate when stimuli are mapped to spatially compatible responses than when they are mapped to spatially incompatible responses. The difference in response times and accuracy for the compatible and incompatible mappings is termed the *stimulus-response compatibility* (SRC) effect. SRC has been recognized as one of the critical principles for human interface design (Proctor & Vu, 2006) as well as a major motivation for theories of human perception and action (Hommel, Müsseler, Aschersleben, & Prinz, 2001).

The SRC effect is known to be so robust that it is obtained even when stimulus location is irrelevant to performing the task, the variation known as the *Simon Effect* (Simon, 1990). The robustness of the SRC/Simon effects has also been demonstrated using a variety of stimuli (Proctor, Yamaguchi, Zhang, & Vu, 2009), response modes (Wang, Proctor, & Pick, 2003), and more realistic tasks such as flight operations (Yamaguchi & Proctor, 2006).

However, the Simon effect can be reduced, eliminated, or even reversed when participants practice a choice-reaction task with the incompatible mapping prior to performing the Simon task (Proctor & Lu, 1999). Similarly, the Simon effect is attenuated when participants perform the Simon task concurrently with the SRC task (Marble & Proctor, 2000); that is, when location-irrelevant (the Simon task) and location-relevant (the SRC task) tasks are intermixed. The Simon effect increases somewhat when the SRC task requires a compatible mapping but reverses to a negative effect of at least the same absolute size when the SRC task requires an incompatible mapping.

A dominant cognitive explanation of the SRC/Simon effects is a *dual-route account* (Proctor & Vu, 2006), which assumes two distinct response-selection processes, characterized as direct and indirect routes. The indirect route is presumed to activate a response based on the intentions created through the instructed stimulus-response (S-R) mappings. In contrast, the direct route is presumed to automatically activate a response corresponding to the stimulus location, which facilitates responding when that response is correct but interferes when it is incorrect. However, given recent findings that the SRC/Simon effects can be attenuated in mixed-task conditions and after practice with an incompatible-mapping task, the response-selection process that gives rise to the SRC/Simon effects does not seem to be as purely automatic as it is typically described in the literature.

In contrast to the dual-route account, the present paper provides a computational model of the SRC/Simon effects developed based on the Instance-Based Learning Theory (IBLT; Gonzalez, Lerch & Lebiere, 2003). The goal of the current paper is to determine how the IBLT would predict the learning and performance obtained from an experiment in which human subjects performed mixed Simon and SRC tasks.

We first provide a description of the task and the human data collection protocols, and then present the development of the IBLT model and the fits of the model predictions to human data. The paper concludes with examples of how the model can be used to generate predictions for novel task conditions.

Experiment on Mixed Simon/SRC Task

The task adopted here is similar to those used by Marble and Proctor (2000), in which participants performed mixed location-relevant and location-irrelevant tasks. Though their experiments separately examined the influences of the compatible and incompatible mappings on the Simon effect by individually mixing these mapping trials with the Simon task, the present experiment mixed both compatible- and incompatible-mapping trials with the Simon task.

Thirty-two undergraduate students at Purdue University participated in the experiment. They were recruited from the subject pool of introductory psychology courses and received partial course credits. All participants reported having normal or corrected-to-normal visual acuity, normal color vision, and normal hearing.

The experiment was conducted individually in a dimly lit cubicle and controlled by a custom application constructed by VisualBasic 6.0 (VB). The imperative stimuli were circles (5 mm in diameter) presented on the left or right side of the screen (6 cm from the center of the screen). The circles were colored in green, red, or white. Participants responded according to the color of the stimulus on some trials (the Simon trials) and to the location of the stimulus on other trials (the SRC trials). Green and red circles were used for the Simon trials, and a white circle was used for the SRC trials. Responses were made by pressing a left (‘z’) or right (‘/’) key on the computer keyboard.

On the Simon trials, a red circle required pressing of one response key, and a green circle required pressing of the other response key. The color-key mapping was counterbalanced across participants. The location of the circle was task-irrelevant.

On the SRC trials, a mapping cue, a horizontal or vertical line (5 mm in length) colored in white and centered on the screen, was presented simultaneously with the white circle. For half of the participants, a horizontal line required pressing a response key on the same side as the location of the circle (compatible-mapping trials), and a vertical line required pressing a response key located on the opposite side to the location of the circle (incompatible-mapping trials). For the other half, the cue-mapping relation was reversed.

Each trial started with a white fixation cross presented at the center of the screen for 500 ms, followed by a blank screen lasting for 500 ms. Then, a circle appeared on the left or right of the screen, with a horizontal or vertical line if it was an SRC trial. The circle was presented until a response was made or for 1,500 ms if no response was made. When participants pressed an incorrect key, an error tone was presented from the external speakers positioned on the left and right of the screen. The frequency of the tone was 400 Hz, lasting for 500 ms. No feedback was given for a correct response but a blank display was presented for 500 ms. Thus, the inter-trial intervals for correct and incorrect responses were the same. A trial ended with a 1-s blank screen, and the next trial started with the fixation cross.

Response time and accuracy were recorded on each trial. Response time was the interval between onset of a circle and depression of a response key. Both speed and accuracy were emphasized in instructions. An experimental session lasted less than an hour.

Each participant performed four trial blocks. In each block, 80 trials were the Simon trials, and another 80 trials were the SRC trials (40 trials for the compatible mapping and 40 trials for the incompatible mappings). These trial conditions appeared equally often in each block in a random order.

An experimental session started with a block of practice trials. The practice block consisted of 16 Simon trials and 16 SRC trials (8 trials for each mapping). Participants were allowed to repeat the practice block as many times as they wanted, so that they were sufficiently familiar with the task requirements (no participants repeated more than 3 practice blocks). Results of this experiment are presented in a later section, where they are compared to the results of the IBLT computational model.

Development of the IBLT Model

We propose that IBLT provides reasonable cognitive explanations for the SRC and Simon effects. IBLT was originally developed as a way to explain and predict decision making in dynamic, complex tasks (Gonzalez, et al., 2003). The theory evolved from the idea that decisions are made from experience and that one could manipulate experience and therefore predict decisions made in the future.

IBLT proposes that people remember past experiences in terms of “instances.” At each decision-making situation, an instance is retrieved and reused depending on the similarity of the current situation to the cues stored in the instance.

An instance is composed of three parts: situation, decision, and utility of that decision in that situation (situation-decision-utility or SDU triplet). In IBLT, decisions from experience involve five mental stages in a closed-loop decision making cycle: recognition, judgment, choice, execution, and feedback. Although IBLT the general decision process and particular mechanisms of decisions from experience are independent from the computational implementation of the theory, IBLT has borrowed many of the proposed mechanisms from the mathematical representations in ACT-R (Anderson et al., 2004).

ACT-R is an integrated computational cognitive architecture resulting from decades of cumulative effort by an international community of cognitive researchers, and it provides IBLT with the following advantages for a computational implementation; (a) procedural and declarative memory modules, including both conscious and unconscious (i.e., statistical) reasoning and learning mechanisms, that have been validated by hundreds of laboratory experiments; (b) perceptual and motor modules that incorporate many known human-factors parameters and provide principled limitations in the interaction with an external learning environment; and (c) a method for assembling small, sub-second cognitive steps into computational models that can learn to perform increasingly complex dynamic tasks while interacting directly with information-processing systems and other human and synthetic agents. ACT-R has two levels of knowledge representation and manipulation; symbolic (knowledge representation) and subsymbolic (set of statistical and mathematical procedures to manipulate the symbolic level).

Symbolic level of the Simon/SRC model

For the current Simon/SRC task, the SDU instance (referred to as "chunk" in ACT-R) had the structure shown in Table 1. The first column defines the slot names of the instance while the second and third columns provide description of SDU slots. *Color* in Table 1 refers to the value of the color slot in the IBLT model, where it can contain the values red (R) and green (G) for the Simon trials and white (W) for the SRC trials. *Orient* is the value in the orientation slot, representing the orientation of the mapping cue used only for the SRC trials. Orient can contain horizontal (H) or vertical (V) for the SRC trials, and it is set at NO for the Simon trials where no mapping cue is used. *Position* slot provides the position of the imperative stimulus on the screen and can take only two values; left (Lt) and right (Rt). The *Decision* slot in Table 1 defines whether the decision is to press the left ('z') or right ('/') key on the computer keyboard to respond to the stimulus on the SRC and Simon trials. The *Utility* slot stores the utility of the decision, which is unknown at first and then updated after the IBLT model receives feedback from the task on its previous decision. The Utility slot can take three values; +1 (for correct decision), -1 (for incorrect decision), and 0 (unknown).

Table 1: SDU structure of Instance

Slot Name	Description	SDU
Color	Stimulus Color	Situation
Orient	Stimulus Orientation	Situation
Position	Stimulus Position	Situation
Decision	Key-press Decision	Decision
Utility	Utility of Decision	Utility
IBLT-State	State in IBLT Theory	Meta-Slot

In the above table, the *IBLT-State* (a Meta-Slot not used in IBLT model processing) could hold a value from any of the five process states of recognition, judgment, choice, execution, and feedback depending on the state of execution of the IBLT model on the Simon/SRC task. This slot only serves to distinguish the stage of IBLT modeling process.

As in Gonzalez et al. (2003), in IBLT the decision making starts with the recognition process in search for alternatives (the left or right keys) and the classification of the current situation as *typical* or *atypical*. The current situation is typical if there are memories of similar situations (i.e., instances of previous trials that are similar enough to the current situation). If it is typical, then the retrieved instance is used in judging the value of the decision to be made in the current situation. If the situation is atypical (i.e., no instance similar to the current conditions is found), a judgment heuristic is applied. Next, a decision point comes into place; whether to search for more alternatives or to execute the current best alternative. The answer to the choice is determined by the decision maker's "aspiration level," similar to Simon and March's (1958) satisficing strategy. In the Simon/SRC task, given their simplicity, the choice is simply made by making the same choice as the one in the decision slot of the retrieved instance (if nothing was retrieved, then a choice is made randomly for a key press, i.e., by a random judgment heuristic). After the execution of an action, if the response was incorrect, the SDU that led to the incorrect decision is modified by updating the Utility so as to provide a better representation of the "goodness" of that action.

The exact sequence of events in the IBLT Simon/SRC model is provided below. Each of the IBLT stages is represented by production rules (If-then rules) in ACT-R.

Recognition According to the similarity of a task situation and instances stored in memory, if there is a recognition (or retrieval) failure (as it is the case in the first trial, since there is no instances stored) the model applies a random judgment heuristic to select the type of action required by the task; else if there is a recognition (or retrieval) success the model applies an instance based judgment procedure.

Judgment When there is a recognition failure the model chooses a random number between 0 and 1 and if the number is greater than 0.5 then the right key is selected; else if it is less than or equal to 0.5 then the left key is selected. In case of recognition success, the model applies the decision of the instance that was retrieved successfully as the decision of the current instance. The model assigns the Utility slot of the current instance a value of unknown (i.e. 0) at this point.

Choice This refers to picking the selected key to press once the decision of retrieved instance or random heuristic has been made.

Execution At this step the model presses the selected key and waits for feedback for the action.

Feedback On obtaining the outcome of the decision just executed (error tone), the model updates the Utility of the current instance. If the decision made was correct it assigns Utility a value of +1; else if the decision made was not correct then it assigns Utility a value of -1.

In the above algorithm, the Utility slot is used at the time of retrieval (i.e., the recognition process) to divide the declarative memory (DM) into collections of correct and incorrect decision instances and confine the retrieval to only those instances that have in the past yielded correct decision outcomes (i.e., their Utility slots have a value of +1).

Also, in the above algorithm, the productions were assumed to take an architectural default value of 50 ms (Anderson et al., 2004). There were some steps executed to read and encode the stimulus from the screen (visual time) and also to hear and encode the feedback tone of 400 Hz frequency (auditory time) in the model (in case of negative feedback). The visual and auditory times were assumed to be at the ACT-R default values of 185 ms and 100 ms, respectively.

Sub-Symbolic level of the Simon/SRC model

In ACT-R, each instance (or chunk) has an activation value that is used for retrieval in the recognition phase of the IBLT modeling process. An instance is retrieved from memory if the activation exceeds a retrieval threshold (RT), which sets the minimum activation with which an instance can be retrieved, and if the activation is the highest of all other instance activations at the time of retrieval. ACT-R defines activation of an instance as:

$$A_i = B_i + \sum_l PM_{li} + \varepsilon \qquad (1)$$

where *Bi* is the base-level activation and reflects the recency and frequency of practice of the *i*th instance, which is given by

$$B_i = \ln(\sum_{j=1}^{n} t_j^{-d}) \qquad (2)$$

where *n* is the number of presentations of the *i*th instance in the past; t_j is the time since the *jth* presentation; and *d* is the decay parameter (bll) which is usually set at 0.5.

Specification elements *l* in the PM summation are computed over the slot values of the retrieval instance specification (i.e., the current task context). Match Scale *P* reflects the amount of weighting given to the similarity in the slot *l*, which is a constant across all slots with the value set at 1.0. Match Similarities M_{li} represents the similarity between the value *l* in the retrieval specification and the value in the corresponding slot of the instance *i*. The PM mechanism as described above determines similarity between the retrieval specification and the potential retrievable instances in DM. We used a function to calculate the degree of similarity based on the absolute value of distance between Color, Position and Orient slots of the retrieval specification and the instances stored in DM.

Finally, ε is the noise value, which is composed of two components; a permanent noise associated with each instance and an instantaneous noise computed at the time of a retrieval request. Both noise values are generated according to a logistic distribution characterized by a parameter *s*. The mean of the logistic distribution is 0 and the variance σ^2 is related to the *s* value by

$$\sigma^2 = (\pi^2/3)\, s^2 \qquad (3)$$

We set the instantaneous noise *s* value in the IBLT model to make it a part of the activation equation.

For the purpose of modeling the Simon/SRC task, the parameters described above had the values given in Table 2.

Table 2: IBLT (ACT-R) Parameters with Values

Parameter/Slots	Value
RT	-1.0
bll	0.5
s	0.25
P	1.0
Color Slot Value	G = 3, R = 5, W = 0
Orient Slot Value	H = 3, V = 5, NO = 0
Position Slot Value	Lt = 0, Rt = 1

Running the IBLT Model in the Simon/SRC Experiment

The Simon/SRC task used for the experiment was originally developed in VB, and to make things compatible we used a VB version of IBLT that we have developed and calibrated to ACT-R (in LISP), reported in another research report (Dutt, Gonzalez & Lebiere, in preparation).

We ran a total of 32 dummy model participants (the same as the number of human participants in the experiment) using exactly the same task software used to conduct the human experiment. Human participants were provided with a few practice blocks prior to the test blocks (see descriptions of the method). The collected human data revealed a high accuracy in the first block of the experiment, which suggests that the participants had obtained a certain amount of familiarity with the task prior to beginning the experiment. Therefore, the model performed two blocks of the practice phase (32 trials x 2) prior to the test trials to make the initial level of model performance comparable to that of human subjects.

Soon after the model completed the practice blocks, it was run in the experiment. The model's data on the Simon/SRC task was recorded by the task software in a text file. We later analyzed and compared the data collected from the model runs to that collected earlier on humans using commonly employed metrics R^2 (for trend) and Root Mean Squared Error (RMSE; for closeness of fits). We used response time as our dependent variable for the purpose of analysis in this paper.

Model Fits

In the present paper, we focus on two main aspects of the human data; practice and sequential effects. To examine practice effects, we first analyzed the human data across the four learning blocks separately for Simon Corresponding trials, Simon Non-Corresponding trials, SRC Compatible trials, and SRC Incompatible trials. Figure 1 presents the practice effects observed from human participants and those generated by the IBLT model (error bars show 90% confidence interval around the point estimate). For human data, RT decreases with Block for all four conditions, and the shape of the functions shows typical learning curves.

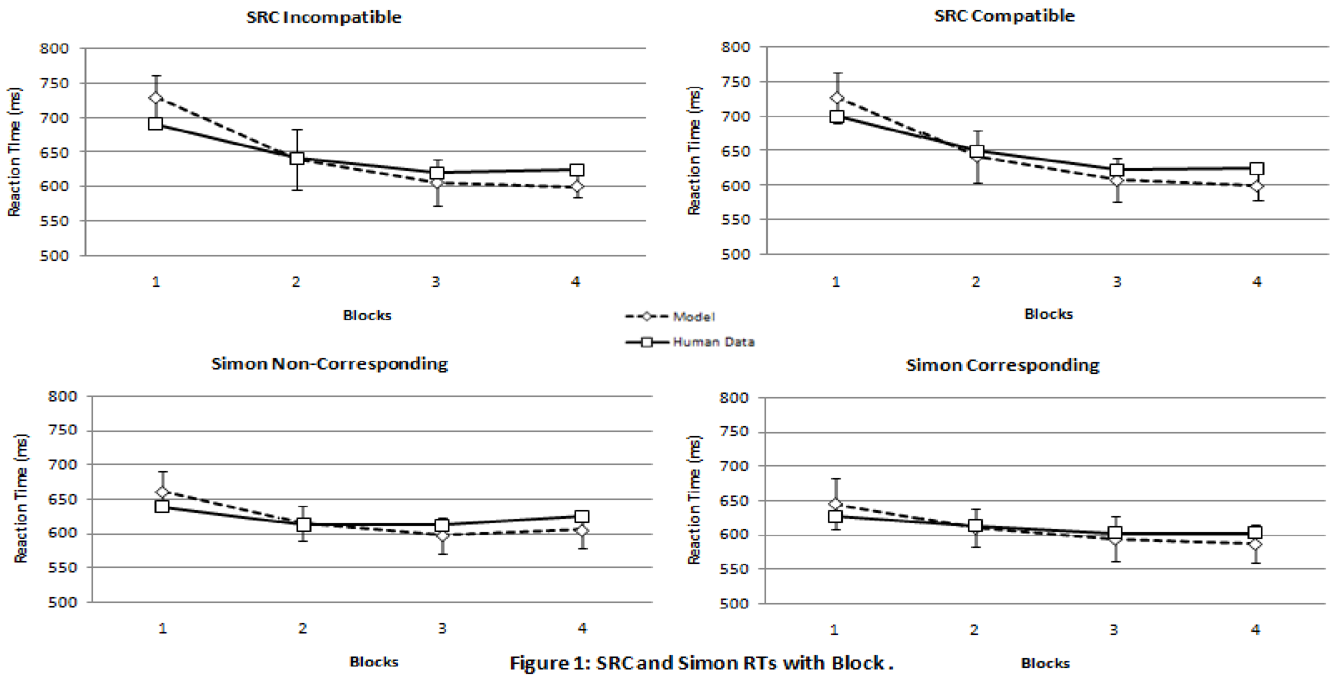


Figure 1: SRC and Simon RTs with Block. The R^2 and RMSD are reported in text.

The model output mimicked the human results, indicating that the IBLT model learned to perform the task in a similar way as human subjects did. The fit of IBLT model to the human data had R^2 = 0.93 and RMSD = 80.54 ms (Simon Corresponding trials), R^2 = 0.64 and RMSD = 84.54 ms (Simon Non-Corresponding trial), R^2 = 0.98 and RMSD = 61.21 ms (SRC Compatible trials), and R^2 = 0.97 and RMSD = 62.96 ms (SRC Incompatible trials), respectively.

To examine sequential effects, we analyzed response times as a function of Task Sequence (repeat/switch) and Mapping Sequence (repeated/switched) separately for the four trial conditions. Figure 2 (on next page) summarizes the results (error bars show 90% confidence interval around the point estimate). The trend is clear: When both task and mapping repeated, response times were reduced in both model and human data. Similarly, when both task and mapping switched, response times increased both in model and human data. The model outputs show similar patterns. The fit results are R^2 = 0.96 and RMSD = 24.87 ms, for the Simon Corresponding trials, and R^2 = 0.91 and RMSD = 31.32 ms for the Simon Non-corresponding trials. For the SRC trials, the model fits had R^2 = 0.97 and RMSD = 15.86 ms for the Compatible trials, and R^2 = 0.95 and RMSD = 28.82 for the Incompatible trials.

Discussion

As shown, model fits were generally good with respect to practice and sequential effects in the present experiment, suggesting that the IBLT model provides a good account for performance in the mixed SRC/Simon task. The learning effect in the model is explained by the IBLT process and the ACT-R mechanisms involved, in which similarity and activation play a key role.

Because the IBLT model uses only the correct instances in the selection of a response, the activation of the correct instances becomes much higher due to their repeated use and this increase in activation reduces the retrieval time for these instances. Thus, more and more correct instances are accumulated and retrieved, so that the model gradually transfers from an exploration phase (random judgment and retrieval of incorrect instances) to an instance exploitation phase (consistent retrieval of correct instances), thereby reducing reaction time over trials. The human RT for Simon Non-Corresponding trials in Figure 1 shows a slightly U-shape pattern where the RT increases for the last two blocks. On the other hand, due to recency and frequency effects just described the IBLT model reduces RT even for those two blocks. This is the reason why the fits for this condition ,particularly for R^2, are poorer compared to other fits.

Similarly, the sequential effects in the IBLT model occur because when the task and mapping repeat, the instance used on the previous trial has higher activation due to recency of its use. This increases the probability of that instance being retrieved on the current trial. Similarly, this instance-based retrieval also provides the explanation for the outcome that response times were longer when task and mapping switched.

The present model can also be used to generate predictions in novel situations (Dutt & Gonzalez, 2008). Several generalizations of the model are possible. One candidate for such a generalization is changing the proportion of the SRC and Simon trials in the experiment. The current IBLT model was implemented for an experiment where the numbers of the Simon and SRC trials are equal and thus their occurrences are equally likely on each trial. One could create situations in which the numbers of these trials are unequal, so that their likelihoods of

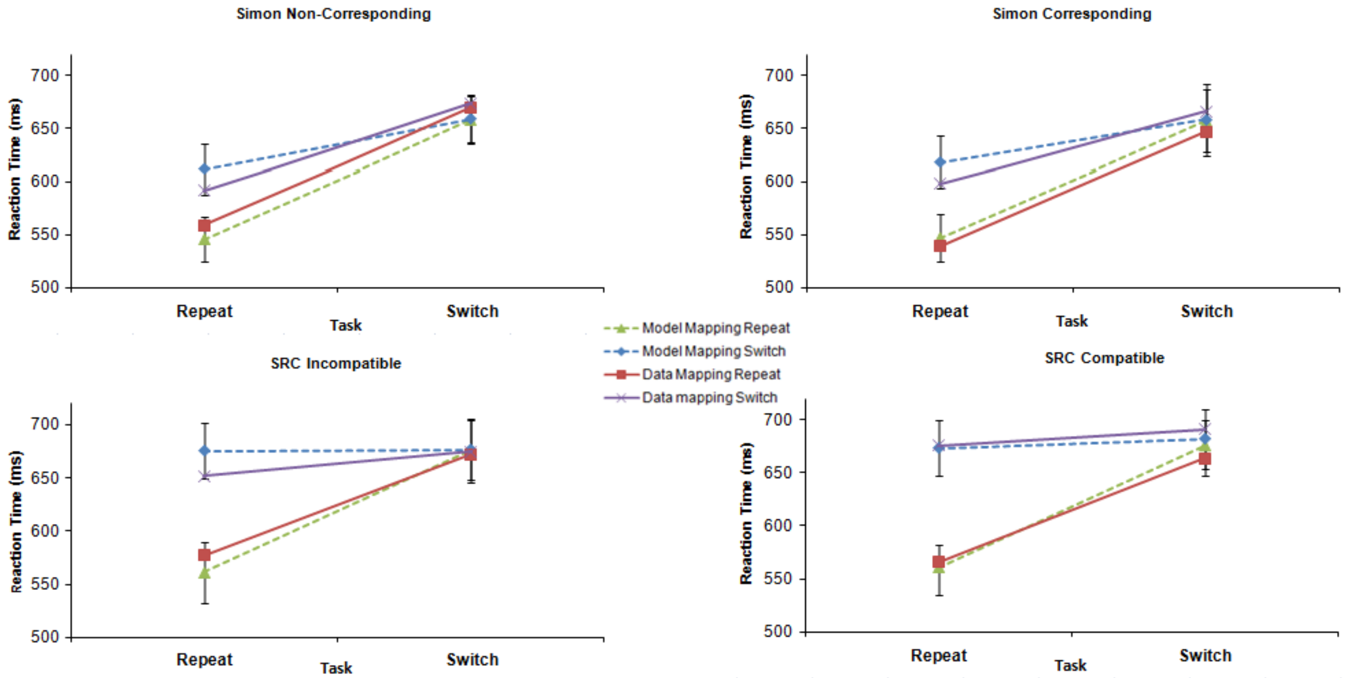


Figure 2: Task Repeat/Switch and Mapping Repeat/Switch. The R^2 and RMSD are reported in text.

occurrences are biased. Another possibility is the condition where payoffs of correct/error responses for the Simon and SRC trials are varied. For example, rather than using a tone to indicate incorrect trials, we could use monetary payoffs as feedback to the model and create conditions in which correct decisions on certain trial types are reinforced, and incorrect decisions are penalized, more than those on other trial types. Also, although the present experiment mixed location-relevant and location-irrelevant tasks, the current model can be used to predict human behavior in pure SRC or Simon tasks. This is because the model's current SDU structure does not change across the Simon and SRC trials.

The IBLT model discussed in the present paper has a general structure that, when coupled with the general functioning of the IBLT approach, provides a starting point for future investigations in the present research field. The current IBLT model can be used in a wide range of experimental conditions to generate predictions for novel tasks without major changes in the model structure, before a human experiment is conducted .

Acknowledgements

This research was supported by a Multidisciplinary University Research Initiative grant from the Army Research Office (W0112NF-05-1-0153).

References

Anderson, J. R., Bothell, D., et al. (2004). An integrated theory of mind. *Psychological Review, 111*, 1036-1060.

Dutt, V., & Gonzalez, C. (2008). Instance and strategy based models in ACT-R. *Proceedings of 2008 Modeling, Simulation & Gaming (MS&G) Student Capstone Conference* (pp. 19). Suffolk, VA: ODU-VMASC.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science, 27*, 591-635.

Hommel, B., Müsseler, J., Aschersleben, G., & Prinz, W. (2001). The theory of event coding (TEC): A framework for peception and action planning. *Behavioral and Brain Science, 24*, 849-937.

Marble, J. G., & Proctor, R. W. (2000). Mixing location-relevant and location-irrelevant choice-reaction tasks: Influences of location mapping on the Simon effect. *Journal of Experimental Psychology: Human Perception and Performance*, *26*, 1515-1533.

Proctor, R. W., & Lu, C.-H. (1999). Processing irrelevant location information: Practice and transfer effects in choice-reaction tasks. *Memory & Cognition, 27*, 63-77.

Proctor, R. W., & Vu K.-P. L., (2006). *Stimulus-response compatibility principles: Data, theory and application*. Boca Raton, FL: CRC Press

Proctor, R. W., Yamaguchi, M., Zhang, Y., & Vu, K. P.-L. (2009). Influence of visual stimulus mode on transfer of acquired spatial associations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *35*, 434-445.

Simon, H. A., & March, J. G. (1958). *Organizations*. New York: Wiley.

Simon, J. R. (1990). The effects of an irrelevant directional cue on human information processing. In R. W. Proctor & T. G. Reeve (Eds.), *Stimulus-response compatibility: An integrated perspective* (pp. 31-86). Amsterdam: North-Holland.

Yamaguchi, M., & Proctor, R. W. (2006). Stimulus-response compatibility with pure and mixed mappings in a flight task environment. *Journal of Experimental Psychology: Applied*, *12*, 207-222.

Processing grammatical and ungrammatical center embeddings in English and German: A computational model

Felix Engelmann (felix.engelmann@uni-potsdam.de)
Department of Linguistics, Karl-Liebknecht Str. 24-25,
14476 Potsdam, Germany

Shravan Vasishth (vasishth@uni-potsdam.de)
Department of Linguistics, Karl-Liebknecht Str. 24-25,
14476 Potsdam, Germany

Abstract

Previous work has shown that in English ungrammatical center embeddings are more acceptable and easier to process than their grammatical counterparts (Frazier, 1985; Gibson & Thomas, 1999). A well-known explanation for this preference for ungrammatical structures is based on working-memory overload: the claim is that the prediction for an upcoming verb phrase is forgotten due to memory overload, leading to an illusion of grammaticality (Gibson & Thomas, 1999). However, this memory-overload account cannot explain the recent finding by Vasishth, Suckow, Lewis, and Kern (2008) that in German no illusion of ungrammaticality occurs. We present a simple recurrent network model that can explain both the presence of the grammaticality illusion in English and its absence in German. We argue that the grammaticality illusion emerges as a function of experience with language-specific structures, not working memory constraints as argued for in Gibson and Thomas (1999).

Keywords: sentence comprehension ; center embeddings ; illusion of grammaticality ; working-memory models ; connectionist models

Introduction

Consider the contrast in (1), discussed first by Frazier (1985) (the original observation is attributed by Frazier to Janet Fodor). Although the rules of English grammar allow a sentence like (1a), such a complex structure is perceived by native English speakers to be less acceptable than its ungrammatical counterpart (1b), in which the middle verb phrase, *was cleaning every week*, is missing.

(1) a. The apartment that the maid who the service had sent over was cleaning every week was well decorated.
b. *The apartment that the maid who the service had sent over was well decorated.

The first published study involving this contrast was an offline questionnaire-based experiment by Gibson and Thomas (1999). Their main finding was that ungrammatical sentences such as (1b) were rated no worse than grammatical ones such as (1a). In related work, Christiansen and Macdonald (2009) show that ungrammatical sentences were rated significantly better than the grammatical ones. We will refer to this surprising finding as the *grammaticality illusion*.

At least two competing explanations exist for this illusion. One is due to Gibson and Thomas (1999), who argue that the prediction for the middle verb phrase is forgotten if memory cost exceeds a certain threshold; this explanation relies on the assumption that working memory overload leads to forgetting. The second explanation is due to Christiansen and Chater (1999) and Christiansen and Macdonald (2009), who attribute the illusion to experience (exposure to particular regularities in the syntax of a language) as encoded in a connectionist network. They trained a simple recurrent network (SRN) on right-branching and center-embedding structures and then assessed the output node activations after seeing the ungrammatical sequence NNNVV (i.e., sentences like 1b). The activations showed a clear preference for ungrammatical structures, consistent with empirical data from English speakers.

An important theoretical question is whether these two explanations—the memory-overload account and the experience-based account—can be distinguished. Although the English data is consistent with both explanations, recent work by Vasishth et al. (2008) provides revealing new evidence regarding the grammaticality illusion. Vasishth and colleagues carried out several self-paced reading and eyetracking studies demonstrating that although the English grammaticality illusion can be replicated in online measures like reading time, in German the pattern reverses: readers find the *ungrammatical* sentence (1b) harder to process than its grammatical counterpart (1a). In other words, German readers do not experience the grammaticality illusion.

Specifically, for English Vasishth and colleagues found (across several experiments) longer reading times in the grammatical condition (1a) either at the final verb or the word immediately following it (or in both regions); whereas for German they reported shorter re-reading times in the *grammatical* condition either in the final verb region and/or the region following it.

The absence of the grammaticality illusion in German is interesting because it cannot be explained by the memory-based forgetting account as stated in (Gibson & Thomas, 1999). The explanation due to Christiansen and Chater (1999), however, may be able to explain the German results (in addition to the patterns seen in English): since German relative clauses are always head-final, German readers are exposed to head-final center embeddings much more often than English speakers. This greater exposure to head-final structures could be the reason why German speakers are able to identify the missing verb but the English speakers are unable to do so.

In this paper, we extend the connectionist model of Christiansen and Chater (1999) to generate predictions for both the English and German structures, and demonstrate that this experience-based account provides a better explanation for the English and German data than an account based on language-independent working-memory constraints.

The Model

Network Architecture, grammar and corpora

We used a simple recurrent network (Elman, 1990) for modeling the effect of experience on forgetting. SRNs have been used previously to model the effect of structural properties in the language on comprehension performance (Christiansen & Chater, 1999; MacDonald & Christiansen, 2002). Since the predictions of an SRN are sensitive to probabilistic constraints in the input structure, they serve well to assess the effect of language-specific properties on learning. Furthermore, the architectural limitations of an SRN and its gradient nature give rise to human-like processing properties that have been explained in terms of working memory capacity limitations and decay in symbolic models. Our claim is that the grammaticality illusion is dependent on experience with word order regularities of the language in question. In order to show this we used a simple artificial language resembling simple sentences and subject- and object-extracted relative clauses. We also held the number of subject- and object-relatives equal in the corpus. In doing so we made sure that the only varying factor between the two training languages was whether its relative clauses are head-final or not.

The Corpora were generated from probabilistic context-free grammars (PCFGs) originally designed by Lars Konieczny (English) and Daniel Müller and Lars Konieczny (German).[1] For generating corpora and likelihood predictions the Simple Language Generator (Rohde, 1999) was used. Every training corpus consisted of 10,000 randomly generated sentences. Test corpora were generated for every condition consisting of 10 test sentences each. The networks described below were built, trained, and tested in the Tlearn simulator (Elman, 1992) on a Windows platform.

Training and Testing Procedure

Prior to training, all networks were initialized with random connection weights in the range of [-0.15, 0.15] and the hidden units received an initial bias activation of 0.5. Each training included 10 individually initialized networks that were trained on 10 different corpora, respectively. The networks were trained for three epochs, where one epoch corresponded to a full run through a corpus.

The SRNs were trained on a word-by-word continuation prediction. Each input word produced an activation distribution over the output nodes which represented lexical entries. In combination with a cross-entropy error calculation (all output activations sum to 1) the activation distribution was comparable to a probability distribution over words.

The SRN's prediction were assessed using grammatical prediction error (Christiansen & Chater, 1999). The GPE algorithm is based on the numerical differences between the PCFG probabilities and the actual output. The GPE value is a difficulty measure for every word in the sentence, which can be used as a reading time predictor (MacDonald & Christiansen, 2002).

Modeling the grammaticality illusion

The SRN trained on English sentences had 31 input and output units and 60 hidden units. Each input and output unit stood for one lexical entry in the lexicon. The lexicon consisted of five nouns, four intransitive and four transitive verbs in singular, plural and past tense forms and one end-of-sentence marker (EOS). At every NP the probability of an RC embedding was 0.1.[2] An RC could be realized as a subject relative (SRC) or an object relative clause (ORC) with equal probability.[3] Probabilities for transitivity and number status were also equal. The longest sentence in the corpus for English had 18 words. The German lexicon contained 21 words, including four verbs and nouns in singular and plural forms, the respective determiners in nominative and accusative, the comma and the EOS marker. In consequence the SRN trained on German had only 21 input and output units. The longest corpus sentence had 41 words, including the obligatory commas in German relative clauses. Both the English and German grammars included a number agreement between subjects and their predicates. In German a number and case agreement between determiner and noun was also included.

Christiansen and Chater (1999) reported node activations for the region after an NNNVV sequence. For better comparison with empirical data we extended their study to obtain GPE values for both conditions on all regions after the missing verb. Consider for example the error values on seeing V1 after the sequence 'N1 N2 N3 V3', which is ungrammatical because V2 is missing. In case the network is not aware of the ungrammaticality, this should be reflected by similar GPE values for both the grammatical and the ungrammatical condition at V1. In order to model that we set the target probability at V1 to the same value as in the grammatical condition. (Meaning the probability distribution is conditioned by the assumption that V2 has actually been seen.) In consequence, an expectation of a V2 at this point would increase the GPE. So, in the ungrammatical condition an SRN with a more accurate grammar representation would produce a higher pre-

[1] Both grammars can be found at http://cognition.iig.uni-freiburg.de/teaching/veranstaltungen/ws03/projekt.htm.

[2] These are the probabilities used by Konieczny in his grammar; MacDonald and Christiansen have used 0.05. The precise number is arbitrary; the essential point is that relative clauses should be less frequent than simple sentences.

[3] We did not encode the well-known difference in probability of occurrence between SRCs and ORCs because we were not modeling this difference; this assumption does not affect the results presented here.

diction error than an SRN wrongly predicting V1 instead of V2.

For the English case, the GPE values would be lower in the ungrammatical condition. This effectively means that the SRN is unable to make correct predictions based on long-distance dependencies, but bases its predictions on rather locally consistent sequences. For example after seeing V3 the network only predicts one more verb because the observation of N1 is too weakly encoded in the hidden representations to influence the predictions. Consequently, on V1 the error for the ungrammatical condition should be lower because in the grammatical condition V1 is the third verb which is inconsistent with the SRN's predictions. The preference for the ungrammatical structure should continue on the post-V1 regions because a locally coherent context with two verbs is easier to handle than a context of three verbs.

We first tested whether the SRN makes the same predictions as previous work on the English grammatical and ungrammatical structures (Christiansen & Macdonald, 2009).

Simulation 1: English

The SRN, which was trained on the English corpus, was tested on the grammatical and the ungrammatical condition after one, two, and three epochs.

The grammar we used was more complex than Christiansen and Chater's, but structurally compatible. Therefore we expected that we would replicate their findings for English. In particular, the GPE values for the V1 and post-V1 regions should receive lower values in the ungrammatical condition (see corpus example 2b).

(2) a. The judge that the reporters that the senators understand praise attacked the lawyers .

b. *The judge that the reporters that the senators understand attacked the lawyers .

Results for simulation 1 In order to compare the results for the English self-paced reading and eyetracking experiments in Vasishth et al. (2008) the assessed regions in the simulation were the three verbs V3, V2, V1 and the post-V1 region. The V2 region contains no datapoint in the ungrammatical condition because the verb is dropped in the testing stimuli.

Figure 1 shows GPE values for the SRNs trained and tested on the English grammar after one, two and three epochs of training. The pattern corresponded to the empirical results; the SRNs predicted an advantage for ungrammatical structures at V1 and post-V1. No effect was predicted on V3 because no difference in stimuli and probability between the conditions is present at this point.

Simulation 2: German

We turn next to the simulations for German center embeddings. German relative clauses differ from English in at least two respects (a third difference is the morphology of the relative pronoun; but we do not discuss this difference here due to space constraints). First, German relative clauses are obligatorily head final; second, commas are obligatory in German relative clauses (see 3 for an example). We return to the role of commas later in the paper.

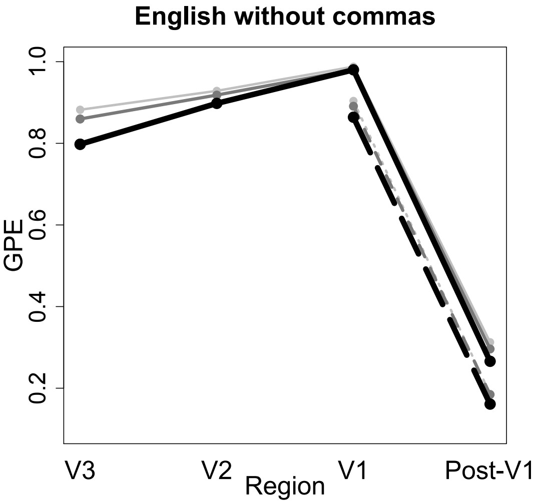


Figure 1: Simulation 1. English double-embedded object relative clauses. The figure shows the GPE values (for three epochs) for the three verbs and the subsequent region of the grammatical and ungrammatical conditions. The dotted line shows the ungrammatical condition. Epochs 3, 2, and 1 are colored black, dark grey and light grey, respectively.

(3) a. Der Polizist , den der Mensch , den der Passant verspottet , ruft , trifft den Jungen .

b. *Der Polizist , den der Mensch , den der Passant verspottet , trifft den Jungen .

Results of simulation 2 Figure 2 summarizes the findings. First, in the regions V2 and V1, the GPEs were lower compared to the English sentences. Second, in contrast to the English case, the comparison by conditions did not reveal any difference on the main verb (V1). Finally, a small but significant preference for the grammatical structure was found on the post-V1 region ($p < 0.001$).

Discussion

The English and German center-embedding simulations suggest that experience with head-final structures may furnish a better explanation for the grammaticality illusion in English (and its absence in German) than working-memory based accounts such as Gibson and Thomas'. Both the English and German reading patterns found in the literature can be modeled by the SRN, whereas the working-memory based explanation can only explain the English results.

Our results do not imply that working memory plays no role in these constructions; rather, our claim is that experience plays a dominant role. A plausible way to reconcile the two accounts into one composite theory would have experience modulating working-memory overload. These details are or-

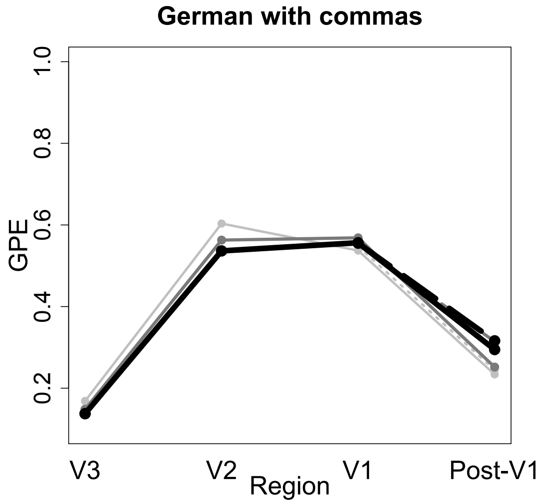


Figure 2: Simulation 2. German double-embedded object relative clauses. The figure shows the GPE values (for three epochs) for the three verbs and the subsequent region of the grammatical and ungrammatical conditions.

thogonal to our main finding, which is that experience determines whether English and German readers can correctly maintain predictions for upcoming verbs.

The role of commas in processing English center embeddings

One objection to this experience-based explanation for the grammaticality illusion (and its absence) is that the difference between English and German center embeddings could be related to the obligatory presence of commas in German. The commas in German relative clauses could lead to a strategy that is not available in the English structures previously studied. For example, readers could simply be counting the number of commas in German, and this could make it easier for them to detect ungrammaticality.

If commas alone (and not the head-final nature of relative clauses) are responsible for the patterns observed in German, then two straightforward predictions are that: (a) adding commas to English relative clauses should result in a German-like pattern for English sentences; and (b) removing commas from German relative clauses should result in an English-like pattern for German sentences.

Prediction (a) can be evaluated empirically but prediction (b) cannot because, as mentioned earlier, commas are obligatory in German relative clauses. As it turns out, Vasishth et al. (2008) tested the prediction for English and found that the presence of commas in English does not change the pattern; the grammaticality illusion persists.

The question we address next is: What does the SRN model predict for English RCs when commas are present?

Simulation 3: English with commas

For the simulation we extended the English grammar with appropriate comma insertions and trained the SRNs on the resulting corpora. In English non-restrictive object relative clauses (ORCs), commas would appear after nouns in the beginning of the sentence and after the verbs in the end. In a double-embedded ORC there would be a comma after V3 and V2. Thus, the grammatical/ungrammatical sequence pair is N,N,NV,V,V vs. N,N,NV,V. See (4) for examples.

For the SRN the comma effectively appears as a word category with only one token which attaches to nouns or verbs and is not involved in long-distance dependencies. Hence, the activation pattern representing it should not be too complex. In fact the learning of comma usage in ORCs can be reduced to a counting recursion problem of the pattern *aabb* instead of *abba*. As discussed in (Christiansen & Chater, 1999), counting recursion is the easiest of the three recursion types for both humans and connectionist networks. Thus, it is very likely that the inclusion of commas facilitates processing in the grammatical condition, lowering the respective GPE values.

(4) a. The lawyer , who the senator , who the judges attack , understands , praises the reporters .

b. *The lawyer , who the senator , who the judges attack , praises the reporters .

Results for simulation 3 See Figure 3 for the results after one, two and three epochs. Compared to simulation 1, there was a global improvement for both conditions, i.e., the GPEs were lower in each region. On V1 training had more effect in the ungrammatical than in the grammatical condition, resulting in a preference for the ungrammatical structure on V1 (as in simulation 1). On post-V1 training affected the grammatical condition more, however, not resulting in a grammaticality preference.

In summary, the SRN model suggests that although the insertion of commas in English helps to make better predictions overall, training effects seem to be driven by rather local consistency (Tabor, Galantucci, & Richardson, 2004), (Konieczny & Mueller, 2007), affecting the ungrammatical condition more than the grammatical one.

Importantly, the grammaticality illusion persists for English even when commas are present. This is consistent with the empirical findings for non-restrictive English relative clauses: Vasishth et al. (2008) also found in a self-paced reading study that the comma cue did not affect the grammaticality illusion in English.

The above findings raise an interesting question for German: is the reversal of the grammaticality illusion in German due only to the head-final nature of relative clauses, or do commas also play a role in determining the outcome? The only way to empirically disentangle the effect of head-finality and commas in German would be to examine a language such as Hindi, which also has head-final relative clauses but does not require commas.

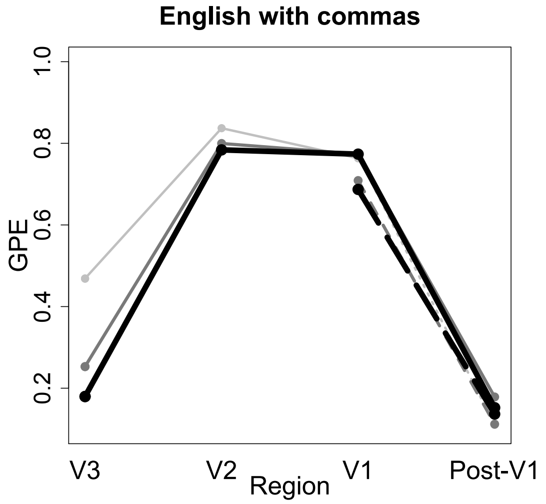


Figure 3: Simulation 3. The figure shows the GPEs (for the three epochs) of English center embeddings with commas.

Until such empirical evidence becomes available we cannot definitively answer the question about the role of commas, head-finality their interaction with experience. The SRN model can however generate predictions regarding the role of commas versus head-finality in German. We simulated the acquisition of experience with German head-final relative clauses which do not have any commas at all; in effect, we can simulate the learning of Hindi-type relative clauses in German. If commas are (partly) responsible for the reversal of the grammaticality illusion in German, then we should see an English-like pattern; if head-finality alone is the critical factor, then we should see a preference for grammatical structures even when commas are absent. This simulation is presented next.

Simulation 4: German without commas

In German, the presence of commas could have a facilitating effect because the counting-recursion pattern *aabb* is not only applicable in the ORC as in English but also in the SRC (both are head-final structures in German, unlike English). Consequently, the SRN trained on the German corpus should be very skilled on center-embedding recursion and comma counting-recursion and hence will have much lower error rates for the grammatical condition.

Thus, in German the removal of commas should make the SRN's predictions more error-prone. The verb-finality regularity in German, however, could still result in better predictions for the grammatical condition in German than in English. In order to test these predictions, simulation 4 tested SRNs trained on a comma-free German grammar.

Results of Simulation 4 The GPE values of the simulation involving German without commas (Figure 4) show a similar pattern as in English without commas. In the first epoch, an ungrammaticality preference was found in a small effect on V1 and a very pronounced effect on the region following it. After completion of training, V1 and post-V1 show a similar sized preference for the ungrammatical structure. Surprisingly, the regularity of verb-final structures does not seem to support correct predictions in German any more than in English. Rather, the more regular application of commas in German has a very facilitating effect on both conditions, slightly more on the grammatical.

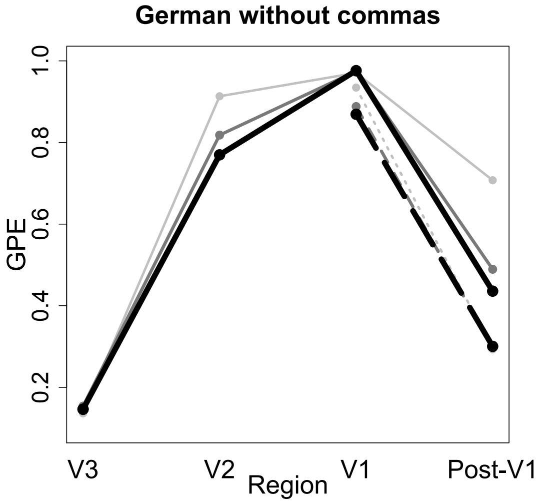


Figure 4: Simulation 4. The GPEs for German center embeddings without commas.

General Discussion

The results of simulation 1 (English without commas) and 2 (German with commas) were consistent with existing empirical data from both offline studies and online (self-paced reading and eyetracking) studies (Gibson & Thomas, 1999; Christiansen & MacDonald, 1999; Vasishth et al., 2008; Christiansen & Macdonald, 2009): the grammaticality illusion occurs in English but not in German.

These simulations demonstrate that the inherent architectural constraints of SRNs correctly predict both the grammaticality illusion in English double-embedded ORCs, as well as the absence of the illusion in German. In addition, the SRN model also makes the correct predictions regarding the effect of commas in English relative clauses: although commas reduce the GPEs, the grammaticality illusion persists in English. This is consistent with the evidence presented by Vasishth et al. (2008). Finally, we showed that in German head-finality alone does not explain the absence of the grammaticality illusion; commas appears to be crucial for the patterns observed.

Conclusion

This paper investigated the explanatory power of a particular implementation of the experience-based account for the grammaticality illusion. The well-known SRN modeling approach of MacDonald and Christiansen (2002), Christiansen and Macdonald (2009) was adopted to test its predictions on the forgetting effect in complex center-embedding.

The grammaticality illusion was predicted for English but not for German, consistent with human data. However, further simulations revealed the comma insertion as an important factor for the German pattern.

A caveat is necessary here. An SRN trained on a simple grammar obviously does not learn exactly the same constraints as humans do. These simulations are rather approximations that are suggestive of the role that experience plays in modulating memory processes. An important issue with the SRNs' predictions is their dependency on local coherence. Interestingly, however, there is evidence that even human readers rely on local coherence in certain structures (Tabor et al., 2004). Another finding is that the simulations reported by Christiansen and Chater (1999), and also the comma issue in simulations presented here, showed that the SRN handles counting-recursion better than other types. That may be the reason for the strong facilitating effect of comma insertion compared to head-finality.

More broadly, this work argues in favor of a uniform account of language-specific differences that are grounded in experience and that emerge as a consequence of architectural constraints. This account is broadly consistent with a range of recent work that characterizes processing modulated by experience (Hale, 2001). At the same time, it is clear that working-memory centered accounts capture a great deal of the empirical base that purely experience-based accounts cannot explain. Some examples are: the presence of both similarity-based interference and similarity-based facilitation effects (Logačev & Vasishth, 2009), the interaction of interference with locality (Van Dyke & Lewis, 2003) and with antilocality (Vasishth & Lewis, 2006). Thus, it appears that a principled composition experience as well as working-memory constraints is necessary to explain the range of empirical phenomena in sentence processing.

Acknowledgements

We are very grateful to Lars Konieczny for permission to use the grammar developed in his lab.

References

Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*(2), 157–205.

Christiansen, M. H., & Macdonald, M. (2009). *A usage-based approach to recursion in sentence processing.* (Submitted)

Christiansen, M. H., & MacDonald, M. C. (1999). *Processing of recursive sentence structure: Testing predictions from a connectionist model.* (Manuscript in preparation)

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Elman, J. L. (1992). *Tlearn simulator.* Software available at: http://crl.ucsd.edu/innate/tlearn.html.

Frazier, L. (1985). Syntactic complexity. In D. R. Dowty, L. Kartunnen, & A. M. Zwicky (Eds.), *Natural language parsing: Psychological, computational, and theoretical perspectives* (pp. 129–189). Cambridge University Press.

Gibson, E., & Thomas, J. (1999). Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, *14*(3), 225–248.

Hale, J. (2001). A probabilistic earley parser as a psycholinguistic model. *North American Chapter Of The Association For Computational Linguistics*, 1–8.

Konieczny, L., & Mueller, D. (2007). Local coherence interpretation in written and spoken language. In *Proceedings of the 20th annual CUNY conference on human sentence processing.* La Jolla, CA.

Logačev, P., & Vasishth, S. (2009). Morphological ambiguity and working memory. In P. de Swart & M. Lamers (Eds.), *Case, word order, and prominence: Psycholinguistic and theoretical approaches to argument structure.* Springer.

MacDonald, M. C., & Christiansen, M. H. (2002). Reassessing working memory: Comment on Just and Carpenter (1992) and Waters and Caplan (1996). *Psychological Review*, *109*(1), 35–54.

Rohde, D. L. T. (1999). The simple language generator: Encoding complex languages with simple grammars (Tech). *Mellon University, Department of Computer Science*, 99–123.

Tabor, W., Galantucci, B., & Richardson, D. (2004, May). Effects of merely local syntactic coherence on sentence processing. *Journal of Memory and Language*, *50*(4), 355–370.

Van Dyke, J. A., & Lewis, R. L. (2003). Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. *Journal of Memory and Language*, *49*(3), 285–316.

Vasishth, S., & Lewis, R. L. (2006). Argument-head distance and processing complexity: Explaining both locality and antilocality effects. *Language*, *82*(4), 767-794.

Vasishth, S., Suckow, K., Lewis, R., & Kern, S. (2008). *Short-term forgetting in sentence comprehension: Crosslinguistic evidence from head-final structures.* (Submitted to Language and Cognitive Processes)

First Steps Towards a Social Comparison Model of Crowds

Natalie Fridman, Gal A. Kaminka and Meytal Traub
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
{fridman, galk, traubm}@cs.biu.ac.il

Abstract

Modeling crowd behavior is an important challenge for cognitive modelers. Unfortunately, existing computational models are typically not tied to cognitive science theories, and are rarely evaluated against human crowd data. We investigate a general cognitive model of crowd behavior, based on Festinger's Social Comparison Theory (SCT). We evaluate the SCT model on general pedestrian movement, and validate the model against human pedestrian behavior. The results show that SCT generates behavior more in-tune with human crowd behavior then existing non-cognitive models. Moreover, we examine the impact of the different SCT model components on the generated pedestrian behavior.

Introduction

Modeling crowd behavior is an important challenge for cognitive science and psychology (Le Bon, 1895; Allport, 1924; Turner & Killian., 1972). Accurate models of crowd behavior are sought in training simulations, safety decision-support systems, traffic management, and organizational science. Indeed, a variety of computational models have been proposed that exhibit crowd-like behavior in different tasks. For instance, cellular automata models are used to model pedestrian movements (Blue & Adler, 2000; Helbing & Molnar, 1997) or people evacuating an area in emergency (Helbing, Farkas, & Vicsek, 2000; Kretz, 2007).

Unfortunately, only a handful of existing models of crowd behavior have been evaluated against real-world human crowd data. Moreover, essentially no computational cognitive models have been proposed which are tied to cognitive science theory. Instead, existing models are often inspired by particle physics (modeling individuals as particles), or by cellular automata. Thus fitting in the models with a deeper cognitive model of humans, or the mechanisms of a cognitive architecture, is difficult.

Recently, we presented a novel cognitive model of crowd behavior (Fridman & Kaminka, 2007), which has two key novelties (compared to previous models): First, there is a single computational mechanism (algorithm) used to generate different crowd phenomena (Fridman & Kaminka, 2009); and second, it is inspired by social psychology theory. In particular, the model is based on Social Comparison Theory (*SCT*) (Festinger, 1954), a popular social psychology theory that has been continuously evolving since the 1950s. The key idea in SCT is that humans, lacking objective means to evaluate their state, compare themselves to others that are similar.

We believe that social comparison is a general cognitive process underlying social behavior of each individual in crowd. Unlike previous crowd models that concentrate on specific behavior, the SCT model can account for different crowd behaviors, depending on the perceptions and actions available to each individual (Fridman & Kaminka, 2007). However, while the SCT model proved superior to other computational models in behaviors-specific measures (e.g., the formation of lanes in bidirectional movement), it was never validated against human crowd data.

In this paper we evaluate the SCT model on the specific task of general pedestrian movement which includes individuals, couples, and groups, all walking with different speeds, and in different directions. We contrast the performance of the model with a popular baseline model (Blue & Adler, 2000; Helbing et al., 2000), and explore the impact of different parameters and model components (e.g., bounds) on the generated behavior. The evaluation was carried out by 39 human subjects who compared the behavior generated from the different models to movies of real-world pedestrians. The results clearly justify the the particular parameters selected in earlier work (Fridman & Kaminka, 2007), and also demonstrate the SCT model is superior to others in its fidelity to human pedestrian behavior.

Background and Motivation

Social psychology literature provides several views on the emergence of crowds and the mechanisms underlying its behaviors. These views can inspire computational models, but are unfortunately too abstract to be used algorithmically. In contrast, computational crowd models often ignore cognitive and psychological processes underlying human behavior. Moreover, only a little work was done in validating computational models against data of human behaviors.

General crowd psychology. A phenomenon observed with crowds, and discovered early in crowd behavior research is that people in crowds act similar to one another, often acting in a coordinated fashion, which is achieved with little or no verbal communication. Moreover, the crowd may cause its members to behave differently than they would have individually. There are several different theories that explain this crowd characteristics, focusing on the cognitive process underlying each individual within the crowd.

Contagion Theory (Le Bon, 1895) emphasized a view of crowd behaviors as controlled by a "Collective Mind", and observed that an individual who becomes a part of the crowd is strongly affected by it, to the extent that she is transformed into becoming identical to the others in the crowd. Le Bon explains the homogeneous behavior of a crowd by two processes: (i) *Imitation*, where people in crowds imitate each other; and (ii) *Contagion*, where people in a crowd behave very differently from the way they usually do, individually.

On the other hand, Convergence Theory (Allport, 1924) states that crowd behavior is a product of the behavior of like-minded individuals. According to Allport's theory, individuals become a part of the crowd behavior when they have a "common stimulus" with people inside the crowd; for example, a common cause (Allport, 1924). Allport agrees with Le Bon (1895) about the homogeneous behavior of the crowd.

Turner and Killian (1972) investigated Emergent-norm Theory, which hypothesizes that crowd members indeed imitate each other, but also create new norms for the crowd as the dynamics of the situation dictate. Thus while crowds are not entirely predictable, their collective behavior is a function of the decision-making processes of their members.

Specific models. Researchers have developed computational models for simulation of collective behavior. However, these models are not often tied to cognitive processes underlying individual behavior in crowd and have rarely been validated against human data.

For instance, to simulate pedestrian movements, Blue and Adler (2000) use Cellular Automata approach, Helbing et al. (Helbing et al., 2000) focus on physical and social forces of attraction and repulsion that underlying each simulated entity. A common theme in all of them is the generation of behavior from the aggregation of many local rules of interaction. These models ignore cognitive theories of crowds.

There are several models that account for psychological and cognitive processes underlying agent behavior in crowd. For example, Yamashita and Umemura (2003), propose a model for panic behavior in which each agent acts based on its instincts such as escape instinct, group instinct and imitational instinct. Osaragi (2004) proposed a model for simulating pedestrian flow by using the concept of pedestrian mental stress which may increase or decrease as a result of density. However, these models only focus on cognitive processes underlying specific behaviors like flocking or evacuation and not account for general individual behavior in crowd.

One of the challenges in modeling crowd behaviors is the validation process. There is a great absence of human crowd behavior data that simulated models can be compared against. Only a handful of investigations have utilized experiments to validate computational models against human data.

For example, Kretz (2007) proposes the Floor field-and-Agent based Simulation Tool model (FAST) which is an extension of probabilistic cellular automata and discrete-space, discrete-time model for pedestrian motion. The FAST model has been validated against human data. In particular, the model simulation results of evacuation scenario was compared to results of evacuation exercise at a primary school.

Wolff (1973) examined pedestrian behavior in typical city block, and noted on the coordinated behavior of crowd, in term of creation of lanes in bidirectional movement or spread effect in unidirectional movement. However, in this experiment no quantitative data was presented. To learn more about pedestrian flows (density, speed), Daamen and Hoogendoorn (2003) performed empirical experiments on human crowds, in particular in terms of movement of pedestrians. However, these experiment focused only on the movement of independent individuals, rather than families or friends.

Our long-term goal is to provide a single cognitive mechanism that, when executed by individuals, would give rise to different crowd behaviors, depending on the perceptions and actions available to each individual. In previous work (Fridman & Kaminka, 2007), we presented such a mechanism, based on Social Comparison Theory. The model was evaluated on specific pedestrian movement phenomena, such as creation of lanes in bidirectional movement; it was not evaluated against human pedestrian movement.

A Model of Social Comparison

Our research question deals with the development of a computerized cognitive model which, when executed individually by many agents, will cause them to behave as humans do in groups and crowds. We build on earlier work on the SCT crowd model, briefly described below; the interested reader is referred to (Fridman & Kaminka, 2007) for details.

According to social comparison theory, people tend to compare their behavior with others that are most like them (Festinger, 1954). To be more specific, when lacking objective means for appraisal of their opinions and capabilities, people compare their opinions and capabilities to those of others that are similar to them. They then attempt to correct any differences found.

Translated into an algorithm, we take each observed agent to be modeled by a set of features and their associated values. For each such agent, we calculate a similarity value $s(x)$, which measures the similarity between the observed agent and the agent carrying out the comparison process. The agent with the highest such value is selected. If its similarity is between the given bounds (S_{max} and S_{min}), then this triggers actions by the comparing agent to reduce the discrepancy. The upper bound (S_{max}) prevents the agent from trying to minimize differences with someone who is already sufficiently similar, since such differences are not meaningful. The lower bound S_{min} filters agents that are too dissimilar, and so should be ignored. Thus, within the bounds an agent compares itself with those that differ from it sufficiently to matter. In experiments, we examine the impact of SCT bounds on the generated simulated behavior.

To reduce discrepancy, we determine the list of features f_i that indicate a difference with the selected agent c. We order these features in an increasing order of weight w_i, such that the first feature to trigger corrective action is the one with the least weight. The reason for this ordering is intuitive, and we admittedly did not find evidence for it in the literature. However, in this paper we examine the impact of the correction order on the quality of the simulated behavior.

```
1. For each known agent $x$ calculate similarity $s(x)$
2. $c \leftarrow \texttt{argmax}\, s(x)$, such that $S_{min} < s(c) < S_{max}$
```

3. $D \leftarrow$ differences between me and agent c

4. Apply actions to minimize differences in D.

To implement final step of the algorithm, we assume that every feature has associated corrective actions that minimize gaps in it, to a target agent, independently of other features. Festinger writes (Festinger, 1954, p.131): "The stronger the attraction to the group the stronger will be the pressure toward uniformity concerning abilities and opinions within that group". To model this, we use a gain function *Gain* for the action *o*, which translates into the amount of effort or power invested in the action. For instance, for movement, the gain function would translate into velocity; the greater the gain, the greater the velocity.

$$Gain \equiv \frac{S_{max} - S_{min}}{S_{max} - s(c)} \qquad (1)$$

Validation Against Human Data

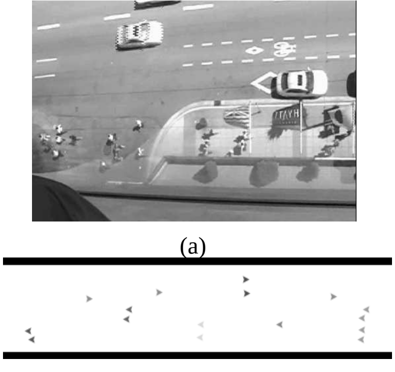
The SCT model was previously evaluated separately on different crowd behaviors (Fridman & Kaminka, 2007). In particular, different types of pedestrian movement phenomena (such as creation of lanes in bidirectional movement of individuals, movement in small groups with and without obstacles, etc.). When evaluated on such specific behavior, it is possible to use community-recognized standard measures, such as flow, number of lane changes, etc. However, when evaluating the model against human data, it must account for a fuller set of behaviors, all mixed together. For example, when watching pedestrians, we can observe people moving as groups like family, friends and couples or as individuals, all walking with different speeds in bidirectional fashion.

A different evaluation methodology is thus needed. One of the greatest challenge in modelling crowd behaviors is the great absence of human crowd behavior data that can be used as a basis for comparison. The main difficulty in creation of such data is that controlled experiments are complex to design, and costly to execute, since they have to be in large scale. There does not exist a standard methodology of evaluation; some researchers generate accurate behavioral data by engaging crowds in virtual environments (Pelechano, Stocker, Allbeck, & Badler, 2008), while others do qualitative comparisons of their models' predictions against movies of crowds, i.e., via observation experiments, e.g., (Helbing et al., 2000; Kretz, 2007). We follow the same approach. Below, we describe the observation experiments we executed to evaluate the SCT model on general pedestrian behavior.

Comparing to Human Behavior

In this experiment we focus on general pedestrian behavior where individuals and small groups (e.g., family and friends, couples) walk with different speeds in bidirectional fashion. Our hypothesis is that generating pedestrian behavior with SCT model is more in tune with human pedestrian behavior, compared to other models from the literature. We also want to examine the impact of the model components (bounds, correction order, gain) on the quality of the simulated behavior.

We used human crowd movies where different pedestrian behavior phenomena are presented (Figure 1(a)) and created screen-capture movies of different models of the same behavior (Figure 1(b)). We rely on experiments with human subjects which compare each of the resulting simulated behaviors to human crowd behavior. In addition, the subjects also voted for the most similar and dissimilar simulated behavior.


(a)

(b)

Figure 1: **Real (a) and Simulated (b) Pedestrian Behavior.**

Simulated Behavior: Experiment Setup. To simulate pedestrian behavior, we used Net-Logo. We define a sidewalk with 104 patches in length and 10 patches at width. To fit to human crowd density, the sample population comprised 30 agents. Agents were able to move in a circular fashion from east to west or in opposite direction with different speeds. Agents that belong to the same group have the same color. In order to create small groups, couples and individuals, we define our population with 15 different colors (a large number considering the population size). Agents were placed in random positions at the beginning of the experiment, each agent had limited vision distance of 10 patches and cone-shaped-field-of-view of 120 degrees.

Each agent has a set of features and their corresponding weights. For simulating pedestrian movement, we used the following features and weights: *color* (weight 3); *Walking direction* east or west (weight 2); and *position* (weight 1), given global coordinates. To account for the western cultural intuition that friends (and family) walk side-by-side, rather than in columns, we used another feature: The similarity in position along the x-axis - *X-Coordinate* (weight 0.5).

The rationale for feature priorities, as represented in their weights, follows from our intuition and common experience as to how pedestrians act. Positional difference (distance, side-by-side) is the easiest difference to correct, and the least indicative of a similarity between pedestrians. Direction is more indicative of a similarity between agents, and color (which we use to denote sub-groups within the crowds) even more so. For instance, if an agent sees two agents, one in the same direction as it (and far away), and the other very close to it (but in the opposite direction), it will calculate greater similarity to the first agent, and try to minimize the distance to it (this may cause a lane change) and only then try to locate itself on the same X-coordinate.

The similarities in different features (f_i) are calculated as follows. $f_{color} = 1$ if color is the same, 0 otherwise. $f_{direction} = 1$ if direction is the same, 0 otherwise, $f_{distance} = \frac{1}{dist}$, where *dist* is the Euclidean distance between the positions of the agents and finally, $f_{x-coordinate} = 1$ if x-coordinate is the same, 0 otherwise. Each agent calculates $s(x)$ according to the model. If the chosen feature for closing the gap is distance, then the velocity for movement will be multiplied by the calculated gain *Gain*. For other features (which are binary), the gain is ignored.

We wanted to examine the impact of the SCT model components on the quality of the simulated pedestrian behavior. In particular, we wanted to examine the impact of SCT bounds (S_{min} and S_{max}), gain function, and correction order on the generated behavior. We define seven models, each emphasizing a different SCT component. The models are explained below, and summarized in Table 1.

First we wanted to examine the impact of SCT bounds on the generated pedestrian behavior. We hypothesize that more narrow bounds will provide more similar behavior to individual model. To examine this hypothesis, we define the following models:

- SCT-B-2-6.5 We set S_{max} to 6.5 (practically: no agent too similar) and S_{min} to 2 (which means that agents that differ only in distance and in X-axis are not consider similar). The gain is calculated according to Eq. 1 and the correction order is from the low weight features (distance) to high weigh features. In this domain agents cannot change their color, thus, the last corrected feature is direction. Our hypothesis that this model will provide most similar behavior to human pedestrians.
- SCT-B-5-6.5 We set the S_{min} to 5 which mean that agents that similar at least in color and direction are consider to be similar. Thus, in this model only agents with same color and direction will move together.

Another component that we want to examine is the impact of correction order on simulated pedestrian behavior. In the SCT-H-L model we define the correction order to be from high to low. Our agents cannot change their colors, and in this model if the selected agent is moving in opposite direction, the agent will first change it direction and then will try to close the distance gap.

Finally, we wanted to evaluate the importance of the gain in the model. We define the following models:

- SCT-NoGain Defined to be without the gain function (i.e., gain is constant 1).
- SCT-G-C2 The gain function is constant (2).
- SCT-G-C3 The gain function is constant (3).
- SCT-G-C4.5 The gain function is constant (4.5).

The various SCT models are contrasted with the *individual choice* model, commonly used in pedestrian crowd research (Blue & Adler, 2000; Helbing et al., 2000). In the individual model, when forward movement of an agent is blocked, an agent will arbitrary chooses different lane. Each agent make its decisions independently of its peers. This model has been shown to be qualitatively compatible with pedestrian motion, and is often used as a baseline technique in crowd research (see, for instance, (Kretz, 2007)).

Comparison to Human Crowd. In order to compare to general behavior and not to be connected to specific video clip, we used several video clips of human pedestrian behavior and several screen-captured movies for each model. In the simulated behavior we created three screen-captured movies for each model that was randomly chosen for each subject. In human behavior we used two sets of video clips that were taken from different locations and in different times. The first set of movie clips were taken in the morning in downtown Vancouver, during rush hour. People are mostly walking individually, and only few are moving in small groups. The second set of movie clips were taken in the afternoon in a street that leads to the Eiffel tower in Paris, during leisure time. Most of the pedestrians are families and friends that move in small groups, or as couples. Each real-world video clip was cut to be one minute long. To generate a one-minute clip in the simulated behaviors, each model was executed for 5000 cycles (6 minutes), and the last minute was used.

We build a web based experiment which enables the subjects to participate in their free time. First we presented a brief description about the experiments. The subjects were told that the purpose of the experiment is to compare each of the simulated behaviors to human crowd behavior. However, the purpose of the simulation is not to simulate each seen pedestrian in the human crowd, but to simulate the general pedestrian behavior. The experiment was carried out in two phases, a training phase that was presented to the subjects after the experiment description, and an experiment phase.

The experiment was carried out using 39 adult subjects (males: 28). Additional 6 subjects were dropped due to technical reasons (such as network problems that prevented them from watching the clips). The subjects were ask to watch the human pedestrian movie that was randomly chosen in each experiment. Then, they were ask to watch screen-captured movie of each model that was also chosen randomly. After each simulated movie, the subjects were ask to rank the seen behavior, that followed by question: To what degree the seen simulated behavior is similar to previously seen human behavior? (1—not similar, 6—most similar). At the end of the experiment, we ask the subjects additional two questions: What simulated movie was the most similar to human behavior and what simulated movie was the most dissimilar. To control for order effects, the order of presentation on the page was randomized.

Initially we wanted to compare eight different simulated behaviors to human pedestrian behavior, the individual choice model and seven SCT models. We run a short pilot in which we presented to three subjects the experiment and afterwards ask their opinion. All subjects claimed that the experiment was too long. Moreover, they claimed that SCT-B-2-6.5 model provide very similar behavior to that of SCT-

Component	**SCT-B-2-6.5**	**SCT-B-5-6.5**	**SCT-H-L**	**SCT-NoGain**	**SCT-G-C2**	**SCT-G-C3**	**SCT-G-C4.5**
Smax	6.5	6.5	6.5	6.5	6.5	6.5	6.5
Smin	2	5	2	2	2	2	2
Gain	Eq. 1 (func.)	Eq. 1 (func.)	Eq. 1 (func.)	1 (const)	2 (const)	3 (const)	4.5 (const)
Correction Order	L-H	L-H	H-L	L-H	L-H	L-H	L-H

Table 1: **SCT Models**

H-L model and similar behavior was also observed in models SCT-NoGain, SCT-G-C2, SCT-G-C3 and SCT-G-C4.5. Thus, we reduced the number of different models that presented to the subjects. In the experiment phase we compared between four simulated behaviors. We used the Individual-choice model, SCT-B-2-6.5, SCT-B-5-6.5 and one of randomly chosen SCT-NoGain, SCT-G-C3 and SCT-G-C4.5 models. The models SCT-H-L and SCT-G-C2 were used only in the training phase, and their results were not used.

Results

We first wanted to examine the ranking of the models in comparison to the actual crowd. The results are summarized in Figure 2. The categories in the X-axis correspond to different models. The Y-axis correspond to grades of the compared models. Each set of bar shows the mean and median results. A higher result indicates improved fidelity, i.e., greater similarity to human pedestrian behavior.

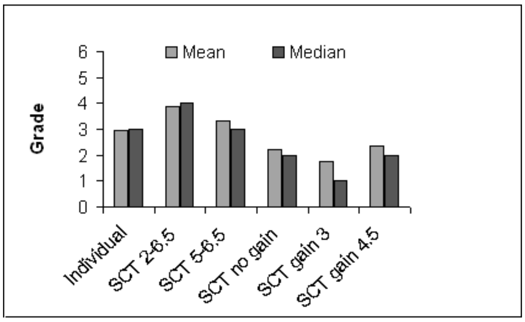


Figure 2: **Comparing to human pedestrian - Results**

The results clearly demonstrate that the SCT-B-2-6.5 model provide most higher results than the compared models. While it may seem that the SCT-B-2-6.5 model results is close to Individual and SCT-B-5-6.5 models results, according to t-test (two-tailed) SCT-B-2-6.5 was found to be significantly different than the Individual model ($p = 0.001$) and significantly different than SCT-B-5-6.5 ($p = 0.03$).

Another hypothesis underlying the experiment is that SCT model with narrower bounds (S_{min}, S_{max}) will provide closer behavior to individual model behavior, but not the same. Indeed, the results demonstrate that SCT-B-5-6.5 is lying in between the SCT-B-2-6.5 and individual models. According to t-test (two-tailed) SCT-B-5-6.5 was found to be significantly different than SCT-B-2-6.5 ($p = 0.03$) and significantly different than the Individual model ($p = 0.017$).

Our last hypothesis was that SCT models without the gain function will provide less similar behavior to human pedestrian behavior. The results clearly demonstrates that SCT-NoGain, SCT-G-C3 and SCT-G-C4.5 models in which the gain is fixed, get the lowest results.

When we ask the subjects: "What simulated behavior was the most similar to human behavior?" The SCT-B-2-6.5 model gets the highest number of votes. To the question: "What simulated behavior was the most dissimilar to human behavior?", the subjects answered with the SCT-NoGain, SCT-G-C3 and SCT-G-C4.5 models. The answers to these two questions are shown in Figure 3.

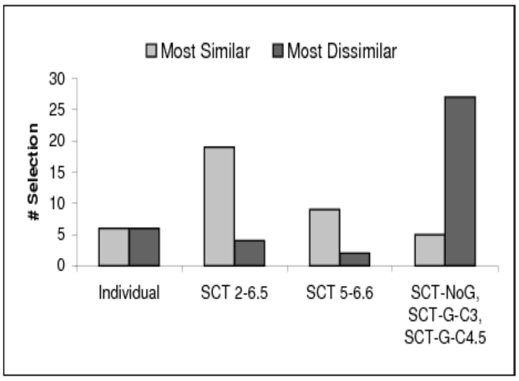


Figure 3: **Most similar/dissimilar: Results.**

Discussion

The SCT model, described and evaluated above, stands on two conceptual cognitive science legs. First, it draws a connection between social comparison theory and crowd behavior. Second, it interprets social comparison theory as admitting superficial comparisons, i.e., at the level of visible differences between agents, in addition to cognitive differences (e.g., intentions). We address these two issues below.

Social Comparison in Crowds. To the best of our knowledge, social comparison theory has never been connected to crowd behavior phenomena. However, we believe that social comparison theory may account for some important characteristics of crowd behavior, as it clearly addresses processes in groups, and no limit is placed on group size.

We focus here on one of the primary characteristics of crowds is the similarity between individuals' behaviors. This is explained by a process of *imitation* (Le Bon, 1895), convergence of like-minded individuals (Allport, 1924), or emerging norms (Turner & Killian., 1972).

Social comparison processes can give rise to this phenomenon. Festinger writes (1954, p. 124): "The existence of a discrepancy in a group with respect to opinions or abilities will lead to action on the part of members of that group to reduce the discrepancy". Indeed, one implication of SCT is the formation of homogeneous groups. Festinger notes (1954, p. 135):"The drive for self evaluation is a force acting on persons to belong to groups, to associate with others. People, then, tend to move into groups which, in their judgment, hold opinions which agree with their own“. This quote, in particular, seems to be compatible with (Allport, 1924).

Do people engage in surface comparisons? Festinger hypothesizes (Festinger, 1954, p. 117): "There exists, in the human organism, a drive to evaluate his opinions and his abilities". Thus a question that emerges with respect to the mechanisms described here is whether in fact the type of surface comparisons are admitted by social comparison theory.

There has been extensive research clarifying the concepts "abilities" and "opinions". Smith and Arnkelsson (2000) explain that ability evaluation refers to person performance at specific task. Festinger itself provide a link between ability and performance: "abilities are of course manifested only through performance which is assumed to depend upon the particular ability" (1954, p. 118). He then provide an example: “Thus, if a person evaluates his running ability, he will do so by comparing his time to run some distance with the times that other persons have taken.” (1954, p. 118).

Moreover, the meaning of opinion comparison, was also extensively investigated during the years. Goethals and Darley (1977) relate this concept to "Related Attributes Hypothesis" meaning people will prefer to compare with others similar to them on attributes that are related to their opinion or performance. Festinger provide the basis for this research claiming: "If persons who are divergent from one’s own opinion or ability are perceived as different from oneself on attributes consistent with the divergent, the tendency to narrow the range of comparability becomes stronger" (1954, p. 133). Goethals and Klein provide an example which directly admit surface comparisons: "An individual evaluating his or her tennis-playing ability. He or she might compare with others who are about the same age, who have the same degree of recent practice and comparable equipment, and who are the same sex" (Goethals & Klein, 2000, p. 25).

Summary

SCT is a cognitive model proscribing crowd behavior, inspired by Festinger’s social comparison theory (Festinger, 1954). A key novelty in SCT is its promise of domain-generality. However, while SCT has been evaluated against existing models in specific tasks, it was not validated against human crowd data.

This paper presented validation of SCT model (and competing models) against human crowd behavior. We evaluate the SCT on pedestrian phenomena and showed that SCT model generated pedestrian behavior more in tune to human pedestrian behavior. The results are promising, and support the general applicability of the SCT model. We are currently exploring the use of SCT in this and other domains.

Acknowledgements. This research was supported in part by EOARD Grant #083065, and by IMOD.

References

Allport, F. H. (1924). *Social psychology*. Boston: Houghton Mifflin.

Blue, V. J., & Adler, J. L. (2000). Cellular automata microsimulation of bidirectional pedestrian flows. *Transportation Research Record*, 135–141.

Daamen, W., & Hoogendoorn, S. P. (2003). Experimental research of pedestrian walking behavior. *Transportation Research Record*, 20–30.

Festinger, L. (1954). A theory of social comparison processes. *Human Relations*, 117–140.

Fridman, N., & Kaminka, G. A. (2007). Towards a cognitive model of crowd behavior based on social comparison theory. In *AAAI-07*.

Fridman, N., & Kaminka, G. A. (2009). Comparing human and synthetic group behaviors: A model based on social psychology. In *ICCM-09*.

Goethals, G. R., & Darley, J. (1977). Social comparison theory: An attributional approach. In *Social comparison processes: Theoretical and empirical perspectives.* Washington, DC: Hemisphere.

Goethals, G. R., & Klein, W. M. P. (2000). Interpreting and inventing social reality: Attributional and constructive elements in social comparison. In *Handbook of social comparison: Theory and research.* New York: Plenum.

Helbing, D., Farkas, I. J., & Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, *407*, 487–490.

Helbing, D., & Molnar, P. (1997). Self-organization phenomena in pedestrian crowds. In F. Schweitzer (Ed.), *Self-organization of complex structures: From individual to collective dynamics* (pp. 569–577). London: Gordon and Breach.

Kretz, T. (2007). *Pedestrian traffic: Simulation and experiments*. Unpublished doctoral dissertation, Universität Duisburg-Essen.

Le Bon, G. (1895). *The crowd: A study of the popular mind.* (1968 ed.). Dunwoody, Ga., N.S. Berg.

Osaragi, T. (2004). Modeling of pedestrian behavior and its applications to spatial evaluation. In *AAMAS-04* (pp. 836–843).

Pelechano, N., Stocker, C., Allbeck, J., & Badler, N. (2008). Being a part of the crowd: Towards validating VR crowds using presence. In *AAMAS-09.*

Smith, W. P., & Arnkelsson, G. B. (2000). Stability of related attributes and the inference of ability through social comparison. In *Handbook of social comparison: Theory and research.* New York: Plenum.

Turner, R., & Killian., L. M. (1972). *Collective behavior* (1993, 4th ed.). Prentice Hall.

Wolff, M. (1973). Notes on the behaviour of pedestrians. *In People in Places: The Sociology of the Familiar*, 35–48.

Yamashita, K., & Umemura, A. (2003). Lattice gas simulation of crowd behavior. In *Proceedings of the international symposium on micromechatronics and human science* (pp. 343–348).

Comparison of Instance and Strategy Models in ACT-R

Cleotilde Gonzalez[1] (coty@cmu.edu)
Varun Dutt[1] (varundutt@cmu.edu)
Alice F. Healy[2] (alice.healy@colorado.edu)
Michael D. Young[2] (mdyoung@psych.colorado.edu)
Lyle E. Bourne, Jr.[2] (lyle.bourne@colorado.edu)
[1]Dynamic Decision Making Laboratory
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[2]Center for Research on Training
University of Colorado, Boulder, CO 80309-0345, USA

Abstract

This paper presents a comparison of two models, built on the same architecture, ACT-R, and on the same dynamic decision making task, RADAR. The two models represent the Strategy-Based Learning (SBL) approach and the Instance-Based Learning (IBL) approach. The SBL approach assumes a certain set of predefined strategies, and learning occurs by selecting the most successful strategy over time. The IBL approach proposes that decisions are made based on retrieval of good past experiences stored in memory. This approach assumes no previous initial experience apart from that gained while performing the task. Both models were tested with respect to two criteria: fit to human data during a training exercise with RADAR and adaptability to test conditions that are either similar to or different from the training conditions. Our comparison results demonstrate that both models fit learning human data successfully, but the IBL model is more robust than the SBL model. This exercise initiates a discussion of the SBL and IBL approaches to modeling choice and decision making in ACT-R and a reevaluation of how to compare and assess computational models.

Keywords: dynamic decision making; instance-based learning; strategy-based learning; consistent mapping; varied mapping; ACT-R.

Introduction

In cognitive psychology there have been at least two views of the world: that humans understand the world by means of rules and by particular domain-related events (Nisbett, 1993). In cognitive modeling these same two views are often reproduced in the behaviorism and connectionism debate (Anderson & Lebiere, 2003). The debate in the late 1980s led to an opposition between the two modeling approaches, in which connectionism was perceived to resemble the underlying neural structure better than did behaviorism, a focus on learning from environmental stimuli rather than from generic rules, and a focus on subsymbolic manipulations rather than symbolic representations. In reality the two approaches have more in common than what was recognized in this debate.

ACT-R is a hybrid architecture composed of both symbolic and subsymbolic aspects (Anderson & Lebiere, 1998, 2003). The symbolic aspects are declarative and procedural. The declarative knowledge is represented in chunks, and the procedural knowledge is represented in productions (if-then rules). The subsymbolic elements of ACT-R are the neural-like statistical and mathematical mechanisms that manipulate the symbolic representations.

ACT-R allows for two different approaches to modeling human behavior that are particularly relevant for decision making and learning: the Strategy-Based Learning (SBL) and the Instance-Based Learning (IBL) approaches.

The SBL approach is the most popular approach to modeling choice and decision making in ACT-R (Lovett, 1998). Under this approach, modelers determine the strategies by which humans perform a task, and they represent these strategies in the form of production rules. Choice among competing production rules is controlled by the ACT-R subsymbolic utility learning mechanisms. Each production has a utility value that represents the rule's probability of success and the costs involved in reaching the goal. The utility learning mechanism produces a gradual switch from less successful to more successful strategies over time.

The IBL approach, although less popular, has been used successfully in representing decision making, mostly in dynamic situations (Dutt & Gonzalez, 2008; Gonzalez, Lerch, & Lebiere, 2003). Under the IBL approach, modelers determine the representation of declarative knowledge (chunks) in a task and represent a generic decision making process in production rules. This approach has been the basis for the development of a theory of decision making in dynamic tasks, called Instance-Based Learning Theory of Dynamic Decision Making, which provides IBL models with a generic decision making process (Gonzalez et al., 2003).

The main learning in this approach occurs at the declarative rather than the procedural level, where actions are based on the storage and retrieval of similar chunks in and from memory. Selection among chunks is based on ACT-R's activation subsymbolic learning mechanisms. Each chunk has a value of activation determined by a number of factors including the recency and frequency of use of that chunk. For example, recency and frequency of usage of a chunk determine the base-level activation, which represents the probability that a chunk is needed. The activation is also modulated by the degree to which a chunk matches the retrieval cues, with chunks encoding similar situations to the current one receiving some activation.

Over time, an IBL model transitions from the use of a general heuristic to the use of instances, as determined by the number of instances stored and the similarity of the situations confronted in the task (Gonzalez et al., 2003).

This paper presents a comparison of two models, IBL and SBL models, both interacting with the same real-time decision making task, and both developed under the same architecture (ACT-R). This effort differs from other model comparison efforts in that other model comparisons are often done to evaluate different "architectures" and often aimed at determining the "winning" model (Anderson & Lebiere, 2003; Cassimatis, Bello, & Langley, 2008). By comparing two different modeling approaches that represent decision making behavior in the same task and in the same architecture, we highlight the real value of model comparison: to understand the processes by which behavior is represented, the constraints that the different approaches impose upon the task models, and the comparison of the theoretical assumptions of the two approaches (Lebiere, Gonzalez & Warwick, 2009). The models interacted in real-time with a dynamic decision making task called RADAR (Gonzalez & Thomas, 2008).

We compared the SBL and IBL models according to two different dimensions: (1) fit: how well each model fits human learning data in the task; and (2) adaptability: how well each model is able to reproduce the way humans having learned in one scenario of the task behave in a testing condition, in scenarios that are similar to or different from the training condition. The fit criterion is common in model comparisons, whereas the adaptability criterion is relatively new (Gluck, Bello, & Busemeyer, 2008). The adaptability criterion we use here is similar to the generalization criterion method (Busemeyer & Wang, 2000), which divides observed data into two sets: a calibration or training set to estimate model parameters and a validation or test set to determine predictive performance. However, we further test the adaptability of our models by examining the models' ability to adapt to test conditions that are either similar to or different from the training conditions.

Experiment on the RADAR Task

The task used for this modeling effort is a dynamic visual detection and decision making task that has been used in past research to study automaticity (Gonzalez & Thomas, 2008) and training principles (Young, Healy, Gonzalez, & Bourne, 2007). The task, called RADAR, is described in detail by Gonzalez and Thomas (2008), and thus here we only summarize the relevant elements.

The goal in RADAR is to detect and eliminate hostile enemy aircrafts by visually discriminating moving targets among moving distractors in a radar screen. RADAR is similar to military target visual detection devices, in which a moving target needs to be identified as a potential threat and a decision is made on how to best destroy the target. The task has two components: (a) visual and memory search and (b) decision making. The visual and memory search component requires the participant to memorize a set of targets and then look for the presence of one or more targets on a radar grid. A target threat may or may not be present among a set of moving blips that represent incoming aircraft. The blips—in the form of digits, consonants, or blank masks—begin at the four corners of the radar grid and approach the center at a uniform rate. The detection of an enemy aircraft must occur before the blips collapse in the middle of the grid. This is the main component used in the experiment described below. The decision-making component is not relevant for this human experiment.

General Experimental Methods

Forty-eight participants at the University of Colorado, Boulder were asked to interact with RADAR to respond as quickly as possible to target letters or digits occurring among distractor letters or digits. In addition to target detection, participants were required to count deviant tones (low and high frequency) among standard tones (medium frequency) that played in the background during the target detection task. The experiment consisted of a training session and a test session with a 1 week-delay between the two sessions. Half the participants trained with both the tone-counting task and the target detection task and half performed the target detection task in silence. At test, half resumed their training condition and half switched.

There were 8 blocks during training and 8 blocks during testing, each consisting of 160 total trials. A trial is a group of 7 frames (RADAR screen and individual attempt to detect a target). A memory set of 1 or 4 possible targets was shown to participants prior to starting a trial. At most 1 frame within each trial contained a target. Each frame included either 1 or 4 non-blank blips among which there could be one target and zero or more distractors in the 7 frames of a trial. Targets and distractors were consistently mapped (CM: a target in the memory set never appeared as a distractor within a block) or varied mapped (VM: a target in memory set could appear as a target in one trial and as a distractor in another trial of a block).

Half the participants saw digits as the targets on CM trials. For these participants, letters were the distractors on CM trials and were both the targets and distractors on VM trials. The remaining participants saw letters as the targets on CM trials. For them, digits were the distractors on CM trials and were both the targets and distractors on VM trials. There were 9 integers 1 to 9 and 9 consonants C, D, F, G, H, J, K, L, M used as targets or distractors.

The 160 trials were divided into two session halves, each with 4 blocks (i.e. 80 trials), separated by a 5-min break. Blocks varied by mapping and processing load (number of items in the memory set and number of blips in each trial) condition. The four blocks in each session half included one of each combination of mapping condition and processing load (CM 1+1, VM 1+1, CM 4+4, VM 4+4). For the first session half these conditions occurred in the order CM 1+1, CM 4+4, VM 1+1, VM 4+4. For the second session half these conditions occurred in the reverse order VM 4+4, VM 1+1, CM 4+4, CM 1+1. Thus, the average block position was the same for each condition across session halves.

We use correct detection time (in ms) as the dependent variable. Results are presented in a later section, where they are compared to the results from the IBL and SBL computational models.

Instance-Based Learning Model

The IBL model was based upon the Instance-Based Learning Theory (IBLT) and other IBL developments (Gonzalez et al., 2003). IBLT was originally developed as a way to explain and predict decision making in dynamic, complex tasks (Dutt & Gonzalez, 2008; Gonzalez et al., 2003). For the RADAR task an instance (referred to as a chunk in ACT-R) had the structure shown in Table 1.

Table 1: Structure of an Instance in RADAR

Slot Name	Description	Chunk
Blip-Situation	Value of Blip	Situation
Decision	Spacebar Press	Decision

The *Blip-Situation* slot corresponded to the blip value (letter or number) occurring on the RADAR screen in one of the north-west, north-east, south-west, or south-east locations, respectively at a time. In the case of 1+1 trials, three out of the four slot locations contained a NIL value. For the purpose of linear similarity calculations (discussed later), the nine consonants were numbered from 10 to 18. The *Decision* slot refers to the act of pressing or not the spacebar. Although typically instances have a *Utility* slot to categorize an experience as good or bad in a situation after the IBL model gets feedback, in this model, due to the task's trial structure and the trivial feedback, we did not use such a slot.

As per Gonzalez et al. (2003), the IBL starts with the recognition process in search for alternatives and the classification of the current situation as *typical* or *atypical*. A situation is typical if there are memories of similar situations (i.e., instances of previous trials that are similar enough to the current situation). If it is typical then the retrieved instance is used in judging the value of the decision to be made in the current situation. If the situation is atypical (i.e., no instance similar to the current conditions is found in memory), a judgment heuristic is applied (in the present case, the heuristic is "wait for next blip"). When a decision point comes into place at one of the four blip positions, NW, NE, SW, and SE, a choice has to be made whether to search for more alternatives or to execute the current best alternative. In the RADAR task, the choice is simply made by seeing if the retrieved instance is similar enough to the one of the current blip situations (in case nothing was retrieved or the instance that was retrieved did not equal the current blip situation, then a choice is made to wait for the next blip situation and not to press the spacebar key, i.e. by a "wait for next blip" judgment heuristic). Thus, if something was retrieved from declarative memory, then the decision is to press the spacebar only if the retrieved instance is exactly the same as the current blip situation.

Before the IBL process starts for each frame's blips in a trial, the IBL model notices a set of target letters or numbers at the beginning of the trial in memory set and stores them in its declarative memory. Also the IBL process moves from one blip situation to another applying the process described below to each filled-in blip situation. The pattern of traversal between blip situations forms a Z (i.e., NW, NE, SW, and SE, respectively) until the frame time of 2.062 s runs out. If the IBL model cannot process all the filled-in blips before the frame time runs out, then it resets and starts at the NW filled-in blip for the next frame. Each of the IBL stages suggested in the IBLT (Gonzalez et al., 2003) is represented by production rules (if-then rules) in ACT-R:

Recognition On a trial if there is a recognition (or retrieval) failure or if the retrieved blip does not match the current situation blip, then apply the "wait for next blip" heuristic; otherwise if there is a recognition (or retrieval) success and a match between retrieved and current blips, then apply an instance-based judgment procedure.

Judgment On a trial if there is a recognition failure or if the retrieved blips do not match the current blip situation, then apply a wait for next blip judgment heuristic in which the spacebar is not pressed but the next blip situation is considered in a Z order. In case of recognition (or retrieval) success where the retrieved instance matches the current blip situation, apply an instance based judgment where the decision is to press the spacebar.

Choice The choice refers to picking the spacebar to press once the decision to press or not to press the spacebar has been made.

Execution Execute the spacebar or no spacebar press decision and wait for feedback from the system.

Also, in the above algorithm, the productions were assumed to take a commonly used value of 50 ms in ACT-R. There were some steps executed to read and encode the blip stimulus from the screen (i.e., visual time) in the model as well as some time expended in hearing deviant tones in the tone counting task that ran in the background. The visual and auditory times to see and hear each blip situation or each tone respectively were assumed to be at the ACT-R default values of 185 ms and 100 ms, respectively.

Sub-Symbolic Level of the IBL model

In ACT-R each instance (or chunk) has an activation value that is used for making retrieval in the recognition phase of the IBL model. An instance is retrieved from memory if the activation exceeds a retrieval threshold (RT), which sets the minimum activation with which an instance can be retrieved, and if the activation is the highest of all instance activations at the time of retrieval. ACT-R defines activation of an instance as:

$$A_i = B_i + \sum_l PM_{li} + \varepsilon \qquad (1)$$

Where B_i is the base-level activation and reflects the recency and frequency of practice of the *i*th instance, which is given by

$$B_i = \ln(\sum_{j=1}^{n} t_j^{-d}) \qquad (2)$$

Where n is the number of presentations of the *i*th instance; t_j is the time since the *jth* presentation; and *d* is the decay parameter (bll) which is usually set at 0.5.

Specification elements l in the PM summation are computed over the slot values of the retrieval instance specification. Match Scale *P* reflects the amount of weighting given to the similarity in Slot l, which is a constant across all slots with the value set at 1.0. Match Similarities *Mli* represent the similarity between the value l in the retrieval specification and the value in the corresponding slots of the current instance *i*. The PM mechanism as described above was computed by the Blip-Situation slot of the instance. We used a function to calculate the similarity based on the absolute value of the distance between the Blip-Situation slot of the current instance and those retrieved from memory.

Finally, ε is the noise value, which is composed of two components: permanent noise associated with each instance and instantaneous noise computed at the time of a retrieval request. Both noise values are generated according to a logistic distribution characterized by a parameter *s*. The mean of the logistic distribution is 0 and the variance σ^2 is related to the *s* value by

$$\sigma^2 = (\pi^2/3)\ s^2 \qquad (3)$$

We set the instantaneous noise *s* value in the IBL model to make it a part of the activation equation.

For the purpose of modeling the RADAR task, the parameters described above had the values given in Table 2.

Table 2: IBLT (ACT-R) Parameters with Values

Parameter/Slots	Value
RT	-18.0
bll	0.5
s	0.25
P	1.0
Blip-Situation	Integers from 1 to 18

Strategy-Based Learning Model

In the SBL model we used four strategies. One of these strategies called "exhaustive equals" strategy was an optimal strategy, which would always yield the optimal press of the spacebar key and produce 100% accuracy in the detection task. The other three strategies were suboptimal strategies. These strategies represent practically feasible strategies for the task, and they provide competition that can be used to model performance, through the utility learning mechanism in ACT-R. The chunk structure for the SBL model was exactly the same as the one for the IBL model.

The SBL model starts by making use of one of the four strategies defined in the model (if a strategy could not execute before a frame ended, then the model resets and tries to apply strategies again in the next frame). When the model executes, there is a competition set up between the three suboptimal strategies and the optimal "exhaustive equals" strategy. The initial utility of the optimal strategy is set lower than that of the suboptimal strategies, and one of the suboptimal strategies executes in the task during the initial blocks. The suboptimal strategies give negative rewards, whereas the optimal strategy gives a positive reward whenever executed. The end effect is that although the suboptimal strategies fire initially, later the optimal strategy picks up because it has increased its utility through repeated positive rewards. Given below are the details of the different strategies in the RADAR's SBL model.

Exhaustive Equals Strategy Compare all filled-in blips on the RADAR screen with all targets seen at the beginning of the trial and press spacebar if a match is found.

Random Equals Strategy Compare a randomly selected filled-in blip on the RADAR screen with a randomly selected target seen at the beginning of the trial and press spacebar if a match is found.

Bottom Two Equals Strategy Compare the bottom two (SW, SE) filled-in blips with all targets seen at the beginning of the trial and press spacebar if a match is found.

Top Two Equals Strategy Compare the top two (NW, NE) filled-in blips with all targets seen at the beginning of the trial and press spacebar if a match is found.

Each strategy is represented in an ACT-R production rule. Each production has a utility associated with it that can be set directly by setting a parameter :u. Like activations, utilities for productions could have noise added. The noise is controlled by the utility noise parameter s, which is set with the parameter :egs in ACT-R. The noise is distributed according to a logistic distribution with a mean of 0 and a variance of σ^2. If there are a number of productions competing with expected utility values Uj the probability of choosing production *i* is described by the formula:

$$\text{Probability (i)=Exp (Uj/(2)}^{0.5}\text{s) / Sum(Exp (Uj/(2)}^{0.5}\text{s))} \qquad (4)$$

The summation is over all the productions that are currently able to execute (their conditions were satisfied during the matching). Note however that Equation 4 only describes the production selection process. It is not actually computed by the system. The production with the highest utility (after noise is added) is the one chosen to execute. Also the utility learning mechanism updates the utility of a production (strategy) using the following equation:

$$\text{Ui(n)} = \text{Ui(n-1)} + \alpha * (\text{Ri(n)} - \text{Ui(n-1)}) \qquad (5)$$

If $U_i(n-1)$ is the utility of a production i after its n-1st application and $R_i(n)$ is the reward the production receives for its nth application (set by :reward parameter), then its utility is $U_i(n)$ after its nth application. In the above equation, α is the learning rate and is typically set at .2 (this value can be changed by adjusting the :alpha parameter with the sgp command). According to this equation the utility of a production is gradually adjusted until it matches the average reward that the production receives. A reward is delivered when a strategy fires, and the reward $R_i(n)$ that production i receives is the external reward received minus the time from the production's selection to the reward. This subtraction serves to give less reward to more distant productions. This reinforcement goes back to all the productions that have executed between the current reward and the previous reward.

For the purpose of the RADAR task, the parameters as described above had the following values.

```
:egs 0.1 :ul t                                   (9)
Exhaustive-Equals-Strategy :u -4 :reward +1
Random-Equals-Strategy :u 5 :reward -1
Bottom-Two-Equals-Strategy :u 10 reward -1
Top-Two-Equals-Strategy :u 5 :reward -1
```

The utility of the optimal strategy is lower than that of the three non-optimal strategies because we want to model to make errors similar to humans when it executes but reduce these errors overtime. The reward given to the suboptimal strategies decreases their utility, whereas the reward given to the optimal strategy increases its utility over time. The structure on utility and rewards might yield a monotonic dominance from the SBL approach even when changing environments and incorporating changes in the reward structure based upon changes in the environment is part of future work. Also, production compilation was not used in this model and it is a part of future work i.e. whether doing production compilation will make the SBL approach behave more like an IBL approach to modeling the experiment.

Model Fits to Human Data

The IBL and SBL models were run over 8 simulated participants in training and test conditions in RADAR. Figures 1 and 2 present the average times for correct responses during the training phase, including human data (Young et al., 2007) and SBL and IBL predictions. Figure 1 gives the average data for the within-subjects blocks CM1+1, CM4+4, VM1+1 and VM4+4. Both, the IBL and the SBL models fit the human data quite well, R^2=0.98 and RMSD=69 ms for IBL, and R^2=0.90 and RMSD=163 ms for SBL.

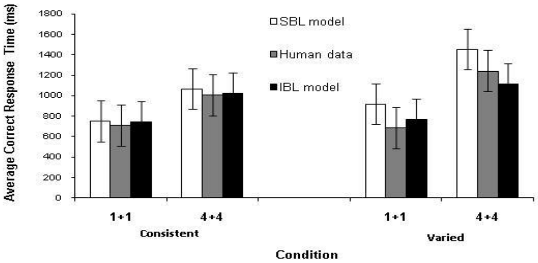


Figure 1: Average correct response times (ms) for CM 1+1, VM 1+1, CM 4+4, and VM 4+4 blocks in human data and SBL and IBL models during training. The error bars show 90% confidence intervals.

Figure 2 gives the average time for correct responses for the IBL, SBL, and human data across the silent and tone between-subjects conditions in the RADAR task. Again, both the IBL and the SBL models fit the human data very well, R^2=1.00 and RMSD=43 ms for IBL, and R^2=1.00 and RMSD=174 ms for SBL. In Figures 1 and 2, the SBL model seems to give generally higher time values compared to human data, and the SBL model has higher RMSD. This difference may be because in the SBL model the four strategies execute in productions in a fixed time (50 ms per production) and there is not speedup in the correct response times due to this fixed strategy execution time, whereas in the IBL model the speedup comes on account of activation-retrieval time speedup. The retrieval time decreases if the activation of instances increases over blocks (Anderson & Lebiere, 1998). Also, it is clear from Figure 1 that both models (i.e., IBL and SBL) take more time in 4+4 blocks than 1+1 blocks (for both consistent and varied mapping). This finding demonstrates the effects of workload well known in behavioral studies of automaticity (Gonzalez & Thomas, 2008). The workload effect results from the extra time taken to process four rather than one item.

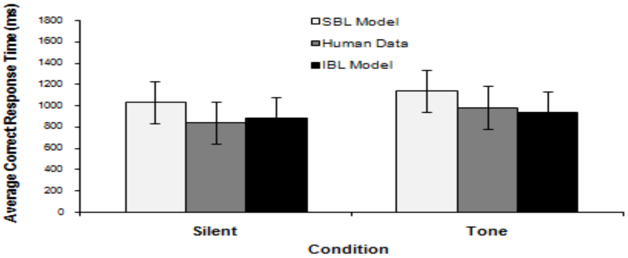


Figure 2: Average correct response times (ms) for silent and tone conditions for human data and SBL and IBL models during training. The error bars show 90% confidence intervals.

Similarly, the tone takes slightly more time to process than silent trials for both IBL and SBL models, as a result of the auditory productions to process the tones. Also, the difference is greater for the SBL model than the IBL model from the human data because in the SBL model there is no activation-retrieval speedup to compensate for time spent in tone counting whereas in the IBL model there is such a speedup, which reduces the overall time.

To test the adaptability of both SBL and IBL models and given the limited space in this paper, we report the data for only those groups that switch: tone-to-silent (Figure 3) and silent-to-tone (Figure 4). The R^2s for both the SBL and IBL models are very high at test (all are 1). Thus, the main difference between the models at test is in the RMSD measure. The SBL model has an RMSD = 160 ms when it is trained in tone and transferred to silent, whereas the IBL model's RMSD = 50 ms. The SBL model's RMSD when trained in silent and transferred to tone is 248 ms, whereas the RMSD value for the IBL model is 62 ms.

Thus, one can conclude that both models, SBL and IBL, are quite good according to the adaptability criterion, but the IBL model produces values closer to the human data than the SBL model does.

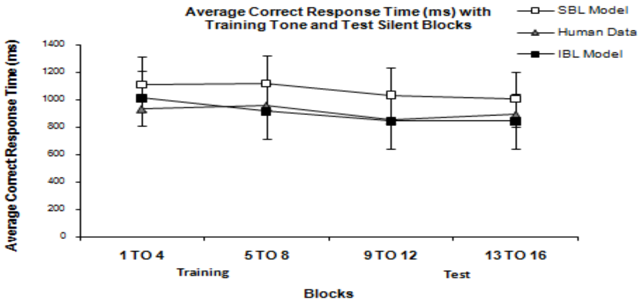


Figure 3: Average correct response times (ms) for human data and SBL and IBL models across blocks, for training in the tone and testing in the silent condition. The error bars show 90% confidence intervals.

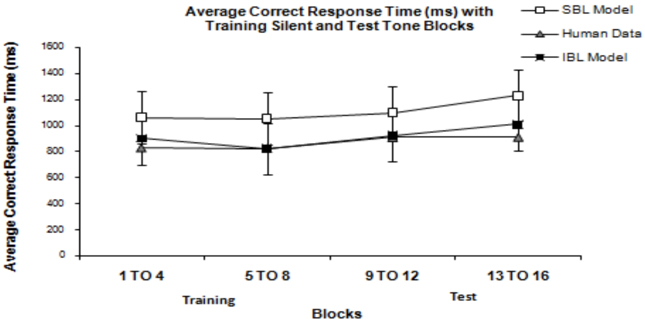


Figure 4: Average correct response times (ms) for human data and SBL and IBL models across blocks, for training in the silent and testing in the tone condition. The error bars show 90% confidence intervals.

Discussion and Future Work

Researchers often evaluate computational models of human behavior by comparing how different architectures or modeling approaches would represent a common task. This mode of model evaluation has been highlighted more recently by several model comparisons and competitions. The research we present here compares SBL and IBL approaches to modeling choice, but in this comparison in addition to using the same task, RADAR, we compare SBL and IBL approaches under the same architecture, ACT-R.

According to traditional goodness of fit measures, R^2 and RMSD, both SBL and IBL approaches to model choice fit human performance during a training experiment in RADAR quite well. Both representations are able to reproduce human data during the training conditions that varied both between subjects in tone/no tone training, and within subjects on the consistency of mapping and workload. When we compare the models in terms of their ability to adapt to transfer conditions, just as humans do, again both the SBL and IBL models have equally high values of R^2. But the IBL model was found to be closer to human data than the SBL model according to the RMSD measure during both training and test.

These results demonstrate that the numerical measures might not be good enough to tease two models apart. Further, the generalization criterion might not be sufficient either. To us, the IBL model has some advantage over the SBL model that the numerical measures do not show: Because the IBL model continues filling the chunk structure from the environment during test, the changes in conditions of the environment are captured in the instances stored and retrieved from memory, whereas the SBL approach is blind to changes in the environment. The SBL model continues applying the same strategies at test, which might not be as effective as they were during training, once the conditions of the task change. In addition, in dynamic situations the strategies are often unknown a priori or difficult to define at all. These are often discovered with task practice, and there is much evidence that learning in dynamic decision making tasks is implicit (Gonzalez et al., 2003). Often humans are unable to explain any rules or strategies used to solve a dynamic problem. Thus, we think that the IBL approach is more appropriate to model dynamic decision making (Gonzalez et al., 2003) than the SBL approach.

Acknowledgments

This research was supported by a MURI grant from the Army Research Office (W911NF-05-1-0153).

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. (2003). The Newell test for a theory of mind. *Behavioral and Brain Sciences, 26*, 587-639.

Busemeyer, J. R., & Wang, Y.-M. (2000). Model comparisons and model selections based on generalization criterion methodology. *Journal of Mathematical Psychology, 44*, 171–189

Cassimatis, N., Bello, P., & Langley, P. (2008). Ability, breadth, and parsimony in computational models of higher-order cognition. *Cognitive Science, 32*, 1304-1322.

Dutt, V., & Gonzalez, C. (2008). Instance and strategy based models in ACT-R. *Proceedings of 2008 Modeling, Simulation & Gaming (MS&G) Student Capstone Conference* (p. 19). Suffolk, VA: ODU-VMASC.

Gluck, K., Bello, P., & Busemeyer, J. (2008). Introduction to special issue: Model comparison. *Cognitive Science, 32,* 1245-1247.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science, 27,* 591–635.

Gonzalez, C., & Thomas, R. P. (2008). Effects of automatic detection on dynamic decision making. *Journal of Cognitive Engineering and Decision Making, 2,* 328-348.

Lebiere, C., Gonzalez, C., & Warwick, W. (2009). *Emergent complexity in a dynamic control task: Model comparison*. Manuscript submitted for publication.

Lovett, M. C. (1998). Choice. In J. R. Anderson, & C. Lebiere (Eds.). *The atomic components of thought* (pp. 255-296). Mahwah, NJ: Erlbaum.

Nisbett R. E. (Ed.) (1993). *Rules of reasoning*. Erlbaum Associates.

Young, M. D., Healy, A. F., Gonzalez, C., & Bourne, L. E., Jr. (2007, July). *The effects of training difficulty on RADAR detection*. Poster presented at the joint meeting of the Experimental Psychology Society and the Psychonomic Society, Edinburgh, Scotland.

Using Information Flow for Modelling Mathematical Metaphors

Markus Guhe (m.guhe@ed.ac.uk)
Alan Smaill (A.Smaill@ed.ac.uk)
Alison Pease (A.Pease@ed.ac.uk)
School of Informatics, Informatics Forum, 10 Crichton Street
Edinburgh EH8 9AB, UK

Abstract

We argue for two points in this paper. Firstly, formal models can be a useful means for cognitive modelling, in particular for domains that traditionally already use this kind of model. Secondly, we present a formal model of how two of the grounding metaphors for arithmetic proposed by Lakoff and Núñez (2000) can be linked to basic notions of arithmetic using the infomorphisms of the Information Flow theory.

Keywords: formal model, logic, metaphor, mathematics, scientific discovery

The Cognition of Mathematics

As of yet, there is no cognitive model of the way in which people invent mathematical concepts. As part of our research on understanding the cognition of creating mathematical concepts we are working towards such a model (Guhe, Pease, & Smaill, 2009). We build on two streams of research: embodied conceptualisation, which analyses mathematical ideas as being constructed by the cognitive process of metaphor (Lakoff & Núñez, 2000), and societal conceptualisation based on Lakatos's (1976) philosophical account of the historical development of mathematical ideas. Both argue strongly against the 'romantic' (Lakoff and Núñez) or 'deductivist' (Lakatos) style in which mathematics is presented as an ever-increasing set of universal, absolute, certain truths which exist independently of humans. In contrast to this view, our main interest in the *Wheelbarrow* project is how mathematical concepts are formed and modified by the embodied and situated human mind.

While there are cognitive models of learning mathematics (eg Lebiere, 1998; Anderson, 2007), there are to our knowledge no models of how humans *create* mathematics. Collecting empirical data on how scientific concepts are created is difficult, and this is true for case studies as well as laboratory settings. Using case studies (see, for example, Nersessian, 2008) suffers from the problems that they are not reproducible (and therefore anecdotal) and that they are usually created in retrospect, which means that they are very likely to contain many rationalisations instead of an actual protocol of the thought processes. Using a laboratory setting in contrast (cf Schunn & Anderson, 1998) means that the experiment has to be designed in such a way that the participants are limited in their possible responses, ie their degree of freedom is limited and it is uncertain whether or how this is different form the unrestricted scientific process.

Lakoff and Núñez (2000) claim that the human ability for mathematics is brought about by two main factors: our embodied nature and our ability to create and use metaphors. They describe how starting from interactions with the environment we build up (more and more abstract) mathematical concepts by processes of metaphor and abstraction. More precisely, they distinguish two kinds of metaphors: grounding metaphors and linking metaphors (p 53). In *grounding* metaphors one domain is embodied and the other abstract, eg the four grounding metaphors for mathematics, which we will describe below. In *linking* metaphors, both domains are abstract, which allows the creation of more abstract mathematical concepts. For example, having established the basics of arithmetic with grounding metaphors this knowledge is used to create – among others – the concepts of points in space, spaces of any number of dimensions and functions (p 387).

We follow Gentner (1983; see also Gentner & Markman, 1997, p 48) in assuming that metaphors are similar to analogies. Gentner proposes that when comparing two concepts we can distinguish between analogies, metaphors, literal similarities or mere appearance similarities by looking at the number of relations and properties that (the representations of) the two concepts have in common. For analogies, mainly relations between concepts are matched, while for metaphors a larger amount of properties are involved. Thus, the distinction between analogy and metaphor is only a difference in degree.

According to Gentner's (1983, p 156) *structure mapping theory* the main cognitive process of analogy formation is a mapping between the (higher-order) relations of conceptual structures. Although we use this approach for creating computational cognitive models of mathematical discovery (see Guhe et al., 2009 for an ACT-R model using *path-mapping* – a realisation of structure mapping in ACT-R developed by Salvucci & Anderson, 2001), in this paper we will present a formal model that specifies the particular grounding metaphors that Lakoff and Núñez (2000) propose. This formalisation will be a basis for enhancing the ACT-R model.

Lakoff and Núñez's Four Basic Metaphors of Arithmetic

Lakoff and Núñez (2000, chapter 3) propose that humans create the conceptual space of arithmetic with four different grounding metaphors that create an abstract conceptual space from embodied experiences, ie interactions with the real world. Since many details are required for describing these metaphors adequately, we can only provide the general idea here.

Object Collection The first metaphor, *arithmetic is object collection*, describes how by interacting with objects we experience that objects can be grouped and that there are certain regularities when creating collections of objects, eg by removing objects from collections, by combining collections, etc. By the process of metaphor (analogy) these regularities are mapped into the domain of arithmetic, for example, collections of the same size are mapped to the concept of number and putting two collections together is mapped to the arithmetic operation of addition.

Object Construction Similarly, in the *arithmetic is object construction* metaphor we experience that we can combine objects to form new objects, for example by using toy building blocks to build towers. Again, the number of objects that are used for the object construction are mapped to number and constructing an object is mapped to addition.

Measuring Stick The *measuring stick* metaphor captures the regularities of using measuring sticks for the purposes of establishing the size of physical objects, eg for constructing buildings. Here numbers correspond to the physical segments on the measuring stick and addition to putting together segments to form longer segments.

Motion Along A Path The *motion along a path* metaphor, finally, adds concepts to arithmetic that we experience by moving along straight paths. For example, numbers are point locations on paths and addition is moving from point to point.

Note that these metaphors are not interchangeable. All are used to create the basic concepts of arithmetic. For this initial proposal we will only consider the first two metaphors.

Formal Models

The field of cognitive modelling makes only little use of formal methods.[1] A reason for this may be the recognition that traditional claims that logic describes the way humans reason do not stand up to scrutiny – at least not in this generality.[2] Consequently, logic is hardly used for modelling human cognition. However, this is throwing out the baby with the bath water, because the rigour of logical models is a great methodological advantage. Moreover, for the domain that we are interested in (the cognition of mathematics) the results of the cognitive processes (the mathematical structures and processes) are usually already modelled with logic, which makes them easy to use. Having said this, it is also clear that such models are on a high level of abstraction, one comparable to differential equations or statistics. An advantage of this high level is the models' conciseness, which makes it easy to have models with a broad coverage.

Artificial intelligence, mathematics and automated theory formation, which all mainly use formal models, usually do not consider the work carried out in cognitive modelling. A major aim of our project is to bring the research in these disciplines closer to the research in cognitive modelling. Cognitive modelling will also profit from our approach, because most of the work on the cognitive abilities we are investigating (linguistics, mathematics) is not done as cognitive modelling approach but formally. Instead of recreating this research in cognitive modelling it is advantageous to transfer or link the existing research in a principled manner to cognitive modelling.

Finally, cognitive modelling is only concerned with creating models of the mind. Only rarely is there a computational or formal characterisation of the properties of the model itself. For example, a cognitive reasoning model is not usually specified with respect to completeness (is the model able to make all valid deductions?) or soundness (are all inferences drawn by the model correct given the used premises?) Determining such properties of a model (theory/system) is a strong point of formal systems.

To illustrate that logic is still a useful way to describe cognition we would like to draw attention to the Wason selection task.[3] The apparent failures of humans in this task can convincingly be explained as being effects of the participants having problems 'with interpreting how the experimenter intends the task and materials to be understood' (Stenning et al., 2006, p 63). In the case of the Wason selection task Schooler (2001) and Stenning et al. (2006) demonstrate that the apparent shortcomings of the participants are due to their understanding the task as being an inductive information gathering task rather than a deductive one, where they are supposed to reason from a set of premises to a conclusion. It has been observed that participants have no problems drawing the conclusions desired by the experimenter in a task with the identical logical structure but framed as a task of, for example, reasoning about the drinking age of youngsters.[4] The reason, however, that the inferences are 'correct' in this case is not, as is often suggested, that the problem is set in a different domain (a social situation instead of an abstract logic task) but that the participants understand the goal of the task in the way the experimenter intends, namely as being a deductive task – for which logic is a good model.

General Reasoning with Local Processing

A major difference between formal and cognitive modelling is that formal models usually consider all the knowledge in the

[1] 'Formal' in the sense of *logic* or *mathematics*. Computational models are formal as well, of course, and as they are usually realised on digital computers, they are also logical models.

[2] To be fair, it should be noted that most logicians today would say that logic describes how humans *ought* to reason. However, we propose that formal theories can contribute to understanding cognition.

[3] In the Wason selection task, the participants are presented with four cards, showing letters and numbers, for example: A, B, 3, 8. They are given a rule like *If one side has a vowel, then the other has an even number*. The participants now have to decide, which cards must be turned to see whether this rule is correct for these four cards. They should turn as few cards as possible. (Stenning, Lascarides, & Calder, 2006, p 28) The failure consists in the fact that participants usually turn over more cards than is necessary to draw the requested conclusion. This is often considered to be an example of a *confirmation bias* (Ross & Anderson, 1982, p 149), ie the preference to seek information that confirms held beliefs instead of trying to disconfirm such beliefs.

[4] The cards show the name of a drink on one side and the age of the drinker on the other, eg whiskey, orange, 19, 16. The rule is *If you drink alcohol, then you must be over 18 years old*. The participants are instructed to check whether all drinkers follow the rule.

system in each step. However, it is clear that this is not how cognition works. Instead only a small subset of the available knowledge is used for each computational step such as the firing of a production rule. This reduction of the considered knowledge to what we call a *local context* (Guhe, 2007) is the main reason why cognitive processes require much less computational power than artificial systems. Furthermore, the reason that (natural) cognitive systems can cope with the complexities of the real world while an artificial system is either prone to fall off a cliff (not enough knowledge considered) or being caught by a predator (computations are too slow), is that current artificial systems are very bad at establishing suitable local contexts – if they do it at all.

A reason for this is that the idea of a localised processing is a big challenge for formal models, because not taking all available knowledge into account can introduce inconsistencies, which will almost inevitably cause the system to fail. However, the *Information Flow* theory by Barwise and Seligman (1997) provides just what is needed to define distributed, localised formal systems. This means, the system consists of multiple subsystems (classifications, theories, local logics) that are connected by *infomorphisms* in a formally sound way. This makes it almost ideally suited for describing cognition in a formal manner, because humans are not only good at establishing local contexts but also at connecting the local contexts.

Coming back to the reasons for using formal methods the advantage of using logic for cognitive modelling is that it provides a general-purpose mechanism for reasoning – which is a main motivation for inventing and using logics in the first place. Although it is clear that human reasoning is strongly influenced by the current task and the current task demands, there is also a general ability to reason from premises to conclusions. It seems wasteful to have, for example, a different version of modus ponens in each task model. This does not mean that we propose a 'logic module', just that a general reasoning ability exists somewhere in the system. It can be implemented with means provided by existing cognitive architectures.

We have three main cases in mind where such a general reasoning mechanism is useful. Firstly, it can be used as a general model of distributed reasoning: if the system knows something within a local context and also knows how this knowledge is connected to another local context, then there is a principled way to use this connection to reason about the distal local context. Secondly, on a local level the reasoning on the chosen local context retains all the desirable properties of the chosen logic (soundness, completeness). Thirdly, such a mechanism is a good way to approach cognitive mathematics, because the results of the cognitive process (the mathematical structures) are already represented formally.

Information Flow

This section provides a short introduction to Information Flow theory. We will focus on the aspects that we need for our formalisation; a detailed discussion of Information Flow can be found in Barwise and Seligman (1997). We only need three of the main notions for our purposes here: classification, infomorphism and channel.

Classification A *classification* $\boldsymbol{A}$ consists of a set of tokens $\text{tok}(\boldsymbol{A})$, a set of types $\text{typ}(\boldsymbol{A})$ and a binary classification relation $\models_{\boldsymbol{A}}$ between tokens and types. In this way, the classification relation classifies the tokens, for example, for a token $a \in \text{tok}(\boldsymbol{A})$ and a type $\alpha \in \text{typ}(\boldsymbol{A})$ the relation can establish $a \models_{\boldsymbol{A}} \alpha$.

Graphically, a classification is usually depicted as in left part of figure 1, ie with the types on top and the token on the bottom.

$$\begin{tikzcd}
\text{typ}(\boldsymbol{A}) \arrow[r, "f^{\wedge}"] \arrow[d, dash, "\models_{\boldsymbol{A}}"'] & \text{typ}(\boldsymbol{B}) \arrow[d, dash, "\models_{\boldsymbol{B}}"] \\
\text{tok}(\boldsymbol{A}) & \text{tok}(\boldsymbol{B}) \arrow[l, "f^{\vee}"]
\end{tikzcd}$$

Figure 1: Two classifications ($\boldsymbol{A}$ and $\boldsymbol{B}$) and an infomorphism (f) in Information Flow

Infomorphism An *infomorphism* $f : \boldsymbol{A} \rightleftarrows \boldsymbol{B}$ from a classification $\boldsymbol{A}$ to a classification $\boldsymbol{B}$ is a (contravariant) pair of functions $f = \langle f^{\wedge}, f^{\vee} \rangle$ that satisfies the following condition:

$$f^{\vee}(b) \models_{\boldsymbol{A}} \alpha \quad \text{iff} \quad b \models_{\boldsymbol{B}} f^{\wedge}(\alpha)$$

for each token $b \in \text{tok}(\boldsymbol{B})$ and each type $\alpha \in \text{typ}(\boldsymbol{A})$, cf figure 1.

Note that the 'type relation' $f^{\wedge}$ and the 'token relation' $f^{\vee}$ point in opposite directions. (They are contravariant.) As a mnemonic the $^{\wedge}$ of $f^{\wedge}$ points upwards, where the types of classifications are usually written.

Channel A *channel* is a set of infomorphisms that have a common codomain. For example, the channel C depicted in figure 2 consists of a family of four infomorphisms f_1 to f_4 that connect the four classifications $\boldsymbol{A}_1$ to $\boldsymbol{A}_4$ to the common codomain $\boldsymbol{C}$. The common codomain is the *core* of the channel. Note that the infomorphisms of defining a channel are all pairs of functions, ie $f_1 = \langle f_1^{\wedge}, f_1^{\vee} \rangle$, etc.

$$\begin{tikzcd}
\boldsymbol{A}_1 \arrow[dr, "f_1"'] & & \boldsymbol{A}_2 \arrow[dl, "f_2"] \\
& \boldsymbol{C} & \\
\boldsymbol{A}_3 \arrow[ur, "f_3"] & & \boldsymbol{A}_4 \arrow[ul, "f_4"']
\end{tikzcd}$$

Figure 2: Channel $C = \{f_1, f_2, f_3, f_4\}$ and its core $\boldsymbol{C}$

The core is the classification that contains the information connecting the tokens of the classifications $\boldsymbol{A}_1$ to $\boldsymbol{A}_4$. The tokens of $\boldsymbol{C}$ are called *connections*, because they connect the tokens of the other classifications. In our application to arithmetic the core is the arithmetic knowledge that represents what is common to the different source domains – the common arithmetic properties of object collections, object constructions, etc.

Channels and cores are the main way in which Information Flow achieves a distributed, localised kind of representing knowledge. In other words, this is the property of the Information Flow approach the fits to the localised representation and processing found in cognition. At the same time, infomorphisms provide a principled way of representing the connection between the different local contexts. This is not the place to go into the details about this aspect of Information Flow, but Barwise and Seligman (1997) give a comprehensive account of the properties that are or are not preserved when following an infomorphism from one classification to another one.

Formalisation of the Arithmetic Metaphors

The basic idea of how to apply Information Flow theory to the four basic metaphors of Lakoff and Núñez (2000) is that each domain (object collection, object construction, measuring stick, motion along a path and arithmetic) is represented as a classification and the metaphors/analogies between the domains are infomorphisms.

Information flow (which give the theory its name) captures regularities in the distributed system (see the First Principle of Information Flow, Barwise & Seligman, 1997, p 8). So, the infomorphisms between the four source domains and the core (arithmetic) capture the regularities that link these domains to arithmetic, and the arithmetic classification represents the knowledge of what these domains have in common. (A full arithmetic classification contains more than these commonalities – think of arithmetic concepts arising by linking metaphors like the concept of zero –, but for our current purposes it suffices to think of it this way.)

Object Collection

Classification We define a classification ***CL*** for the domain of object collections, cf table 1. The tokens of the object collection domain are actual physical instances of collections of objects that are or have been encountered by the cognitive agent. Formally, we represent them as sets of objects named $coll_A, coll_B, \ldots$

The tokens are classified by the size (cardinality) of the collection, ie types are sets with a number of distinct elements. Following Lakoff and Núñez (2000, p 55) we assume an innate or early developed subitising ability, ie the ability to determine the cardinality of small object collections of up to three or four objects. As a convention we write oc_1 for the type set with one object, oc_2 for the one with two objects, etc.

The classification relation $\vDash_{\boldsymbol{CL}}$ for object collections relates those sets for which each object of the token set can be mapped to exactly one object of the type set, ie no object of the token set and no object of the type set is left over and each object is mapped to exactly one element of the other set.

Table 1: The arithmetic is object collection metaphor.

object collection	arithmetic
collections of objects of the same size	numbers
size of collection	number
bigger	greater
smaller	less
smallest collection	the unit (one)
putting collections together	addition
taking a smaller collection from a larger collection	subtraction

Given this classification, we can now assign a type to each token, eg $coll_A \vDash_{\boldsymbol{CL}} oc_2, coll_B \vDash_{\boldsymbol{CL}} oc_1$. Figure 3 shows an example for an object collection with three objects.

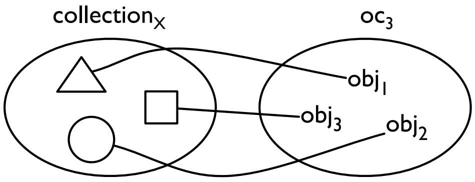


Figure 3: Example of the token–type relation for an object collection with cardinality 3

By proposing this classification we do not want to suggest that this is the only suitable classification for object collections; it is simply one that is suited for our goal of linking this source domain to arithmetic. A classification suitable for other purposes may be a classification by kind like blocks or balls.

Size The size of an object collection is the cardinality of the type set. Thus, given a token set $coll_A$, a type set oc_A with $coll_A \vDash_{\boldsymbol{CL}} oc_A$

$$size_{\boldsymbol{CL}}(col_A) = |oc_A|.$$

Smallest collection The smallest collection (the unit collection) is the type set with a single object, ie oc_1.

Bigger and smaller For comparing two object collections the type sets are aligned and a one-to-one mapping is established between the two type sets for as many elements as possible. The bigger collection is the one with at least one unmapped object. The smaller object collection is the other collection.

Formally, we define it as follows. Given two type sets oc_A and oc_B:

$$bigger_{\boldsymbol{CL}}(oc_A, oc_B) = \begin{cases} true, & \text{if } |oc_A| > |oc_B|, \\ false, & \text{if } |oc_A| < |oc_B|. \end{cases}$$

$smaller_{\boldsymbol{CL}}$ is the inverse.

Although not mentioned by Lakoff and Núñez (2000) in this context, two object collections are equal (in size) if they have the same type (or if they have the same cardinality):

$$\mathit{equal}_{\boldsymbol{CL}}(oc_A, oc_B) \text{ iff } oc_A = oc_B.$$

Putting collections together Putting separate and disjoint collections together is the union of token sets:

$$\mathit{putTogether}_{\boldsymbol{CL}}(\mathit{coll}_A, \mathit{coll}_B) = \mathit{coll}_A \cup \mathit{coll}_B.$$

The corresponding type is the type set that has a one-to-one-mapping between all objects in both sets. We write this as $oc_A \bullet_{\boldsymbol{CL}} oc_B$ (where $\mathit{coll}_A \vDash_{\boldsymbol{CL}} oc_A$ and $\mathit{coll}_B \vDash_{\boldsymbol{CL}} oc_B$), which is another possible type. Note that because the sets are disjoint $|oc_A \cup oc_B| = |oc_A| + |oc_B|$.

Taking a smaller collection from a larger collection Taking a smaller set from a larger set[5] is the set difference. Thus, for two object collections coll_A and coll_B with $\mathit{coll}_A \vDash_{\boldsymbol{CL}} oc_A$, $\mathit{coll}_B \vDash_{\boldsymbol{CL}} oc_B$ and $\mathit{bigger}_{\boldsymbol{CL}}(oc_B, oc_A)$:

$$\mathit{takeSmaller}_{\boldsymbol{CL}}(\mathit{coll}_A, \mathit{coll}_B) = \mathit{coll}_A \setminus \mathit{coll}_B$$

The corresponding type is $oc_A \circ_{\boldsymbol{CL}} oc_B$ (with $\mathit{coll}_A \vDash_{\boldsymbol{CL}} oc_A$ and $\mathit{coll}_B \vDash_{\boldsymbol{CL}} oc_B$), which is another possible type.

Object Construction

For the object construction domain we define a classification ***CN***. In contrast to ***CL*** the tokens and types of this classification are not defined as simple sets but as sets of sets. We name token sets as $\mathit{cons}_A, \mathit{cons}_B$, etc and type sets as $\mathit{ocn}_A, \mathit{ocn}_B$, etc. Note that in contrast of ***CL*** there is no implicit (and incidental) coding of size in the types of ***CN*** as the types can have just as complex substructures as the tokens, see the examples below.

Table 2: The arithmetic is object construction metaphor.

object construction	arithmetic
objects	numbers
smallest whole object	the unit (one)
size of object	size of number
bigger	greater
smaller	less
constructed object	result of arithmetic operation
whole object	a whole number
putting objects together to form larger objects	addition
taking smaller objects from larger objects to form other objects	subtraction

Smallest Whole Object A smallest whole object is represented as a singleton set for token sets, ie a set that contains one physical object that cannot be deconstructed. The corresponding type set is the set containing the empty set, ie $\{\varnothing\}$.

Size of Object The size of an object is the cardinality of the flattened type set. Thus, given a token set cons_A and a type set ocn_A with $\mathit{cons}_A \vDash_{\boldsymbol{CN}} \mathit{ocn}_A$

$$\mathit{size}_{\boldsymbol{CN}}(\mathit{cons}_A) = |\mathit{flat}(\mathit{ocn}_A)|.$$

A flattened set is obtained by applying the function flat:

$$\mathit{flat}(\{A_1, A_2, \dots\}) = \begin{cases} \{A_1, \mathit{flat}(A_2, \dots)\}, & \text{if } |A_1| = 1, \\ \{\mathit{flat}(A_1), \mathit{flat}(A_2, \dots)\}, & \text{otherwise.} \end{cases}$$

flat is defined on sets in general and is not restricted to ***CN***.

Bigger and smaller Analogous to the definition for object collections, bigger is defined as

$$\mathit{bigger}_{\boldsymbol{CN}}(\mathit{ocn}_A, \mathit{ocn}_B) = \begin{cases} \mathit{true}, & \text{if } \mathit{size}(oc_A) > \mathit{size}(oc_B), \\ \mathit{false}, & \text{if } \mathit{size}(oc_A) < \mathit{size}(oc_B). \end{cases}$$

$\mathit{smaller}_{\boldsymbol{CN}}$ is the inverse.

Constructed Object A constructed object is an object that consist of other objects. Thus, a constructed object is a set that has other object sets as elements. That is, given sets cons_A, cons_B, … the constructed object set is $\{\mathit{cons}_A, \mathit{cons}_B, \dots\}$.

Whole Object Physical objects are always whole objects.

Putting Objects Together to form Larger Objects Putting objects together is defined as the union of the sets of object sets:

$$\mathit{putTogether}_{\boldsymbol{CN}}(\mathit{cons}_A, \mathit{cons}_B, \dots) = \{\mathit{cons}_A, \mathit{cons}_B, \dots\}.$$

The corresponding type is $\mathit{ocn}_A \bullet_{\boldsymbol{CN}} \mathit{ocn}_B$ (with $\mathit{cons}_A \vDash_{\boldsymbol{CN}} \mathit{cns}_A$ and $\mathit{cons}_B \vDash_{\boldsymbol{CN}} \mathit{cns}_B$), which is another possible type.

For example, given the object sets (o_X being a representation of a physical object) $\{\{o_1\}, \{o_2\}\}$, $\{\{o_3\}, \{\{o_4\}, \{o_5\}\}\}$ and $\{o_6\}$ the assembled object is $\{\{\{o_1\}, \{o_2\}\}, \{\{o_3\}, \{\{o_4\}, \{o_5\}\}\}, \{o_6\}\}$.

Taking Smaller Objects from Larger Objects to Form Other Objects Taking smaller objects from larger objects is defined as the set difference. More precisely, the result of this operation is a pair of sets – the difference set and the subtracted set. Thus, given two sets cons_A and cons_B with $\mathit{cons}_B \subset \mathit{cons}_A$

$$\mathit{takeSmaller}_{\boldsymbol{CN}}(\{\mathit{cons}_A, \mathit{cons}_B\}) = (\{\mathit{cons}_A \setminus \mathit{cons}_B\}, \mathit{cons}_B).$$

For example, given the object set $\{\{o_1\}, \{\{o_2\}, \{o_3\}\}, \{o_4\}\}$ the object $\{o_4\}$ can be removed by

$$\{\{o_1\}, \{\{o_2\}, \{o_3\}\}, \{o_4\}\} \setminus \{o_4\}$$

The smaller object taken from the larger object is then $\{o_4\}$ and the remainder of the larger object is $\{\{o_1\}, \{\{o_2\}, \{o_3\}\}\}$. The type of this operation is $\mathit{ocn}_A \circ_{\boldsymbol{CN}} \mathit{ocn}_B$ (with $\mathit{cons}_A \vDash_{\boldsymbol{CN}} \mathit{cns}_A$ and $\mathit{cons}_B \vDash_{\boldsymbol{CN}} \mathit{cns}_B$), which is another possible type.

[5] Note that attempting to take a larger set from a smaller set has no physical correlate. This operation can only be performed in the abstract arithmetic domain, where it leads to the invention (or discovery) of negative numbers.

Arithmetic

The classification $\boldsymbol{AR}$ representing arithmetic consists of the abstract numbers as tokens and concrete instances (uses) of numbers as types (for example as in *There are seven trees*).

The smallest number is 1.[6]

The arithmetic operations are defined as usual.

Metaphor Infomorphisms

With these three classifications, the infomorphisms between the two classifications representing the source domains ($\boldsymbol{CL}$ and $\boldsymbol{CN}$) and the core $\boldsymbol{AR}$ are straightforward.

Infomorphism from Object Collection to Arithmetic The infomorphism linking the object collection domain to arithmetic is defined as $f : \boldsymbol{CL} \rightleftarrows \boldsymbol{AR}$. The relation between types is then $f\hat{}(oc_A) = |oc_A|$, the relation between tokens $f\check{}(num) = coll_A$ where num is an arithmetic number and $coll_A$ is a representation of the physical object collection that the number refers to.

The smallest collection is $f\hat{}(oc_1) = 1$, and the comparison relations and operations are defined as follows:

- $f\hat{}(bigger_{\boldsymbol{CL}}) = >$
- $f\hat{}(smaller_{\boldsymbol{CL}}) = <$
- $f\hat{}(\bullet_{\boldsymbol{CL}}) = +$
- $f\hat{}(\circ_{\boldsymbol{CL}}) = -$

Infomorphism from Object Constructions to Arithmetic Similar to the definition above, the informorphism $g : \boldsymbol{CN} \rightleftarrows \boldsymbol{AR}$ relates types as $g\hat{}(obj_A) = |flat(obj_A)|$ and tokens as $g\check{}(num) = cons_A$, where $cons_A$ represents the object being referred to by the number and where $num \vDash_{\boldsymbol{AR}} \nu$, $cons_A \vDash_{\boldsymbol{CN}} cns_A$ with $size(cns_A) = \nu$. The other properties are defined as:

- $g\hat{}(\{\varnothing\}) = 1$
- $g\hat{}(bigger_{\boldsymbol{CN}}) = >$
- $g\hat{}(smaller_{\boldsymbol{CN}}) = <$
- $g\hat{}(\bullet_{\boldsymbol{CN}}) = +$
- $g\hat{}(\circ_{\boldsymbol{CN}}) = -$

Channel

Given these two infomorphisms, the channel C is the set of these two infomorphisms ($C = \{f, g\}$).

Conclusions and Future Work

We have argued that a formal approach like Information Flow can be used to great advantage for cognitive modelling. We provided a formalisation of the basic aspects of two of the grounding metaphors proposed by Lakoff and Núñez (2000) that humans use for creating arithmetic. Although this high level of modelling does not directly address human task performance, it offers important insights into the generalisations of the different source domains required to invent arithmetic.

We will extend our formalisation (1) to include the other basic metaphors and the linking metaphors within arithmetic; (2) by adding further notions from Information Flow to the formalisation, in particular regular theories and local logics. With these extensions we will be able not only to represent the required knowledge but also to model the corresponding processes.

Acknowledgments

The research reported here was carried out in the *Wheelbarrow* project, funded by the EPSRC grant EP/F035594/1.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Barwise, J., & Seligman, J. (1997). *Information flow: The logic of distributed systems*. Cambridge University Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155–170.

Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *Am. Psychologist*, *52*(1), 45–56.

Guhe, M. (2007). *Incremental conceptualization for language production*. Mahwah, NJ: Lawrence Erlbaum.

Guhe, M., Pease, A., & Smaill, A. (2009). A cognitive model of discovering commutativity. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*.

Lakatos, I. (1976). *Proofs and refutations: The logic of mathematical discovery*. Cambridge: Cambridge University Press.

Lakoff, G., & Núñez, R. E. (2000). *Where mathematics comes from: How the embodied mind brings mathematics into being*. New York: Basic Books.

Lebiere, C. (1998). *The dynamics of cognition: An ACT-R model of cognitive arithmetic*. Unpublished doctoral dissertation, CMU Computer Science Department.

Nersessian, N. J. (2008). *Creating scientific concepts*. Cambridge: MIT Press.

Ross, L., & Anderson, C. A. (1982). Shortcomings in the attribution process. In D. Kahneman, P. Slovic, & A. Tversky (Eds.), *Judgment under uncertainty: Heuristics and biases* (pp. 129–152). Cambridge University Press.

Salvucci, D. D., & Anderson, J. R. (2001). Integrating analogical mapping and general problem solving: The path-mapping theory. *Cognitive Science*, *25*(1), 67–110.

Schooler, L. J. (2001). Rational theories of cognition in psychology. In W. Kintsch, N. J. Smelser, & P. B. Baltes (Eds.), *International encyclopedia of the social and behavioral sciences* (pp. 12771–12775). Oxford: Pergamon.

Schunn, C. D., & Anderson, J. R. (1998). Scientific discovery. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (pp. 385–427). Mahwah, NJ: Erlbaum.

Stenning, K., Lascarides, A., & Calder, J. (2006). *Introduction to cognition and communication*. Cambridge: MIT Press.

[6] Note that there is no physical correspondence to 0. The absence of a physical object collection is not a collection. The object constructed of no object is not an object (it does not exist). Remember that historically 0 is a very late invention/discovery, a reaction to certain needs in arithmetic, cf also Lakoff and Núñez (2000).

The Drunken Mice

Julia Hagg (julia.hagg@stud.uni-bamberg.de)
Institut für Theoretische Psychologie. Universität Bamberg, Markusplatz 3,
D-96047 Bamberg, Germany

Dietrich Dörner (dietrich.doerner@uni-bamberg.de)
Institut für Theoretische Psychologie. Universität Bamberg, Markusplatz 3,
D-96047 Bamberg, Germany

Abstract

In this project we investigate, how a theory about the impact of drugs (alcohol) on cognition could look like. We used simulated agents ("mice"), which live in groups, to understand the effects of alcohol. The agents are programmed according to the Psi - theory, which is a theory about the interaction of cognitive, emotional and motivational processes. We added to the normal (simulated) environment of the mice, which includes sources of food, water and "healing herbs", sources of "julihuana", which is a beer-like beverage. Although in the original formulation of the theory the consumption of drugs was not considered, the behaviour of the "mice", which have access to "julihuana" changed in a way, which exhibits strong parallels to human behaviour. The drunken and (after a time) often addicted mice lost their social contacts (less friends), their cognitive processes (perception, remembering, planning) became rough and shallow. The mice felt "strong" when drunken and very weak without julihuana. When drunken therefore they became very aggressive, but depressive when without julihuana.

Keywords: memory; plan; addiction; cognitive map, alcohol; action-regulation.

Cognition and Drugs

In nearly every human culture drugs play an important, often however detrimental role. Men (but animals, too) are inevitably attracted by drugs. The relationship of drugs to cognitive processes is close. Especially intellectuals, writers, painters, composers seem to exhibit a strong tendency to alcoholic beverages (see Lange-Eichbaum, 1986). Accordingly you will find Beethoven as a heavy drinker, Mozart, too. Goethe every day drank two bottles of wine, he was an alcoholic according to the standards of today, Schiller drank even more. A theory about human behaviour should be able to answer the question, why drugs are so attractive. In this paper we will try to answer this question – and some more about the impact of alcohol on cognitive, emotional and motivational processes.

The Mice

The mice are simulated agents, which live in an environment an example of which can be seen in figure 1. The environment forms an island with different regions. On region 1 food is growing, region 2 offers water, region 3 offers healing herbs, where the mice can cure their wounds. Wounds can be the result of aggressions of other mice (the mice can even kill each other!) or the results of falling stones in certain dangerous regions in the "world" of the mice. Region 4 is such a dangerous region. (The "world" of figure 1 is only an example. The mice' worlds normally are much larger!)

In "alcoholic" environments some of the waterholes are exchanged by regions which offer julihuana instead of water. Julihuana is an alcoholic beverage, a kind of palm-beer. It was invented by Julia Hagg and therefore is called julihuana.

The mice do not know their environment completely, but have to explore it to learn where fresh water, food or healing herbs (or julihuana, region 5) could be found or which regions are dangerous. The mouse' memory contains cognitive maps in the form of landmark paths to goals in the environment. A projection of such a path on the "world" of the mice can be seen in figure 1.

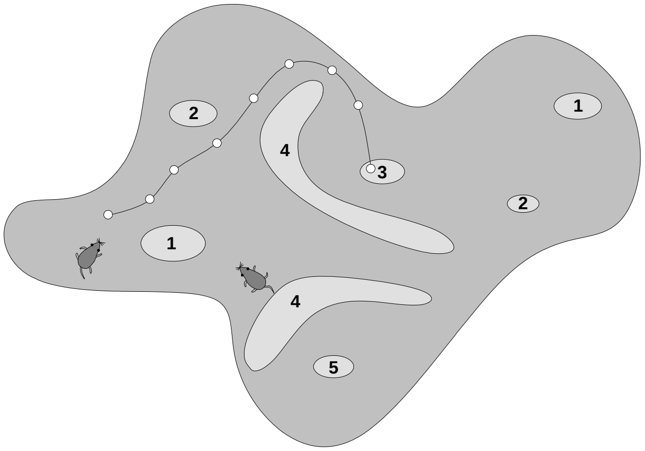


Figure 1: The environment of the mice.

Additionally the mice have a social memory about their friends and enemies. The mice have a desire for social contacts and those mice with whom they have such contacts become "friends". Friends help each other with food or water or when a mouse is attacked by another one. (Such an aggression naturally is the basis for enmity!) Help in a dangerous situation strengthens friendship. If it is not strengthened the memory of friends and enemies decays by time. The same is true for the cognitive maps of the environment.

The mice can get offspring. For this purpose they have sexual desires, too. The mothers educate their children, i.e. they "tell" them, where food or water can be found and which places are dangerous and who is a friend or an enemy.

The Psychic Organization of the Mice

Figure 2 exhibits a rough sketch of the mice' psychic organization. There is a motivational system, which controls the direction of behaviour. Additionally there is a unit 'Perceive and Act' which transforms the general directives of the motivational system into real actions. It adapts general plans or directives to the conditions of the current situation. The third unit, Thinking, is used if no appropriate plan for the actual goal (the actual motive) is available in memory. In this case the unit Thinking tries to construct a new plan by planning activities.

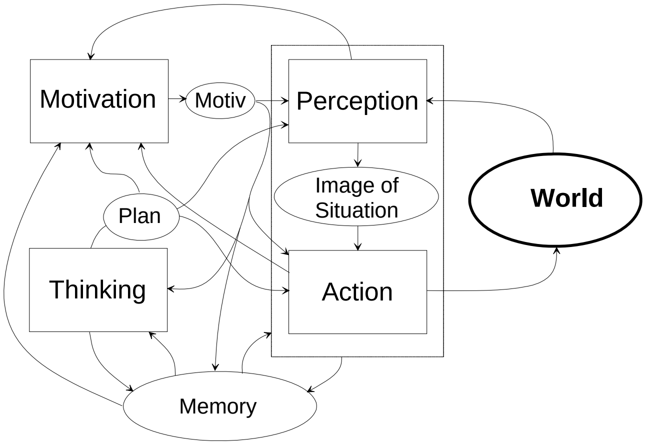


Figure 2: The Aristotelian soul of the mice.

This organization is nothing else but the basic organization of the "soul" which Aristotle postulated (Aristoteles, Buch II). It includes the three basic Aristotelian "capabilities", Motivation, Perception (and Action) and Thinking, which together produce the capability of "striving". This Aristotelian Architecture is rough, but it is sound and a good starting point to debate the overall structure of the "soul". (The Aristotelian concept of "soul" is quite simple and straightforward and astonishing close to information theory: "Soul" is nothing else but the controlling and monitoring device of the body and the soul itself is a physical function.)

The mice are programmed according to the Psi-theory, a theory about the interaction of cognitive, motivational and emotional processes (Bach; 2009; Dörner, 1999; Dörner et al. 2002). The mice are cognitively reduced (to fit in rather big numbers into a normal PC), but emotionally elaborated. Therefore these beings are called "mice", as mice are nice, but small. Now we are going to explain some basics of the Psi-theory.

Motivation The unit Motivation is the core of the whole system. Its organization is shown in figure 3. There are a number of "tanks" (mathematically: accumulative stores), which represent the need-system. The "Hunger-tank" for instance is filled up by the intake of food and emptied by the consumption of energy by the activities of the body. Attached to the tank is a system, which indicates the setpoint deviation of the actual level in the tank. All the other tanks are organized in the same way.

We assume that five needs or need groups are sufficient to generate all kinds of human motives, namely existential needs (hunger, thirst, pain, …), sexuality, affiliation (need for binding to a group), certainty (need for predictability of the events in the environment or for the explanation of such events) and competence (need for mastery, need for the ability to solve problems). These needs can amalgamate to form "macro-needs" or need-amalgames.

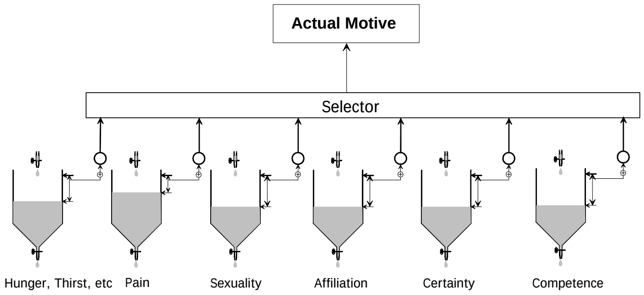


Figure 3: The motivational system.

The Selector-system is very important for action regulation. It makes a choice between the different active needs (setpoint-deviation >0) and selects the need with the highest motive-strength. Motive-strength is calculated according to the expectancy (of success) × value (of the need) - principle. The higher the setpoint deviation the higher is the value (of the satisfaction) of the need. The expectancy of success is calculated either on the basis of the knowledge about goal-paths in memory, i.e. knowledge about paths which lead for instance to a water hole. Additionally the "general competence" is taken into account, which is the basis of the calculation of success-expectancy, when knowledge about goal-leading paths doesn't exist. In this case the "General Competence" is heuristic competence, the confidence in finding a method to reach the goal. "General competence" is nothing else, but the level of the competence tank. Roughly the system Selector works according to the expectancy–value - principle: it selects a goal according to the setpoint deviations of a tank, the goals which are known to result in an increase of the level of the "tank" and the success probability. This is calculated on the basis of the success-probability of a known operator (for instance a landmark – path as representing a sequence of motions towards a goal) and the level of the competence tank, representing "general competence".

The tanks for affiliation, certainty and competence are "information-tanks". This means that no material or energy fills or empties these tanks, but signals, information. In the case of the affiliation tank the inputs are signals of legitimacy (L-Signals; Boulding, 1974). These are signals of "okness", a clap on the shoulder or a smile. For the certainty tank the input are signals of certainty, for instance a prognosis which comes true. The tank is emptied by signals of uncertainty, a prognosis which comes false or a situation which is inscrutable. Input signals for the competence tank are signals of efficiency, the solution of a problem for instance is a signal of efficiency. But each satisfaction of a need is a competence signal, too. The competence-tank is emptied by failures.

Additionally to the emptying signals for the affiliation and the competence tank a "leak" exists. These tanks therefore empty by time, without any signals. This is very important, as this means, that automatically a desire to reestablish social relations and to reaffirm competence will arise after some time.

Cognition Cognitive processes are processes of perceiving, memory search and planning, i.e. construction of a plan as a sequence of steps towards a goal. The system Selector of the motivational system activates memory search to look for goal leading paths. If it does not find such a path it activates a planning process to construct a sequence of steps towards a goal. This process in the mice is realized as the GPS – process of Newell & Simon, 1972, the General Problem Solver. – It is very important to understand that cognitive processes are emotionally modified. This means, that their form alters as a consequence of emotional changes.

Emotion We believe that it is sound to consider emotions not as processes of their own, but as forms of the organization of motivational and cognitive processes dependent on the level of the uncertainty and – most important – the competence tank. So for instance anxiety means a low level in the certainty and the competence – tank. These low levels produce a high level of arousal, a low resolution level, i.e. rough cognitive processes, flight tendencies, but aggressive tendencies, too. Additionally in anxiety "weak" processes of exploration may be started to diminish uncertainty. Anxious persons however will mostly try to defend their model of the world against falsification, for their image of the world is the last hold to protect them from despair. Therefore anxious persons will easily believe, what is in correspondence with their image of the world, but will never accept news disconfirming their view of the world. This combination of credulousness and distrust is a modulation of perceptive processes and impedes effective exploration.

In a similar way other emotional states and processes can be characterized as systems of cognitive and motivational processes triggered by the level of the certainty and the competence tank. Emotions are adaptations of cognitive and motivational processes to competence and certainty as measured by the level in the respective tanks. An empty competence tank "says": "You should be extremely cautious when acting, as you are helpless! Whatever you will do could be a mistake or failure with a high probability!" (Naturally the competence tank does not "say" anything. But it generates the respective behavioural tendencies.)

An empty certainty tank "says": "You are not able to foresee, what will happen. Therefore your vigilance should be as high as possible. Not one moment you should stop to monitor your environment!" Under such conditions of high vigilance elaborated planning processes are nearly impossible; therefore vigilance impairs planning processes and hence the quality of action.

The most basic emotions are pleasure and unpleasure. How do they fit into this model of emotions? In the simplest way! Pleasure is a filling of the competence tank. This implies to feel strong, to relax (because there does not exist any danger which cannot be overcome), to foresee nothing but successes in the future.

And unpleasure? It means emptying the competence tank. This however means to feel weak, depressive. Normally it means stress, high arousal, low resolution level of cognitive processes. The current problems seem to be insolvable. Future looks dark.

The General Organization Figure 4 shows a rough sketch of the interplay of processes and data according to the Psi-theory. The System "Selector" generates an actual goal and looks for an appropriate action. For this purpose it looks for a transition from the momentary given situation to the goal. If it finds such an operator, the operator is executed. If not, planning is activated. If this is successful the plan is run. This could be successful or not. If not, explorative activities are activated to gain a better understanding of the structures and possibilities of the environment and hence to be able to construct another plan. If this is successful, the new plan will be run, otherwise the system shifts to a behaviour of the trial-and-error type.

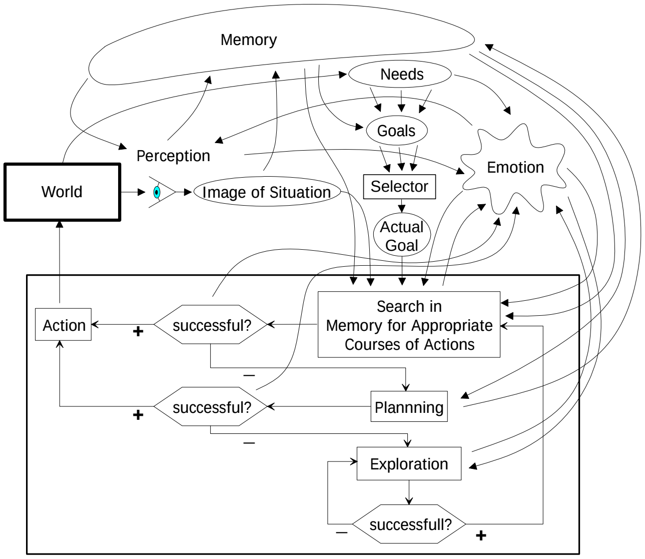


Figure 4 The general organization of the behaviour of the mice. See text.

Very important are "interrupts". As the level of the tanks changes nearly from moment to moment, the conditions for the selection of an actual goal change , too, as the basis of the selection of the actual goal is the expectancy-value – principle. If because of failures the level of the competence tank becomes low, the behaviour of the mice looses sustainability. The goals will change rather frequently as with each failure the competence of the actual intention decreases and other intentions may take over.

The Effects of Alcohol

What are the impacts of alcohol on human behaviour? The direct effect is to feel good, to feel "strong". Everything becomes manageable; there are no problems. On the other hand the ability to follow one line of thought is impaired. Sustainability diminishes. Thinking becomes rough and shallow, but to a certain degree is less supervised and therefore could be "creative" (see Feuerlein, 1998; Lindenmeyer, 2005).

Abuse of alcohol for a longer time results in a loss of social contacts and an inability to sustain intentions for a longer period. Short time and working-memory seem to be impaired. Self-control suffers and people often lose their job because of inefficiency. Alcholics often become depressive (when not drunk) see Schuckit, 1994.

To investigate the impact of alcohol to the mice' behaviour we used environments which were different only in that respect, that in the alcohol-version some of the waterholes were replaced by sources of julihuana. When a mouse drinks julihuana, the first time simply because she is thirsty, there will be an input to the competence tank. This means that the mouse will feel good, much better than if she had drunk water. The input to the competence tank however is different to a normal efficiency signal as it is not accompanied by learning a new method to overcome difficulties or by the affirmation, that, what has been done, has been an appropriate method to achieve goals. Alcohol produces an efficiency signal without any effort. What is the effect of julihuana to the behaviour of the mice?

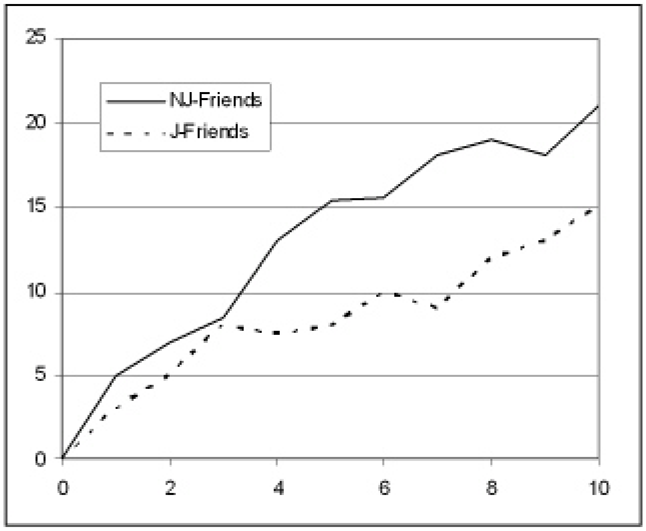


Figure 5: The average number of friends (per mice) of the J- and the Non-J- mice over a period of 100000 (=10) cycles. (One cycle corresponds to 30 Mice-minutes. 100000 cycles mean 50000 hours or 5.7077 years or 5 years and 258 days.)

Motivational Effects

The most significant motivational effect is that the julihuana-mice (J-mice) have a much lower affiliative motivation than the "normal" mice (Non-J-Mice), which have no access to Julihuana. Normally for the mice social contacts are the main source of competence, of "feeling good". To a high degree affiliation is replaced by alcohol with the J-mice! Figure 5 shows that the (average) number of friends diminished considerably with the J-mice. The same applies to the enemies (not indicated on Figure 5.) - The difference is significant on the 0.001 – level. (The same applies for all the other differences between the J- and the Non-J-mice in this article.)

The shrinking of the social contacts of the mice is in good accordance about what a lot of authors report about alcoholics. It is typical for alcoholics, that their social environment shrinks, see Feuerlein, 1998). This again had a severe impact on the population development of the mice. Figure 6 shows, that the growth of the population is much slower with the J-mice than with the Non-J-mice.

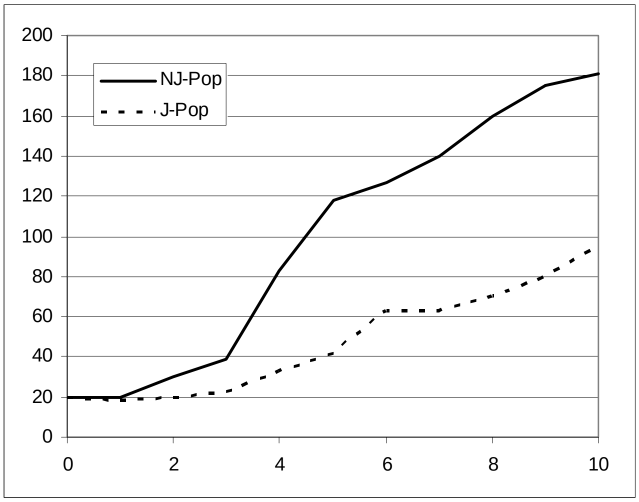


Figure 6: The growth of the population.

This is mainly due to the fact that the J-mice had less sexual contacts. For the mice, as for men, too, mostly sexual contacts have the condition that the partners like each other, are "friends", so to speak. As the number of friends is less for the J-mice than for the others, the number of sexual contacts diminishes, too, with the consequences visible on figure 6.

Additonally to the impact which julihuana has on the competence tank, we increased in another experiment the impact to the certainty tank. This means, that julihuana diminishes uncertainty (lack of predictability and "explainability" of the "world"). This additional impact of julihuana to the need-system of the mice increased the attractiveness of the drug considerably as visible in figure 7. This effect is due to the lower resolution level of the cognitive processes. Perception and remembering on a low resolution level become rough and "overinclusive". Objects, which are only similar are treated as if they were identical. An apple is a pear!

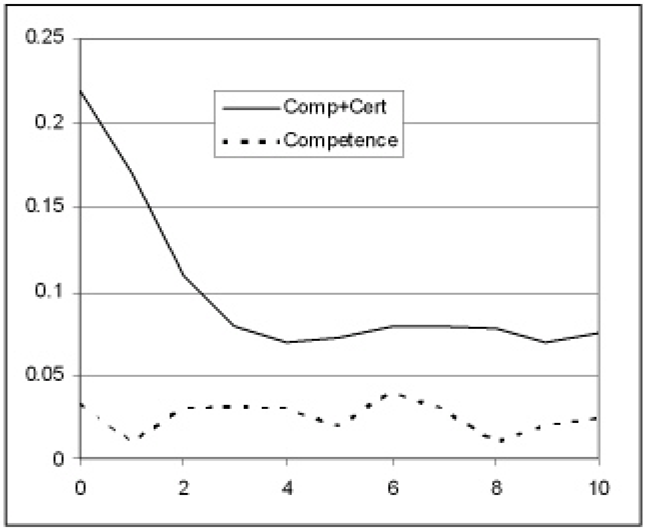


Figure 7: Attractiveness (measured as relative frequency per cycle when alcohol is actual goal) of julihuana when it only has an effect on competence and when it has an effect on uncertainty, too.

May it be, that this is the main reason for the attractiveness of drugs for writers and composers? One of the reasons to create something is to bring order to a world which is perceived as chaotic. The lowered resolution level under the influence of a drug decreases the chaotic character of the world and therefore satisfies the need for certainty. And therefore a drug relieves the "suffering about the world", which is characteristic for many intellectuals.

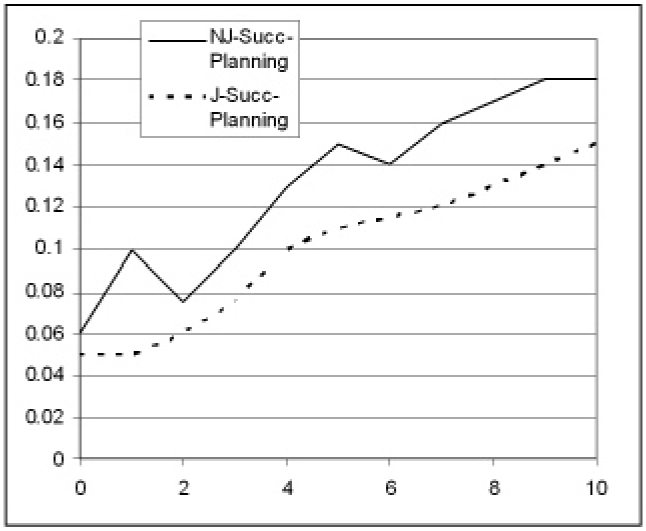


Figure 8: Successful planning activities (relative frequency per cycle) of the J-mice and the Non-J-mice.

Cognitive Effects

To take alcohol has an impact on the cognitive abilities of the mice. Cognition becomes rough, perceiving and planning on a low resolution level become deficient. Figure 8 shows the number of successful planning activities with the J-mice and the Non-J-mice. The relative number of successful plans diminishes significantly with the use of alcohol.

This is due to two factors. One is, as above mentioned, the low resolution level of cognitive processes, which produces bad plans, which will not work. The other one is the lower sustainability of the planning activities with the J-mice. They loose the hope for success earlier than the Non-J-mice, because the level of the competence tank is generally lower for the J-mice (see figure 10).

Planning normally means to construct a branching "tree" of operations or locomotions to fill the gap between the starting point and the respective goal. A branch in this tree is abandoned if there seems not to be any hope that it can be extended further in the direction of the goal. In the mice such a decision is based on the competence level. As the level of the competence tank is lower with the J-mice than with the Non-J-mice, the J-mice have the overall tendency to give up planning too early.

Emotional Effects

Rough and undifferentiated perception is one of the reasons for the J-mice to be more aggressive than the Non-J-mice. On figure 9 this is visible. This difference is due to the fact that the Julihuana-mice simply fail to perceive the strength of their respective opponent because of the low resolution level and therefore become more often entangled in a fight.

Additionally when drunken the J-mice simply felt stronger than their respective opponents and therefore were more daring and incautious.

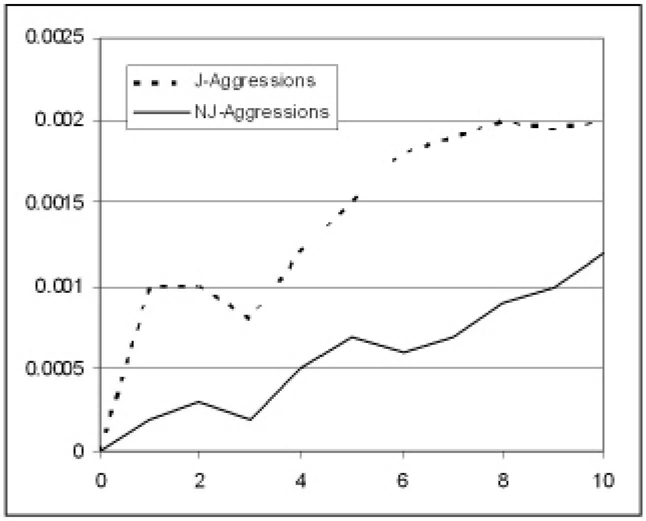


Figure 9: Aggressions in the population of the J- and the Non-J- population (relative frequency per cycle).

But when they were not drunken, and this was the case even for the J-mice most of the time, the J-mice felt less strong, as in this state they noticed their decreased abilities. Figure 10 shows the averaged desire for competence (inverse to the competence feeling) for the J-mice as compared to the Non-J-mice. The J-mice felt generally weaker (higher need for competence) than the mice without access to julihuana.

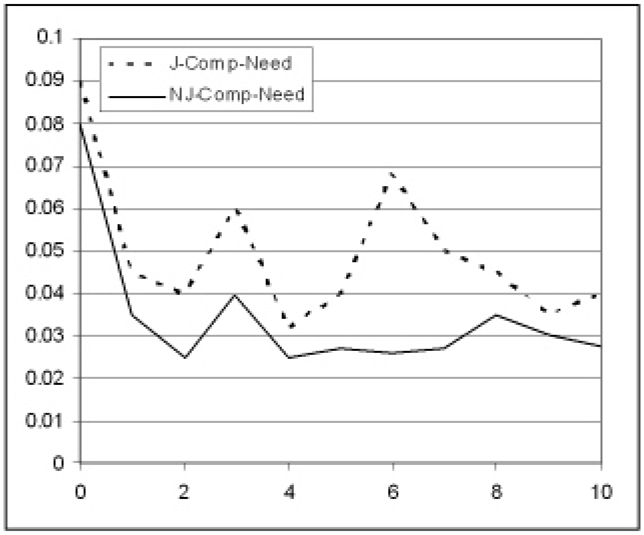


Figure 10: Desire for competence in the J- and Non-J-population.

Summary of the Results

"What I cannot create, I do not understand!" said Richard B. Feynman. We tried to simulate the effects of the consumption of alcohol on psychic processes simply by introducing a beverage for simulated agents, which effects the level on the "competence tank". This produced a number of results which are in good accordance with what is known about the effects of the use of alcohol with humans. The social contacts of the mice diminish and their ability to organize extended actions, too. Additionally the mice became both: more daring when drunken and more cautious and anxious when not drunk.

As we originally not constructed the Psi-theory to cover the phenomena of the consumption of alcohol we consider the success in reproducing these effects as a validation of the theory.

References

Aristoteles (Ed 1995). Über die Seele (About Soul). In Aristoteles. *Philosophische Schriften*, Band 6. Darmstadt (Germany): Wissenschaftliche Buchgesellschaft.

Bach, J. (2008). *Principles of Synthetic Intelligence Psi: An Architecture of Motivated Cognition.* Oxford Series on Cognitive Models and Architectures, New York: Oxford University Press

Boulding, K. E. (1978). *Ecodynamics.* Beverly Hills: Sage.

Dörner, D. (1999): *Bauplan für eine Seele* (Blueprint for a Soul). Reinbek (Germany): Rowohlt.

Dörner, D., Bartl, Ch., Detje, F., Gerdes, J., Halcour, D., Schaub, H. & Starker, U. (2002): *Die Mechanik des Seelenwagens.* (The Mechanics of the Soul-Waggon). Bern (Switzerland): Huber.

Feuerlein, W. (1998): Alkoholismus - Mißbrauch und Abhängigkeit. (Alcoholism – Abuse and Dependency) Stuttgart: Thieme.

Hagg, Julia (2008). *Vorsicht! Betrunkene Mäuse!* (Attention! Drunken Mice!) Diploma Thesis, Institut für Theoretische Psychologie, Bamberg (Germany): Universität Bamberg.

Lange-Eichbaum, W. & Kurth, Wolfram (1986): Genie, Irrsinn und Ruhm (Genius, Madness and Fame). München Basel: Ernst Reinhardt.

Lindenmeyer, J. (2005): Alkoholabhängigkeit (Alcohol-Dependency). Göttingen: Hogrefe.

Newell, A. & Simon, H. A. (1972). *Human Problem Solving.* Prentice Hall: Englewood Cliffs, N.J..

Schuckit, M.A. (1994): Alcohol and Depression: A Clinical Perspective. Acta Psychiatrica Scandinavica 322, Suppl. 28 - 32:

Gaze-following and awareness of visual perspective in chimpanzees

Anthony M Harrison (anthony.harrison.ctr@nrl.navy.mil)
J. Gregory Trafton (greg.trafton@nrl.navy.mil)
Naval Research Laboratory, Washington DC

Abstract

Recent research suggests that chimpanzees are capable of level 1 perspective taking (Flavell, 1992), but that its expression is limited to situations of increased competition (Brauer, Call, & Tomasello, 2007). We present a model utilizing gaze-following that learns in response to the behavior of a competitor. The model not only learns the proper application of the perspective taking strategy but also the critical spatial characteristics that influence the competitive pressure.

Introduction

Under normal conditions most children will eventually develop a full theory of mind and have full visual perspective taking (Corkum & Moore, 1995,1998; Moll & Tomasello, 2006). Most researchers believe that chimpanzees have neither a full theory of mind nor full visual perspective taking (Povinelli et al., 1994; Tomasello & Call, 1997). Whether chimpanzees have *any* perspective taking ability at all has been subject to some recent debate.

Experimental studies using a variety of paradigms have previously been unable to find strong evidence for perspective taking. In fact, two of the major experimental labs consistently agreed that chimpanzees had no visual perspective taking ability (Povinelli et al., 1994; Tomasello & Call, 1997). However, a novel paradigm suggested that chimpanzees did, in fact, know what others could and could not see (Hare et al., 2000; 2001). In this paradigm a subordinate and dominant chimpanzee competed with each other for two pieces of food, one of which was hidden to the dominant (figure 1, left). Since the subordinate preferred the hidden food, Hare et al. concluded that it was aware of the dominant's visual perspective (2000, 2001).

Unfortunately, in a series of experiments, Karin-D'Arcy and Povinelli (2002) were unable to replicate the original Hare et al. (2000) findings. Karin-D'Arcy and Povinelli used a more stringent coding methodology and suggested that chimpanzees do not understand what others can and cannot see but instead use a variety of competitive strategies to succeed in such scenarios, such as preferring food near barriers.

One difference, however, between the two sets of experiments was the size of the testing area. In the original Hare et al. (2000) experiment, the testing area was 3m x 3m, but Karin-D'Arcy and Povinelli (2002) used a smaller testing area that was 2.6m x 1.8m. It is possible that this size difference could have driven the dynamics and the competitiveness of the situation for the chimpanzees. For example, in a smaller area, it is possible that, since the submissive was released before the dominant, the submissive was able to quickly grab the food, making the use of visual perspective taking less relevant. In the larger area, the competitive aspects of the area could make a quick grab of the food less effective since it would take the submissive longer to approach the food.

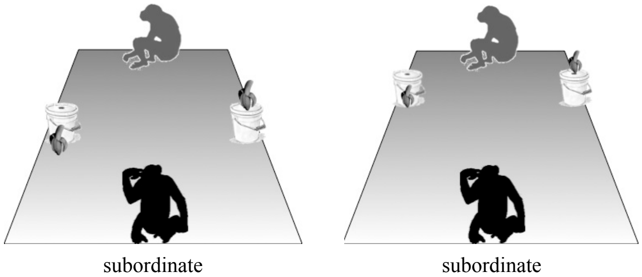


Figure 1. Dual-food layout for Brauer, et al (2007). Visible and hidden food nearer subordinate (left), and further away (right).

Brauer, Call, and Tomasello (2007) tested this idea by making several changes to their experimental paradigm, using the stronger methodology that Karin-D'Arcy and Povinelli (2002) suggested and manipulating the spatial characteristics and therefore the competitive nature of the situation. Specifically, Brauer et al. (2007) manipulated the location of the food to be nearer or farther away from the submissive (figure 1). They found that in the less competitive situation where the food was closer to the submissive, chimps did not seem to use visual perspective-taking. However, in the more competitive situation where the food was further away, chimps did seem to use visual perspective taking, preferring to pursue the hidden food (figure 2).

While the empirical data suggests that chimpanzees do have some form of visual perspective taking, it is unclear what degree of visual perspective taking is needed. Other researchers have suggested different levels of visual perspective-taking, mostly focused around the development of human children (Flavell, 1992). This work suggests that human infants, by one year of age, can follow another's gaze to targets (Corkum & Moore, 1995; 1998). By 12-15 months, a child knows a great deal about what others can and can not see, including (a) that an adult's line of sight is blocked by a screen unless it is transparent or has a window in it (Caron et al. 2002; Dunphy-Lelii & Wellman, 2004); (b) that an adult will not be able to see a target while their eyes are closed (Brooks & Meltzof, 2002); and (c) that an adult can see something that the child can not when the adult looks to locations behind them or behind barriers (Moll & Tomasello, 2004).

Most researchers interpret these findings as evidence of level 1 visual perspective-taking (Flavell, 1992): understanding the content of what a child sees may differ from what another may see. Level 2 visual perspective taking is achieved when a child understands that people can see the same view from different perspectives. After level 1 and 2 visual perspective taking, normally developing human children also achieve a full theory of mind (knowing that others can have different thoughts and beliefs).

Hare, Call, & Tomasello (2001) suggested that chimpanzees are able to engage in level 1 visual perspective taking but not level 2. We modeled level 1 visual perspective taking to determine if it is sufficient to match the data from Brauer et al. (2007). We embed our simulation within a learning framework as well to explore how different competitive strategies can be learned.

Specifically, a model of chimpanzee competitive food foraging was developed within ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004) utilizing the architecture's procedural learning mechanisms and a new gaze-following capability to support level 1 perspective taking.

Experiment

The refined methodology of Brauer et al. (2007) used a testing environment that was 2.5m x 2.6m, with barriers placed at the extreme sides of the cage. In the *near* condition, the barriers were equidistant between the two entrances. For the *far* condition they were moved 0.5m closer to the dominant's entrance. Food pieces were either placed behind the barrier (visible to the subordinate only) or on top (visible to both). On each trial, there could be two pieces of food (one hidden and one visible), one visible or one hidden.

The trial began when the subordinate's door was opened allowing it into the environment. After the subordinate entered the cage, the dominant's door was opened (usually within 2s). The subordinate's food preference was recorded when it made a reaching gesture in the direction of a piece of food before the dominant had approached any barrier.

The single food trials were control conditions testing the possibility that the subordinate might simply prefer food located near barriers (Karin-D'Arcy & Povinelli, 2002). The critical comparison is between the two distance conditions. When the pieces of food were *near* the subordinate, it chose indiscriminately. Because of its head start (~2s), the subordinate could pursue either piece, and was often able to acquire both. However, when the food was closer to the dominant, the subordinate preferred the hidden food almost 2:1 (figure 2).

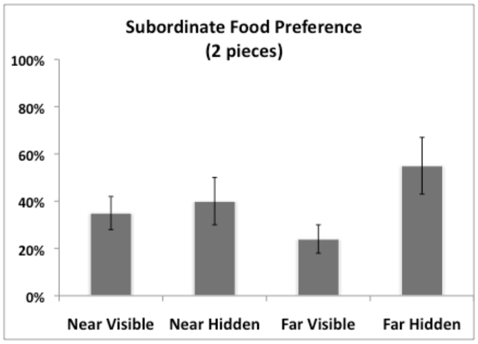


Figure 2. Subordinates prefer hidden food when competitive pressures are greatest (right). Error bars are SE (Brauer, et al, 2007).

Model

Models of both the dominant and subordinate chimpanzees were built in ACT-R (Anderson, et al., 2004). These models were run within the Player/Stage environment (Collett, MacDonald, & Gerkey, 2005) that mimicked the structure of the actual experiment.

As an integrated architecture, ACT-R provides multiple mechanisms for representation and learning. These particular models rely upon ACT-R's procedural memory and learning. At any given time there is a set of productions (if-then rules) that may fire because their conditions match the current external state of the environment or internal state of the model. From this set of competing productions, a single one is selected and fired, ultimately modifying the environment or internal state. ACT-R uses the predefined or learned utilities of productions to determine which will be fired.

To learn production utilities, ACT-R uses an elaboration of the temporal-difference (TD) algorithm (Sutton & Barto, 1998). The elaboration in ACT-R is more applicable for human learning and allows it to be more easily incorporated into a production-system framework (Fu & Anderson, 2006). Briefly, any time reinforcement is given (e.g., a banana eaten or physical punishment) the reinforcement value is propagated back in time through the rules that had an impact on the model receiving that reinforcement. Reinforcements (either positive or negative) gradually shift utility values and therefore the relative probability that a particular production will be selected over others within a set of competitors.

The application of ACT-R to non-human cognition presents many challenges. Even though chimpanzee cognition shares many similarities to that of humans, the architecture may still provide too much capability. Because of this we intentionally used the least-common-denominator in these models. The chimpanzee models make no use of declarative encoding or retrievals, nor does it engage in any imaginal operations. The models are driven predominantly by reactive productions and rely upon an impoverished goal representation (merely storing what target to pursue).

Gaze-following

To implement gaze-following in ACT-R, a new set of optional constraints were introduced to the visual search mechanism. ACT-R's basic visual search mechanism takes a request to find a percept matching some set of features (e.g. where is a red object?). The possibly features include both visual properties (i.e. color, size) and limited spatial information (e.g. nearest the current focus of attention). The location of the first matching object is returned to the model allowing it to attend to that location and encode the actual visual representation of that percept.

Within this mechanism, gaze-following was implemented as a directed visual search along a retinotopic vector. Specifically, instead of returning the first matching location in search, the full set of matches is passed through a secondary filter. This filter merely sorts the locations by their distance from the retinotopic vector. Given a starting point and either an angle or an end point, the visual search returns the location on an object somewhere along that line within a specified tolerance. Knowing the visual location of the dominant chimp (A in figure 3) and the food (C in figure 3), the subordinate performs a visual search for any object along the line segment AC. Finding the barrier (B), the subordinate can (generally) assume that the food is not directly visible to the dominant.

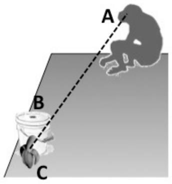


Figure 3. Retinotopic searches to find objects 1) between A and C or 2) along the ray starting at A.

This simple mechanism allows the visual system to find objects along a gaze line, or any potential obstructions between two points. While this mechanism is not accurate for all gaze-directions (particularly as the ray approaches the viewer), they are adequate for basic searches. More advanced gaze-following is addressable by having the model perform more detailed processing of the returned visual locations and the actual visual percepts at those locations, such as testing the distance, size, or opacity of an obstruction. Given the nature of the experimental environment, these higher-level strategies were not implemented.

Model Structure

The dominant and subordinate models are composed of the same constituent parts. Each model performs a full environment scan from its current position, looking not only for the food, but also the other chimpanzee and the buckets. The targets are evaluated to determine which should be pursued.

Environmental Scan The environmental scan is a rapid visual search of the environment that attends to all visible objects. If the object is a piece of food, a bucket, or another chimpanzee, the first occurrence is retained in the model's limited goal representation. If no objects are found, the model physically rotates its body to get a different view of the environment.

Target & Strategy Evaluation Once a target has been attended to it must be evaluated. For the dominant model this is simple: if it's food, pursue it, otherwise keep looking. The subordinate has more to consider. First, the subordinate must determine whether the food is *near* or *far*. Once classified, the subordinate can then choose which strategy to use. It can either try to make a mad-dash for the food (grab-and-go), or use gaze-following to ensure that the coast is clear. If the subordinate chooses grab-and-go, it runs the risk of contention with the dominant, particularly if the food is far away. For gaze-following, the subordinate will use the location of the dominant's head and the target to find any intermediate object that may be a visual barrier. If a visual barrier is found, the subordinate assumes the dominant cannot see the target and will pursue it. If no barrier is found, the subordinate rescans the environment ignoring the rejected target.

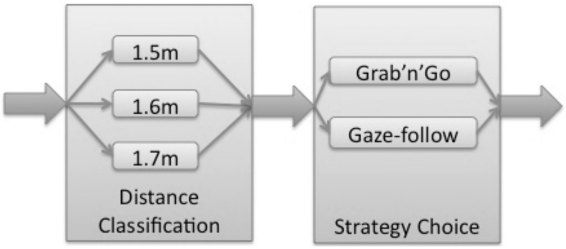


Figure 4. Two choice points for the subordinate model. The model must learn which distance threshold to use for classification and then which strategy to use.

Target Pursuit Since the Brauer, et al. experiment recorded food preference based on the initial reaching behavior, models' food preferences were recorded immediately after evaluation. The full models, however, are able to navigate in the environment, grab food and even strike each other.

Model Assumptions and Parameter Selection

At their heart all models are simplified abstractions of their respective phenomenon. Simplifications can be for reasons of computational tractability, interpretability, or theoretical relevance. The models described here must operate at a high-level of fidelity in order to capture the embodied nature of the task. The computational costs of the embodied simulations required a handful of simplifying assumptions.

Environmental Assumptions

In the actual experiment, doors into the experiment cage were opened allowing the chimpanzees to enter the space. After the subordinate entered, the dominant's door was opened, typically after around 2 seconds. Lacking doors in the simulation, each model was "beamed" into the experiment space. The delay between the subordinate and the dominant was fixed at 2 seconds. Since the subordinate's food preference is only recorded if it is made before the dominant makes one, this delay acts as a scalar for the food preference measure. Increasing the delay allows the subordinate more time to choose, increasing the absolute food preference scores.

Model Assumptions

Learning Brauer, et al (2007), Hare et al. (2000; 2001) and Karin-D'Arcy & Povinelli (2002) all noted a lack of learning within their studies. All concluded that the preferences and skills exhibited had developed prior to testing. For the models to exhibit these behaviors they either have to be hand tuned by the modeler or they must be given sufficient training prior to testing. Having an architecture that can learn allows us to avoid the problem of custom tuned models. Each model was run through a series of learning trials, which consisted of ten sets of the full factorial design of the experiment (e.g. single & dual pieces of food at both the near & far distances), for a total of 60 trials. This was a rough surrogate for the individual's life experience with competitive food foraging.

Additionally, since gaze-following is learned over time in humans (Corkum & Moore, 1995), initial utilities of the gaze-following productions were lowered below those of the grab-and-go productions (to -1.5). This provides an early bias towards grab-and-go, delaying the onset of gaze-following, potentially providing the model with the time necessary to learn the distance classifications.

Reinforcement Probabilities In order to learn from these trials, the models must receive some reinforcement based upon their target choices. However, since the trials terminate after target choices are made they normally wouldn't receive any reinforcement. One alternative would be to run each trial to completion (after either has actually consumed the food or been hit). Unfortunately, full trials, with the possibility of the dominant chasing the subordinate around the cage, are extremely costly computationally (by almost an order of magnitude).

Reinforcements were provided based on the model target choices. When either chooses an uncontested piece of food, it is rewarded. When both the dominant and subordinate decide to pursue the same target there is some chance that the dominant will charge and strike the subordinate. Naturally, as the distance between the target and dominant decreases, the probability that the subordinate will be punished for pursuing that same target increases. All other things being equal, when the distance to the target is equivalent, there is roughly a 50% chance that the subordinate will be able to reach the target first. The chance of being hit is further reduced by the subordinate's two-second head start in the experiment design. The qualitative behavioral pattern (i.e. subordinate preferring hidden food when both pieces are closer to the dominant) holds through probability values where $P(hit|near) < 0.5 \leq P(hit|far) \leq 1$.

Generally speaking, the higher the probability of being hit for any given distance, the more likely the subordinate will select the more conservative gaze-following strategy. The values $P(hit|near)=0.1$ and $P(hit|far)=0.9$ were settled upon after a high-level exploration of the parameter space. Simulations testing the validity of these assumptions using the full trial protocol are ongoing.

Hit Probabilities Reinforcement Values ACT-R's reinforcement learning mechanism relies ultimately on time as its metric (Fu & Anderson, 2006). This forces the modeler to map physical rewards and punishments into a temporal reference frame. For this experiment, the reward for getting a piece of food was set at the average maximum time to complete the task using the gaze-following strategy (4 seconds). The punishment for being hit needs to be greater in magnitude than the food reward in order to pull apart the two primary strategies. Parameter explorations yielded good convergence rates for punishments around 8 seconds.

ACT-R's default utility learning rate of 0.2 was used. The only other parameter modified was the utility noise (0.1), which permits weaker productions to occasionally be selected over their stronger competitors.

Simulation Results

For this model to be a viable account for the subordinate chimpanzee's behavior not only must it fit the aggregate food preference measure, but it must also be able to correctly classify the target distances and prefer the gaze-following strategy for far targets. Because the individual learning histories result in greater downstream behavioral variability, large numbers of models had to be run to arrive at stable results. The results presented here are the derived from 1000 individual model runs.

Distance classification

The key factor in the results presented by Brauer, et al (2007) is that the preference for choosing the hidden piece of food is dependent upon how close the food is to the dominant chimpanzee. While they did not do a full parametric exploration of the factor, the simple difference of half a meter was sufficient to tease apart the behaviors.

Similarly the model had to be able to correctly classify the target distances as near or far. At the distance choice-point (figure 4), three productions are in competition, setting the distance threshold to 1.5, 1.6, or 1.7m. Subsequent productions then classify the target's distance using that threshold. In the simulation, target distances ≥ 1.6m correspond to the *far* condition. Within each model we can simply examine the relative utilities of the distance threshold productions; 41% of the models converged upon the correct threshold of 1.6m, 21% at 1.5m and 14% at 1.7m. The remaining 24% of the models showed no clear preference as the threshold utilities were all within the model's utility noise.

Strategy Selection

When the food is *near*, it is perfectly rational for the subordinate to make a mad-dash for either piece. With the two-second head start, there is little chance that it will be punished. On the other hand, when the food is further away (and closer to the dominant), it makes sense to use the gaze-following strategy even though it takes longer and requires waiting for the dominant to enter the experiment space. If the subordinate were to use grab-and-go for far targets, it would run an increased risk of contention with the dominant, even with its head start. On average gaze-following took 0.75 – 1.5 seconds longer than grab-and-go. While this increase in execution time ultimately reduces the temporally discounted reward, it effectively avoids the much more costly punishment when conflict does occur. Figure 5 shows the percentage of model strategic preferences. The majority of the models preferred grab-and-go when *near* and gaze-following when *far*.

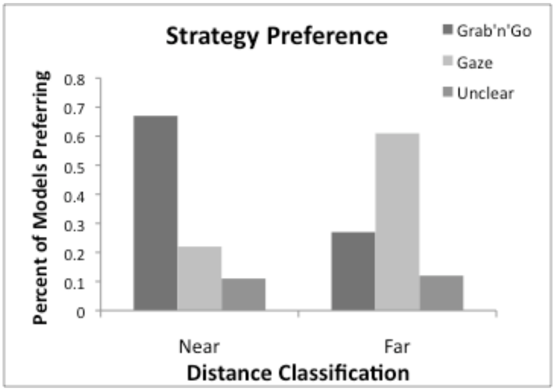


Figure 5. Percentage of models preferring a given strategy for both *near* and *far* target classification.

Model Fit

Even with the model complexity and resulting downstream behavior variability, the fits were strong (RMSE=7.2%, R^2=0.96). The qualitative pattern (i.e. preference for hidden food when far and equivalence for near) holds across the majority of the hit probability ranges discussed earlier.

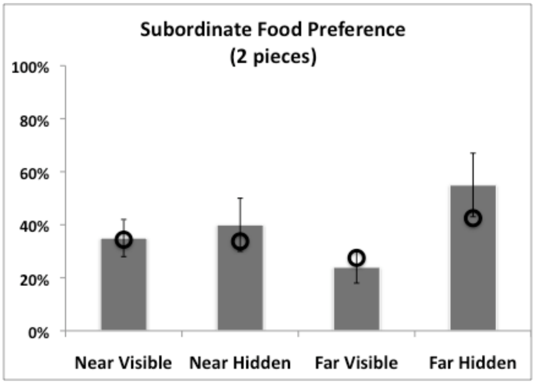


Figure 6. Model (circles) fit to Brauer, et al (2007) data. RMSE=7.2%, R^2=0.96

Distance & Strategy Interactions

The variability in the behavior of any given subordinate model is a direct result of its experiences with the dominant model. That some learned the wrong distance threshold or frequently choose the wrong strategy is hardly surprising. Looking more closely at these models is particularly informative from a rational analysis perspective. All of the models that settled on the 1.5m distance threshold used the gaze-following strategy exclusively for *far* targets (which would have been virtually all of the them). Similarly, over half the models that settled on 1.7m as the distance threshold preferred gaze-following when targets were both *far* and *near*. These overly conservative models were able to stabilize in their patterns because there was no disincentive for misclassifying targets as *far* only *near*, particularly since they could rely upon gaze-following to compensate for incorrect distance classifications.

Discussion

The simulation presented provides a process model of chimpanzee competitive food foraging that combines the awareness that individual visual experiences are different (i.e. Flavell, level 1) and a simple gaze-following mechanism. Leveraging the existing reinforcement-learning component in ACT-R, the model learns to prefer the more conservative gaze-following strategy when the risk of punishment is increased (i.e. when the food is closer to the dominant). The model shows that its "awareness of the other's visual experience" need not entail full visual perspective taking (Hare, Call, & Tomasello, Animal Behaviour, 2001). Knowledge of the particular spatial relationships that the dominant is experiencing are also unnecessary.

Obviously this does not preclude the possibility that chimpanzees possess level 2 skills. It is worth considering how a model of full perspective taking would perform in this situation. Such a model was actually developed before the one reported here. It performed egocentric transformations of its own perspective, aligning them with the perceived position and orientation of the dominant (e.g. Hegarty & Waller, 2004). This model was able to learn the same qualitative behavioral pattern, but at an increased cost. Perspective transformations are particularly costly in terms of time; often taking 2-4x longer than gaze-following depending on assumptions of representational capacity and mental transformation rates.

What is perhaps more interesting is that if full perspective taking and gaze following are allowed to compete, gaze following is consistently preferred. While gaze following isn't as accurate at assessing visibility, it is accurate enough within the confines of the task and significantly faster. Given this, it is unlikely that one could find evidence of full perspective taking in the current experimental paradigm.

These models arose out of our growing interest in embodied cognition. While fully situating a model in an environment makes some tasks quite simple (i.e. inferring intent based on gaze direction), it comes at the cost of requiring higher fidelity models and simulations. This higher fidelity brings with it increasingly complex dynamic interactions between the model and environment (including other intelligent agents). Our work with human-robot interaction has shown us that these dynamic interactions cannot be ignored.

Conclusions

A computational learning model was developed that is able to effectively reason about what another can and cannot see. This embodied model is able to learn and exploit regularities in the environment (target distances) to adapt to a competitor's behavior. The model is able to do this with only a basic gaze-following mechanism instead of relying upon full visual perspective taking (Hare, Call, & Tomasello, 2001). This mechanism, implemented as a general directed visual search, provides an important developmental step towards the development of theory-of-mind (Baron-Cohen, 1995; Butterworth, 1991).

Acknowledgments

This work was performed while the first author held a National Research Council Research Associateship Award and was partially supported by the Office of Naval Research under job order number N0001409WX20173 awarded to the second author. The views and conclusions contained in this document should not be interpreted as necessarily representing official policies, either expressed or implied, of the U.S. Navy.

The models and software are available for download at http://anthonymharrison.com/chimpanzee-gaze-following-and-visual-perspective-awareness/.

References

Anderson, J., Bothell, D., Byrne, M., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111* (4), 1036-1060.

Baron-Cohen, S. (1995). The eye direction detector (EDD) and the shared attention mechanism (SAM): Two cases for. *Joint attention: Its origins and role in development*.

Bräuer, J., Call, J., & Tomasello, M. (2007). Chimpanzees really know what others can see in a competitive situation. *Animal Cognition, 10* (4), 439-448.

Butterworth, G. (1991). The ontogeny and phylogeny of joint visual attention.

Call, J., Hare, B., & Tomasello, M. (1998). Chimpanzee gaze following in an object-choice task. *Animal Cognition, 1* (2), 89-99.

Caron, A.J., Kiel, E.J., Dayton, M., & Butler, S.C. (2002). Comprehension of the referential intent of looking and pointing between 12 and 15 months. *Journal of Cognition and Development*, 3(4), 445–464.

Collett, T., MacDonald, B., & Gerkey, B. (2005). Player 2.0: Toward a practical robot programming framework. *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005)* .

Corkum, V., & Moore, C. (1995). Development of joint visual attention in infants. In C. Moore & P. J. Dunham(Eds.), *Joint attention: Its origins and role in development* (pp. 61–83). Hillsdale, NJ: Erlbaum.

Corkum, V., &Moore, C. (1998). The origins of joint visual attention in infants. *Developmental Psychology*, 34(1), 28–38.

Dunphy-Lelii, S., & Wellman, H. M. (2004). Infants' understanding of occlusion of others' line-of-sight: Implications for an emerging theory of mind. *European Journal of Developmental Psychology*, 1(1), 49–66.

Flavell, J. H. (1992). Perspectives on perspective-taking. In H. Beilin & P. B. Pufall (Eds.), *Piaget's theory: Prospects and possibilities. The Jean Piaget symposium series* (Vol.14, pp.107–139). Hillsdale, NJ: Erlbaum.

Fu, W.-T., & Anderson, J. (2006). From Recurrent Choice to Skill Learning: A Reinforcement-Learning Model. *Journal of experimental psychology: general , 135* (2), 184-206.

Hare, B., Call, J., & Tomasello, M. (2001). Do chimpanzees know what conspecifics know? *Animal Behaviour , 61* (1), 139-151.

Hare, B., Call, J., & Tomasello, M. (2006). Chimpanzees deceive a human competitor by hiding. *Cognition , 101* (3), 495-514.

Hare, B., Call, J., Agnetta, B., & Tomasello, M. (2000). Chimpanzees know what conspecifics do and do not see. *Animal Behaviour , 59* (4), 771-785.

Hegarty and Waller. A dissociation between mental rotation and perspective-taking spatial abilities. *Intelligence* (2004) vol. 32 pp. 175-191

Karin-D'Arcy, M., & Povinelli, D. (2002). Do chimpanzees know what each other see? A closer look. *International Journal of Comparative Psychology , 15*, 21–54.

Moll, H., & Tomasello, M. (2004). 12- and 18-month-old infants follow gaze to spaces behind barriers. *Developmental Science*, 7(1), F1–F9.

Moll, H., & Tomasello, M. (2006). Level 1 perspective-taking at 24 months of age. *British Journal of Developmental Psychology*, 24, 603-613.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction.* Cambridge, MA: MIT Press.

Tomasello, M. (2003). Chimpanzees understand psychological states – the question is which ones and to what extent. *Trends in cognitive sciences , 7* (4), 153-156.

Tomasello, M., Call, J., & Hare, B. (1998). Five primate species follow the visual gaze of conspecifics. *Animal Behaviour , 55* (4), 1063-1069.

A Decision Making Model Based on Damasio's Somatic Marker Hypothesis

Mark Hoogendoorn[1], Robbert-Jan Merk[1,2], and Jan Treur[1] ({mhoogen, treur}@cs.vu.nl, merkrj@nlr.nl)

[1]Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
[2] National Aerospace Laboratory (NLR) , Training, Simulation & Operator Performance
Anthony Fokkerweg 2, 1059 CM Amsterdam, The Netherlands

Abstract

In this paper a computational decision making model is presented based on the Somatic Marker Hypothesis. The use of the model is illustrated for the domain of fighter pilot decision making. Hereby, simulation runs have been performed upon this scenario, and the results thereof have been formally verified based upon properties inspired on Damasio's Hypothesis.

Keywords: decision making, experience, somatic marker.

Introduction

Decision making usually involves expectations about possible consequences of decision options and uncertainty about them. Traditionally the literature on decision making was dominated by the Expected Utility Theory; e.g., von Neumann and Morgenstern, 1944; Friedman and Savage, 1948; Arrow, 1971; Keeney and Raiffa, 1976. Here, decision making takes place by calculating expected utilities for all of the options and choosing the option with highest expected utility. The expected utilities themselves are determined based on the probabilities of the possible outcomes for the option when chosen, and the gain or loss for that outcome, thus founding the approach in probability theory. This approach to decision making can be considered to aim for an idealised rational approach, where, for example, emotions or biases play no role. As a model for practical human decision making the Expected Utility Theory has been strongly criticized, as humans are bad in estimating probabilities, and also may allow emotions and biases to play a role in a decision making process, as is found in several experiments; e.g., (Tversky and Kahneman, 1974; Kahneman and Tversky, 1979).

Contrasting with the aim of the Expected Utility Theory to ban emotions from decision making, Damasio (1994) observed surprisingly bad decision making behaviour in patients with damage of brain regions related to body mapping and regulation and feeling emotions (patients with certain kinds of prefrontal damage and with compromised emotions). They often keep on considering different options without choosing for one of them. This has led Damasio to the view that decision making inherently depends on emotions felt, which relate to sensed body states (Damasio, 1994). His theory is known as the *Somatic Marker Hypothesis*.

In this paper a computational decision making model is presented which draws inspiration from the Somatic Marker Hypothesis. The main purpose of this model is to create agents which show realistic human behaviour, and not to replicate the precise human decision process. This makes the Somatic Marker Hypothesis a suitable choice, as it provides a reasonable amount of detail on these decision processes. Although the validity of the theory is sometimes doubted (see e.g. Dunn, Dalgleish and Lawrence, 2005), it can still considered to be a useful source of inspiration for the development of agents for the aforementioned purpose. The use of the model is illustrated for the domain of fighter pilot decision making. This extends the work as presented in (Hoogendoorn *et al.*, 2009) by having a more sophisticated version of somatic markers (including specific goals and tradeoffs between such goals), as well as a case which addresses more interesting aspects of the decision making process. First, the Somatic Marker Hypothesis is explained in more detail, after which the computational model is described. Thereafter, simulation results are presented, including formal properties that have been verified against the generated results.

Decision Making and Experience

The Somatic Marker Hypothesis provides a theory on decision making which dedicates a central role to experienced emotions. Damasio explains the name of his theory as follows:

> 'Because the feeling is about the body, I gave the phenomenon the technical term *somatic* state ("soma" is Greek for body); and because it "marks" an image, I called it a *marker*. Note again that I use *somatic* in the most general sense (that which pertains to the body) and I include both visceral and nonvisceral sensation when I refer to somatic markers.' (Damasio, 1994, p. 173)

This theory consists of two main ideas: (1) the way in which somatic markers affect decisions, and (2) the way in which somatic markers depend on past experiences. Concerning (1), if a decision is to be made between options which can lead to potentially harmful or advantageous outcomes, each of such options induces a somatic response which is experienced as a feeling and used to mark the option outcome, thus signalling its danger or advantage. For example, when a negative somatic marker is linked to a particular option outcome, it serves as an alarm signal for that particular option. Similarly, a positive somatic marker serves as an encouragement to choose that option. Damasio describes the use of a somatic marker in the following way:

> 'the somatic marker (..) forces attention on the negative outcome to which a given action may lead, and functions as an automated alarm signal

> which says: Beware of danger ahead if you choose the option which leads to this outcome. The signal may lead you to reject, *immediately*, the negative course of action and thus make you choose among other alternatives. The automated signal protects you against future losses, without further ado, and then allows you *to choose from among fewer alternatives*' (Damasio, 1994, p. 173)
>
> 'In short, *somatic markers are a special instance of feelings generated from secondary emotions*. Those emotions and feelings *have been connected by learning to predicted future outcomes of certain scenarios*. When a negative somatic marker is juxtaposed to a particular future outcome the combination functions as an alarm bell. When a positive somatic marker is juxtaposed instead, it becomes a beacon of incentive. This is the essence of the somatic marker hypothesis. (..) on occasion somatic markers may operate covertly (without coming to consciousness) and may utilize an 'as-if-loop'. (Damasio, 1994, p. 174)

Concerning (2), the way in which somatic markers are associated to decision options in a given situation depends on previous experiences with options chosen in similar circumstances. For example, the pain or joy experienced as a consequence of the outcome for a certain option that was chosen in the past has been stored in memory and automatically pop up (are felt again) when similar circumstances and options may occur. How somatic markers relate to past experiences is described as follows:

> 'Somatic markers are thus acquired through experience, under the control of an internal preference system and under the influence of an external set of circumstances which include not only entities and events with which the organism must interact, but also social conventions and ethical rules. (Damasio, 1994, p. 179)

This element of Damasio's theory shows how based on experience 'intuition' or 'gut feeling' is created which aids the decision process in an automatic manner. This makes the theory useful for decision processes where such aspects play an important role, which is the case for the domain of pilot behaviour considered here.

Model Description

The model has been defined as a set of temporal relations between properties of states. A state property is a conjunction of atoms or negations of atoms that hold or do not hold at a certain time. The exact choice for what atoms to use depends on the actual model and domain and is defined by an ontology for that model. To model dynamics, transitions between states are defined.

In order to obtain an executable formal model, the states and temporal relations between them have been specified in LEADSTO, a temporal language in which the temporal relations can be defined in the form of rules that can be executed. Let α and β be state properties. In LEADSTO specifications the notation $\alpha \twoheadrightarrow_{e, f, g, h} \beta$, means:

> *if state property* α *holds for a certain time interval with duration* g, *then after some delay (between* e *and* f *state property* β *will hold for a certain time interval* h.

For more details of the LEADSTO format, see (Bosse, Jonker, van der Meij & Treur, 2007). As all of the temporal relations used in the model are of the form $\alpha \twoheadrightarrow_{0,0,1,1} \beta$, the notation $\alpha \twoheadrightarrow \beta$ will be used instead.

The Decision Making Process

The central process in the model is the Decision Making process. Its input is the current situation, the list of possible options from which one option is to be selected and the somatic markers. The situation is represented by an atom supplied by the environment and can be seen as the result of the agent's perception of its environment. For example, in the case described earlier, the agent could encounter an enemy fighter from its side. In the model the Decision Making process would receive the atom observed(enemy_from_side).

In the Decision Making process for each option the option preference, a real number between 0 and 1, is determined. Both somatic markers and rational utility values are used to calculate option preferences. The option with the highest option preference is then selected for execution.

Execution of the selected option will result in some change in the environment of the agent and the agent will observe this outcome. This outcome is then evaluated, resulting in a set of real numbers between 0 and 1, one per goal, where a higher value means a more positive evaluation. These evaluation values are then used to adapt the appropriate somatic markers associated with each goal. The selected option itself is also input for the evaluation process, as the evaluation is about the consequences of this selected option. The value of the outcome evaluation is then used to adapt the somatic markers the agent has. In subsequent decisions the updated somatic markers are used.

Step 1: Somatic Evaluation

The purpose of the Somatic Evaluation process is to assign a real value between 0 and 1 to each option. This value, the somatic evaluation value, is determined per option by adding the weighted values of the different types of somatic markers associated with the option and current situation. For each goal the agent has, there is a different type of somatic marker. There is also a weight value for each type of somatic marker with which the value of the somatic marker is multiplied. This way, it is possible to vary the influence each type of somatic marker has on the final somatic evaluation value, which can be used to represent personal characteristics. The formula for determining the somatic evaluation value is:

$$\mathrm{sev}(O)_t = w_1 \cdot \mathrm{smv}(G_1,O)_t + w_2 \cdot \mathrm{smv}(G_2,O)_t + \ldots + w_n \cdot \mathrm{smv}(G_n,O)_t$$

where $\mathrm{sev}(O)_t$ is the somatic evaluation value for option O at time t, $\mathrm{smv}(G_i, O)_t$ the value for the somatic marker associated with goal G_i at time t, w_i the weight for goal G_i. Note that the somatic markers are those for the current situation. The weights add up to 1, so that the somatic evaluation value remains within 0 and 1. For the sake of brevity the temporal properties defining this process has been omitted.

Step 2: Option Elimination

The next step is the Option Elimination process. All the atoms of the form somatic_evaluation_value(O, V), generated in the

Somatic Evaluation process, are transformed into atoms of the form remaining_somatic_evaluation_value(O, V) if V does not fall below a threshold value. All other atoms of that form are discarded, effectively eliminating the options associated with those atoms. In P1 and P2 this process is formalised.

P1 somatic_evaluation_value(O, V) & value(threshold, Th) & V ≥ Th → remaining_somatic_evaluation_value(O, V) & somatic_evaluation_ ended

P2 remaining_somatic_evaluation_value(O, V) & not(decision_making_ ended) → remaining_somatic_evaluation_value(O, V)

Step 3: Rational Analysis

The next subprocess is the Rational Analyis. In this process a rational utility is calculated for each option for which the atom remaining_somatic_evaluation(O,V) holds. According to Damasio (1994), the rational phase is partly influenced by the preceding somatic marking. For this reason the assumption is that the remaining somatic markers are used in determining the outcome of the rational phase, which is a number of utility values.

In the design of the model there are atoms of the form belief(utility(S, O, U)) which couple each situation S and each option O with a real value U between 0 and 1, indicating the utility for that option in that particular situation. More elaborate utility functions are certainly possible but fall outside the scope of this paper.

P3 remaining_somatic_evaluation_value(O, V) & belief(current_situation(S)) & belief(utility(S, O, U)) → option_utility(O, U)

P4 remaining_somatic_evaluation_value(O, V) & option_utility(O, U) & value(rational_ratio, R) → option_preference(O, R * U + (1-R) * V)

Property P3 defines the determination of an option utility for each remaining option. This consists of attaching to each remaining option the utility that the agent believes is the expected utility for that option considering the current situation. In P4 for each remaining option the option preference is determined. This value is taken as a weighted average between the somatic evaluation value and the option utility. The parameter rational ratio determines what weight the option utility has in determining the option preference. In other words, a higher rational ratio shifts the Decision Making process more towards the rational side, while a lower rational ratio makes the Decision Making process more intuitive.

After P4 has been applied, the selected option is determined by taking the option with the highest option preference. The temporal properties that define this final selection are not included in this paper for the sake of brevity.

The selected option is then executed, which results in some outcome that is used for adapting the somatic markers.

Adaptation of the Somatic Markers

As Somatic Marking is a process rooted in experience, the model includes a mechanism for adapting the somatic markers according to the evaluations of outcomes that result from the execution of the selected option. This mechanism consists of a update function that takes both previous and current experiences in account. An update function described in (Jonker and Treur, 1999) has been chosen to represent the Somatic Marker adaptation mechanism. This is a typical update function, similar to many other kinds of update function that are regularly used for modelling dynamics but it is certainly possible to use similar functions. The following formula describes the update function as used in the model:

$$smv(G, O)_t = (1\text{-}d) \cdot smv(G, O)_{t\text{-}1} + d \cdot ev(G,O)_{t\text{-}1}$$

In this formula, the variable $smv(G, O)_t$ is the value of the somatic marker of option O associated with goal G at time t. The variable $ev(G, O)_t$ is the evaluation value, a real value between 0 and 1. The parameter d is a real value also between 0 and 1 which determines the decay of the memory of previous experiences . A high value for d will cause the somatic markers to rapidly change in accordance with the evaluation values. In other words, the parameter d determines to what degree previous experiences are retained in relation to new experiences. A lower value for d will result in a more stable memory of experiences, while a higher value for d results in a somatic marker that is heavily influenced by recent experiences.

Determining the evaluation value is based on the concept of a body state. In (Damasio, 1997, p. 180), Damasio states that

> ‘At the neural level, somatic markers depend on learning within a system that can connect certain categories of entity with the enact-ment of a body state, pleasant or unpleasant.’

So it appears that the body either reacts positively or negatively in response to the outcome of an action. The precise dynamics of what body state is generated depends on innate dispositions (primarily survival related), and social conditioning. In the model this is represented by a number of atoms of the form resulting_body_state(G, Oc, V), one for each goal-outcome combination.

The following LEADSTO rules show how the somatic marker adaptation is modelled.

P5 belief(outcome(Oc)) & belief(current_situation(S)) & belief(selected_option(O)) & resulting_body_state(Oc, G, V) → evaluation(G, O, V)

P6 evaluation(G, O, Ev) & somatic_marker(G, S, O, Smv) & value(decay_parameter, D) → updated_somatic_marker(G, S, O) & new_somatic_marker_value(G, S, O, (1-D) * Smv + D * Ev)

Each time the agent observes the outcome of an option it executed, it determines a resulting body state. In P5, the value v of the resulting_body_state relevant to the current outcome of an executed option is used as evaluation value in the update function to modify each somatic marker belonging to that option. This way, the agent learns from its experiences.

Case Study

In order to test the model, a case has been constructed that represents a simplified environment from the domain of fighter airplane combat. In this case there is a single fighter, controlled by an agent, which is flying some kind of mission. Its goal is to arrive at his target. However, at some point it detects an enemy aircraft. This forces the agent to make a decision on how to deal with this situation, which is done by an implementation of the model described in this paper.

There are 3 different situations that the agent can encounter: the enemy approaches from the front, the side or from behind. In this case, the agent has four options to deal with these situations:

1. The agent can continue its flight in order to reach his target.
2. The agent can engage the enemy
3. The agent can turn around and return to base.
4. The agent can take an detour to its target, which requires it to fly over the enemy anti-air position.

The outcome of the execution of one of these options depends on the current situation and is probabilistic determined. For example. executing the option engage_enemy in the situation enemy_from_behind has a 30% chance of the agent being shot down, a 50% chance of the agent defeating the enemy and reaching the target and a 20% chance of defeating the enemy and being force to return to base. Appendix A[†] gives more details on the case and the reasoning behind the choices being made.

In the next two sections the choices for determining the utility values and resulting body states are explained.

The utility for each option in each situation that has been chosen for this case are shown in Table 1.

Table 1: Utilities

		Situation		
		Enemy from side	Enemy from behind	Enemy from front
Option	Continue-mission-direct-route	1	1	0
	Continue-mission-detour	1	1	1
	Engage-enemy	0,5	0,5	0,5
	Return-to-base	0,5	0	1

The reasoning behind this allocation of utility values is that mission success and survival have a higher priority than defeating the enemy fighter. In general the agent has the orders to try to complete the mission and to avoid the enemy fighter and only to engage the enemy fighter if the opportunity to do so is good enough in its own ‘opinion’.

Therefore the *continue-mission* options have high utility values, except when the enemy comes in from the front. In that situation the *continue-mission-direct-route has* low utility, as survivability is important and the agent has to try to avoid the enemy fighter. *Engage-enemy* has always a medium utility, as it is left to the agent’s discretion to choose whether to engage. The utility for *return-to-base* is heavily dependent on the enemy fighter’s angle of approach: if the enemy comes from the front, continuing the mission will be dangerous and so *return-to-base* is a good option, while if the enemy comes from behind, *return-to-base* is a bad option as the agent has the orders to avoid the enemy.

Resulting Body States

Table 2 shows the representation of the resulting body states for each outcome. A value of 1 represents a positive body state, a value of zero a negative body state. The body states are coupled to goals and the allocation of values is based on how good an outcome is for reaching that goal.

Lethality is about defeating the enemy, so all outcomes that include the defeat of the enemy result in a positive outcome. Being shot down is the only way of having a negative body state in regard to survivability, as in all other outcomes the agent survives unharmed. Finally, in this case resource control is mainly about fulfilling the mission objective, so all outcomes in which the target is reached result in a positive body state.

Table 2: Resulting body states

		Goals		
		Lethality	Survivability	Resource control
Outcomes	Shot down	0	0	0
	Back at base	0	1	0
	Reached target	0	1	1
	Enemy defeated & reached target	1	1	1
	Enemy defeated & back at base	1	1	0

Simulation Results

The model described in the previous sections has been used to run a number of simulations, using the LEADSTO software environment as described in (Bosse et al, 2007). An environment and scenario for the agent has been implemented based on the case described earlier. Hereby, all three scenarios as presented before have been addressed.

In order to test whether different weights for somatic markers lead to different behaviour, for four different settings of somatic marker weights simulations have been run. The exact settings are shown in Table 3.

Table 3. Somatic weight settings

Setting	**W(Lethality)**	**W(Survivability)**	**W(resource control)**
1	0,33	0,33	0,33
2	0,50	0,25	0,25
3	0,25	0,50	0,25
4	0,25	0,25	0,50

In setting 1, all types of somatic markers have equal influence in the determination of the somatic evaluation value. In settings 2, 3, and 4 the marker weights for respectively lethality, survivability and resource control are set higher, increasing the influence of the associated somatic markers on decision making.

For all situation-weight setting combination, a simulation has been run. In each simulation the decision making model receives 50 times the same situation to decide on. The results of these simulations have been verified, as shown in the next section. Table 4 shows how many times each option has been selected with different somatic weight settings for the *enemy-from-front* scenario.

Table 4. Option selection in situation enemy-from-front

		Somatic weight setting			
		Setting 1	Setting 2	Setting 3	Setting 4
Option	Continue-mission-direct-route	0	0	0	0
	Continue-mission-detour	3	4	2	4
	Engage-enemy	0	32	0	13
	Return-to-base	47	14	48	33

[†] http://www.cs.vu.nl/~mhoogen/damasio-appendixA.pdf

In this situation, when the somatic markers associated with the lethality goals have a higher weight, the option engage-enemy is selected much more often than with a neutral setting. This also the case to a lesser extent when resource control has a higher weight, as in this situation the option *engage-enemy* leads much more often to the outcome *reached target* than any other option. There is little difference between the results of the neutral setting and setting 3, where survivability has a higher weight, as in the neutral setting *return-to-base* is already predominantly chosen. This is probably due to the allocation of utility values, in which a high emphasis is laid upon survivability.

In Table 5 and 6 the option selection for the other two situations are shown.

Table 5. Option selection in situation enemy-from-side

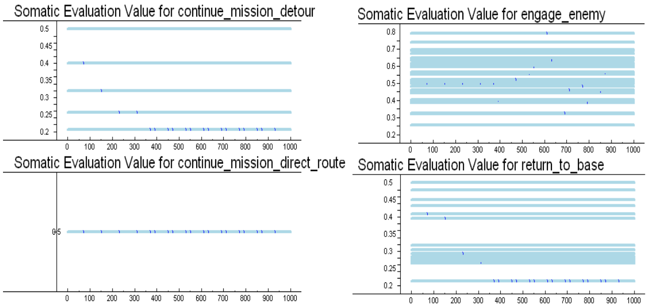
		Somatic weight setting			
		Setting 1	Setting 2	Setting 3	Setting 4
Option	Continue-mission-direct-route	11	36	47	48
	Continue-mission-detour	4	4	3	2
	Engage-enemy	19	9	0	0
	Return-to-base	13	1	0	0

Table 6. Option selection in situation enemy-from-behind

		Somatic weight setting			
		Setting 1	Setting 2	Setting 3	Setting 4
Option	Continue-mission-direct-route	49	48	49	49
	Continue-mission-detour	1	2	1	1
	Engage-enemy	0	0	0	0
	Return-to-base	0	0	0	0

Figure 1 shows an example of how somatic evaluation values change under influence of experience.

Figure 1. Change of somatic evaluation value over time with weight setting 2 in situation enemy-from-front



The somatic evaluation value for the option *continue-mission-direct-route* does not change, as this option is never selected. The somatic evaluation value for *continue-mission-detour* drops under the threshold of 0.25 after 4 selections, which means that this option will not be considered again and consequently not be selected at all. For the option *return-to-base* this happens after 14 selections. The somatic evaluation value for *engage-enemy fluctuates strongly* as there is a great variation between differing outcomes which lead to different resulting body state values.

This example shows that the agent has learned that in this situation *continue-mission-detour* and *return-to-base* are bad options and will only consider *engage-enemy* and *continue-mission-direct-route* in the future.

Verification

In order to verify whether the behavior of the model indeed complies to the Somatic Marker Hypothesis as proposed by Damasio, a logical verification tool has been used. Below, the formal language underlying this verification tool is explained, after which properties are shown that have been verified against a variety of traces.

The verification of properties has been performed using a language called TTL (for Temporal Trace Language), cf. (Bosse et al., 2009) that features a dedicated editor and an automated checker. This predicate logical temporal language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to *states* of the world, *time points* and *traces*, i.e. trajectories of states over time. In addition, *dynamic properties* are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the infix predicate |=, where state(γ, t) |= p denotes that state property p holds in trace γ at time t. Based on these statements, dynamic properties can be formulated in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as ¬, ∧, ∨, ⇒, ∀, ∃. For more details, see (Bosse et al., 2009).

The properties that have been verified against the simulation traces are shown below. The first property (P1) expresses that a negative evaluation of an option in a given situation with respect to a certain goal results in the somatic marker value for that option going down.

P1: Lowering specific somatic marker value

If an option O has been selected, and the evaluation of this option with respect to a goal G is bad, then the somatic marker value of this option for goal G will be lower than before.

∀γ:TRACE, t1:TIME, O:OPTION, S:SITUATION, G:GOAL, V1:REAL, E:REAL
[[state(γ, t1) |= belief(selected_option(O)) &
  state(γ, t1) |= belief(current_situation(S)) &
  state(γ, t1) |= somatic_marker(G, S, O, V1) &
  state(γ, t1) |= evaluation(G, O, E) & E < NEUTRAL]
 ⇒ ∃t2:TIME > t1, V2:REAL
   [state(γ, t2) |= somatic_marker(G, S, O, V2) & V2 < V1]]

In case the overall evaluation of an option in a given situation is below neutral, then the total somatic evaluation value goes down. This is expressed in property P2. The overall evaluation value is the weighted sum of the evaluation values for all goals. Note that the remaining formal forms have been omitted for the sake of brevity.

P2: Lowering overall evaluation value

If an option O has been selected, and the overall evaluation of this option is bad, then the value of the total somatic evaluation value for option O will go down.

The idea of Damasio is that certain options are no longer considered because they are not appropriate in a given situation. This idea is expressed in property P3 which states that once the total somatic evaluation value is below the threshold, the option will no longer be selected.

P3: Ignoring values below threshold
If the total somatic evaluation value for an option O is below the threshold, then this option is never selected.

Finally, property P4 expresses that eventually an option is selected which has a higher evaluation value than neutral.

P4: Eventually a good option is selected
There exists a time point such that an option O is selected which scores good for all goals.

The properties above have been verified against 12 simulation traces (3 situations, each consisting of 4 settings) During the verification process, a value of 0.5 has been used for the constant NEUTRAL. It was shown that property P1-P3 are satisfied for all traces. Property P4 however is not satisfied for the case whereby the enemy comes from the front, and the weight setting 3. The same holds for the case enemy from behind with setting 2. This is due to the fact that the probability of an option having a positive evaluation for these scenarios is very small, and does not occur in the trace which has been checked.

Conclusions

Damasio’s Somatic Marker Hypothesis (Damasio, 1994) shows how emotions play an essential role in decision making. It gives an account of how feeling (or experiencing) emotions in certain situations over time leads to the creation of a form of intuition (or experience) that can be exploited to obtain an efficient and effective decision making process for future situations met. Example of patients with brain damage related to feeling emotions show how inefficient and ineffective a decision making process can become without this somatic marking mechanism. Damasio’s theory contrasts with the earlier tradition in decision making models, where the focus was on rational decision making based on the Expected Utility Theory, and where the aim was to exclude effects of emotions and biases on decision making; e.g., (von Neumann and Morgenstern, 1944; Friedman and Savage, 1948; Arrow, 1971; Keeney and Raiffa, 1976).

To formalise Damasio’s Somatic Marker Hypothesis an approach was chosen based on the following assumptions.

- For a given type of emotion, somatic markers are related to combinations of contexts and decision options for this context.
- When a decision has to be made within a given context, somatic evaluation values associated to the options are used.
- Somatic markers and somatic evaluation values are expressed as real numbers between 0 and 1.
- Contexts and decision options are expressed as discrete instances.
- Within a given context, every decision option gets a somatic evaluation value associated based on the somatic markers.
- Decision options with low associated somatic evaluation value are eliminated from further decision processing.
- For the remaining decision options a (utility-based) rational analysis is made in which the somatic evaluation values serve as biases.
- Based on experiences for outcomes of chosen options for a given context, the somatic markers are adapted over time.

As for fighter pilots crucial decisions have to be made in very short times, it seems plausible that they heavily rely on such mechanisms. When applied to specific scenarios, the model shows patterns as can be expected according to Damasio’s theory. Creating the model is one of the first steps in larger research program. In next steps the model will be compared to decision making behavior of human pilots in a simulation-based training setting.

Acknowledgements

This research has been conducted as part of the “SmartBandits” PhD project which is funded by the National Aerospace Laboratory (NLR) in the Netherlands. Furthermore, the authors would like to thank Jan Joris Roessingh, and the anonymous reviewers for their useful comments that helped to improve the paper.

References

Arrow, K.J., (1971). *Essays in the Theory of Risk-Bearing.* Chicago: Markham, 1971.

Bittner J.L., Busetta, P., Evertsz, R., Ritter, F.E., (2008). *CoJACK – Achieving Principled Behaviour Variation in a Moderated Cognitive Architecture.* Proceedings of the 17th Conference on Behavior Representation in Modeling and Simulation 08-BRIMS-025 Orlando, FL: U. of Central Florida.

Bechara, A. & Damasio, A. (2004) *The Somatic Marker Hypothesis a neural theory of economic decision.* Games and Economic Behavior, vol. 52, pp. 336-372.

Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J., Specification and Verification of Dynamics in Agent Models. *International Journal of Cooperative Information Systems*. In press, 2009.

Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J., (2008). A Language and Environment for Analysis of Dynamics by Simulation. *International Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464.

Damasio, A., (1994). *Descartes' Error: Emotion, Reason and the Human Brain*, Papermac, London.

Damasio, A., (1999). *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*, Harcourt Brace, Orlando, Florida.

Dunn, B.D., Dalgleish, T., Lawrence A.D. (2005). The somatic marker hypothesis: A critical evaluation, Neuroscience and Biobehavioral Reviews 30 (2005) 239–271

Friedman, M., L. J. Savage (1948). The Utility Analysis of Choices Involving Risks, *Journal of Political Economy,* 56 (1948), 279-304.

Hoogendoorn, M., Merk, R.J., Roessingh, J.J.M., and Treur, J., Modelling a Fighter Pilot's Intuition in Decision Making on the Basis of Damasio's Somatic Marker Hypothesis. In: *Proceedings of the 17th Congress of the International Ergonomics Association, IEA'09*, 2009, to appear.

Jonker, C.M., and Treur, J., (1999). *Formal Analysis of Models for the Dynamics of Trust based on Experiences*. In: F.J. Garijo, M. Boman (eds.), Multi-Agent System Engineering, Proc. of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99. Lecture Notes in AI, vol. 1647, Springer Verlag, Berlin, 1999, pp. 221-232.

Kahneman, D., A. Tversky (1979). Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, Vol. 47, 1979, pp. 263-292.

Keeney, R. L., Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* New York: Wiley, 1976.

Tversky,A., Kahneman, D.(1974). Judgment under Uncertainty: Heuristics and Biases, *Science*, 185 (1974), 1124-1 131.

Modeling the Performance of Children on the Attentional Network Test

Fehmida Hussain (f.hussain@sussex.ac.uk)
Representation and Cognition Group, Department of Informatics,
University of Sussex, Falmer, Brighton, BN1 9QJ, UK

Sharon Wood (s.wood@sussex.ac.uk)
Representation and Cognition Group, Department of Informatics,
University of Sussex, Falmer, Brighton, BN1 9QJ, UK

Abstract

Recent research in attention indicates it involves three anatomical networks concerned with alerting, orienting and executive control (cf. Posner & Fan, 2007). The Attentional Network Test (ANT) provides a behavioral measure of the efficiencies of these three networks within a single task (Fan, MaCandliss, Sommer, Raz & Posner, 2002). This work adapts an ACT-R 6.0 model of adult performance on ANT (Hussain & Wood, 2009) to model the performance of children (aged 6, 7, 8, 9 and 10) on a child-friendly version of the task (Rueda, Fan, McCandliss, Halparin, Gruber, Lercari, Posner, 2004). Modifications are carried out within the framework of the ACT-R cognitive architecture (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004; Anderson & Lebiere, 1998). Models simulating the child study results indicate that improvements in latency and error rate can be attributed to incremental improvements in processing time and reduction in errors of commission respectively. In contrast the models indicate a qualitative difference between children under 9 and older age groups in both alerting efficiency attributed to specific reductions in processing surprise stimuli in the younger age groups, and executive control efficiency between 6 year olds and older age groups attributed to a slower ability in 6 year olds to focus the target in incongruent stimuli. An inhibiting effect of the alerting network on congruency, not found in the child study, was found in the model data consistent with adult studies (Callejas, Lupianez & Tudela, 2004; Fan, Xiaosi, Kevin, Xun, Fossella, Wang, Posner, 2009). Investigation of model performance under invalid spatial cueing conditions compared to adult model performance (Hussain & Wood, 2009) finds the models are differentiated by a slower ability to disengage from invalidly cued locations in the child models but are similar in benefiting from the facilitating effects of cueing on processing congruent stimuli.

Keywords: Attentional Networks; Attentional Network Test; ANT; ANT-C; Alerting; Orienting; Executive Control; Computational Modeling; ACT-R; Cognitive Development.

Introduction: Attentional Networks

Posner and Peterson (1990) propose that attention comprises a system of anatomical regions which can be divided into the networks of alerting, orienting and executive control. Alerting performs the function of achieving and maintaining a vigilant state; orienting refers to selective visual-spatial attention; and executive control involves monitoring and resolving conflict in the presence of conflicting information. Neuroscience studies have shown that different brain regions are associated with each network (Raz & Buhle, 2006). Orienting consists of three operations, namely disengagement, movement and engagement each associated with separate brain areas (Posner & Peterson, 1990).

Various behavioral tasks have been used to study the behavior of these networks, such as vigilance tasks, cueing tasks, Stroop task and so forth. Fan and colleagues (Fan, et al., 2002) designed the Attentional Network Test (ANT) that measures the efficiencies of all three networks in a single behavioral task. ANT is a 30 minute reaction-time based task combining cueing experiments (Posner, 1980) and flanker effects (Eriksen & Eriksen, 1974).

Attentional Network Test Adapted for Children

ANT-C is a child-friendly version of the combination of flanker and cueing paradigms used with adults modified to study the development of the networks in children (Rueda, et al, 2004). A series of experiments studied age groups ranging from 6 to 10 years in terms of the latency, accuracy and efficiencies of the networks. Figure 1 shows the design of ANT-C adapted to be more child-friendly by replacing the target stimuli with five colorful fish. There are four cue conditions: no-cue, center-cue, double-cue and spatial-cue and three congruency conditions: neutral, congruent and incongruent. Other than the replacement of the arrows with fish and the colorful display, the experimental setup remains the same.

Each trial begins with a central fixation cross followed by a cue (or a blank interval, in the no-cue condition) informing participants that a target will occur soon, and possibly where (spatial cue). The target always appears above or below the centre screen fixation point. An invalid cue (not part of the child study but explored in this paper to assess the effect of invalid cueing on disengaging) appears as a spatial cue but in the location opposite to where the target subsequently appears. The target array is either a fish on its own (neutral), or a central fish surrounded by flanking fish that point in either the same direction (congruent) or opposite direction (incongruent). Based on the direction of the centre fish, the children press the corresponding left or right button on the mouse. Reaction time (RT) spans stimulus presentation to button press.

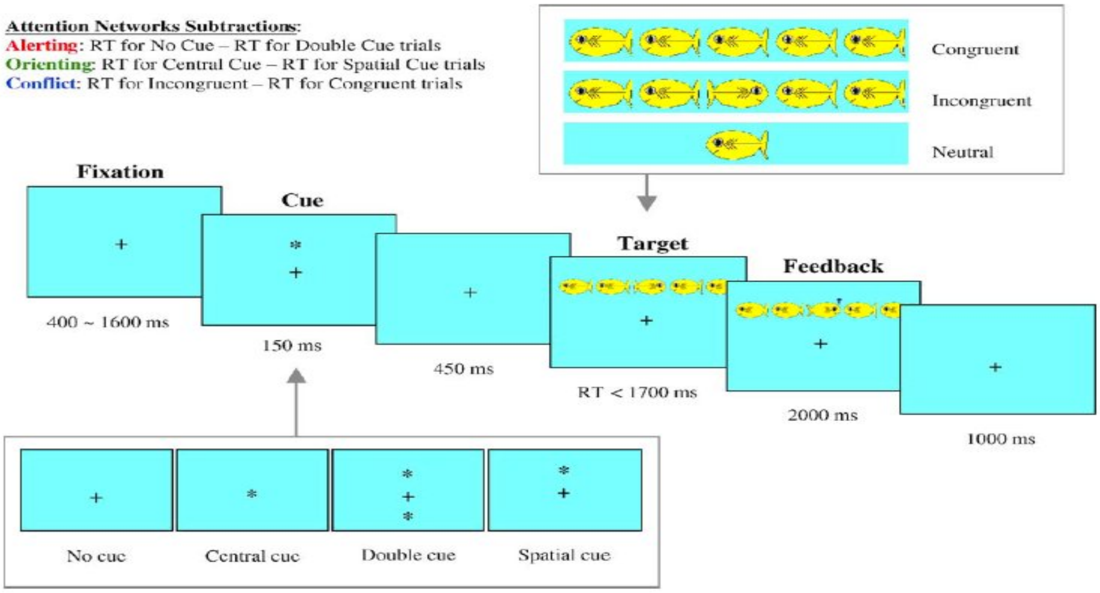

Attention Networks Subtractions:
Alerting: RT for No Cue – RT for Double Cue trials
Orienting: RT for Central Cue – RT for Spatial Cue trials
Conflict: RT for Incongruent – RT for Congruent trials
Congruent
Incongruent
Neutral
Fixation
Cue
Target
Feedback
400 ~ 1600 ms
150 ms
450 ms
RT < 1700 ms
2000 ms
1000 ms
No cue
Central cue
Double cue
Spatial cue

Figure 1: Child version of the Attentional Network Test (ANT-C), in which yellow fish on a blue background replace flanker arrows in the adult version of ANT (Rueda et al, 2004).

The duration of each trial is 25-30 minutes and children are given sufficient practice on the task before the data is formally collected. The formulae used to calculate the efficiencies remain the same as in the adult study, given in equations 1-3 (Fan et al, 2002). An invalid cue condition to study the effect of disengagement of attention is calculated as given in equation 4 (Callejas, et al, 2004; Fan et al, 2009).

Alerting = RT (no-cue) - RT (double-cue) (1)
Orienting = RT (center-cue) - RT (spatial-cue) (2)
Executive control = RT (incongruent) - RT (congruent)(3)
Validity = RT (invalid-cue) - RT (valid-cue) (4)

The child study (Rueda et al, 2004) reported that latency and accuracy improve over age, up to adulthood. The efficiency of the alerting network is much higher in children up to 9 years with no significant change across age. By age 10 and for adults alerting efficiency significantly reduces. The orienting network seems to be relatively stable up to 10 years with no change. Rate of development of executive control seems to reduce significantly from ages 6 to 7, but after that seems to stabilise up to adulthood with no significant change. Results are similar for 10 year olds and adults on both ANT and ANT-C. This paper compares the results from experiment 1 of the Rueda et al (2004) study that reports performance of age groups 6-9, and the partial results from experiment 2 for performance of 10 year olds on ANT-C, with model performance.

Simulating the Performance of Children on ANT-C Using ACT-R

A symbolic model of adult behavior on ANT (Wang and Fan, 2004) re-implemented in ACT-R 6.0 and extended to model invalid cueing and inter-network modulation effects (Hussain & Wood, 2009) is modified and adapted to simulate children's performance on ANT-C (Rueda et al, 2004). The ACT-R model display was not modified to show colorful fish instead of arrows as from the point of view of the functionality and behavior of the ACT-R model, it would not make a difference (ibid.) The important element to be captured here is the behavior in terms of the cuing and congruity information content of the display, and not color, shape and other visual aspects of the stimuli. The child models were also run on a variation of the task incorporating invalid cuing to assess validity efficiency (eq. 4) and the disengaging effect. Performance is compared with recent findings from adult human studies (Fan et al, 2009) and adult model performance (Hussain & Wood, 2009) based on the adult human studies of Fernadez-Duque & Black (2006) and inter-network modulation effects (Callejas et al, 2004).

Design and Functionality of the Model

The major functionality of the model remains the same as the Hussain & Wood (2009) model of ANT simulating healthy young adults. It consists of four blocks of code: (1) fixation and cue expectation, (2) cue processing, (3) stimulus processing and (4) responding to stimulus.

Associated with each functional step are a number of condition-action (if-then) production rules and parameter settings that combine to produce latency and accuracy data. Through a combination of certain rules firing based on the values in its buffers and underlying parameter settings, the model implements the effects of the alerting, orienting and control networks on attention performance, calculated by equations (1-4) and summarized below (for details refer to Hussain & Wood, 2009; 2009a).

Latency and Accuracy: The time between the appearance of a stimulus and the pressing of the key/mouse is the response time which accounts for latency in ms. Each processing step involved in performing the task involves a rule firing with a default timing of 40 ms. The model also reproduces errors seen in human studies. The number of errors made in each cue and flanker condition is recorded and the average percentage of incorrect responses is reported. The technique for modeling errorful performance corresponds to evidence that errors occur either due to confusion and distraction caused by incongruency, that is commission errors (Mezzacappa, 2004) or simply due to imperfect behavior, just randomly making a mistake.

Alerting: The efficiency of alerting is the difference in latency when there is no cue preceding the stimulus and when there is a double cue that prepares the subject but does not cue spatially. The element of surprise leads to the firing of an extra production, *notice-something-but-not-a-cue* [P1], to simulate the effect of alerting or preparing for the stimulus; this has a subsequent effect on the stimulus processing step by making it more costly (by 40 ms for the extra rule fired).

Orienting: The effect of orienting is achieved in two ways: (1) In the case of cueing, the model is made to focus on the target location using the buffer stuffing mechanism in ACT-R (URL 01) by varying the spread of visual attention determining which object is available for selective attention. For example, if the cue is spatial, then a narrower spread of attention will lead to a higher chance of focusing on the target and ignoring distracters as opposed to other cue conditions whereby both the target and distracters stand an equal chance of being selected for processing. (2) Also, when a spatial cue is encountered, the focus of attention is moved to that location in advance of the target appearing, so when the target stimulus is encountered attention is already engaged at the location, speeding up its selection as opposed to other cue conditions where attention had to be shifted to the target taking an extra processing step.

Executive Control: Executive control involves mental operations that are responsible for detecting and resolving conflicting situations. Here in the model, it is about simulating the flanker effect; showing that at times instead of the centre arrow (or fish) a flanker arrow located nearby may be selected due to distraction or even crowding of the scene (Pashler, 1998). The way the model handles this situation in the case where it encounters arrows in same direction (congruency condition), is by recognising the direction of the arrow and responding by pressing a key. There is no conflict or confusion and the model simply encodes the location and responds based on the direction of the arrow. The model responds through the rule *go-ahead-responding-if-congruent* [P2]. Incongruency is handled through competing productions whenever a flanker rather than the centre arrow is picked up (i) *harvest-direct-directly-if-incongruent* [P3] and (ii) *refocus-again-if-incongruent* [P4]. The first strategy using production P3 means that despite selecting a flanker instead of the target, the model encodes and responds to the direction of the centre arrow (taking a default 85 ms to move attention). In contrast, the second strategy, using production P4 requires the model to first shift attention to the centre arrow location and then recognize the direction of the centre arrow. Shifting attention involves firing an additional production (taking an extra 40 ms) at a total cost of 125ms making this strategy more costly. Choosing between competing rules is handled by the sub-symbolic component of ACT-R: [P3] and [P4] have utility values of 7 and 15 respectively corresponding to probabilities of 0.07 and 0.93. The probabilities are calculated on the basis of the default ACT-R equation (5). In this way, if there are a number of productions competing with expected utility value *Uj* then the probability of choosing production *i* is described below:

$$\text{Probability (i)} = \frac{e^{Ui\sqrt{2s}}}{\sum j\ e^{Uj\sqrt{2s}}} \tag{5}$$

Here the summation is over all productions that are currently able to fire, '*s*' is the expected gain noise.

Model Fitting and Justification

Generally there are two ways of modeling cognitive development: (1) either model adult behavior and then modify it to fit child behavior or (2) first model the child behavior (lower performance level) and progressively change to fit the adult behavior (higher performance level) (Jones, Ritter & Wood 2000). Using the former approach, the modeling work reported in this paper is implemented within the constraints of the ACT-R architecture. A cognitive architecture poses constraints on the implementation of a model and therefore influences design choices (ibid).

Researchers have shown that model behavior can be altered by making changes either to the knowledge retrieval capability of the model, the procedural rule based system or by making plausible changes to the sub-symbolic components (Jones & Ritter & Wood, 2000; Serna, Pigot, & Rialle, 2007; Rijn, Someren, Maas 2000). In this paper, the adult model was incrementally modified to simulate children's developmental trajectory. Theoretical interpretation of the human study findings suggested the basis for developmental differences in the various networks and their implementation, described further below. By modifying the adult model of ANT, five new models were created and run for 12 subjects each, to simulate the performance of each age group. In addition, an invalid cueing condition was introduced into the task and performance modeled to assess validity efficiency and the effect of disengaging from an incorrectly cued location. Various approaches with a sound theoretical basis were tried and the one giving the best statistical fit is presented here.

Latency: Response times improved progressively with age up to adulthood which was simulated by starting with an overall higher rule firing time for the model of 6 year olds then reducing this for each later age group to approach the adult rule firing time. Rule firing time is considered the basic information-processing step in ACT-R. Adjusting rule firing time seems a natural choice to obtain uniformly increased latencies across the whole model. Two variations using different set of values both yielded very good correlations with human data, but the model that also showed lower RMSD with the human data were 110, 90, 75, 55 and 45 ms for ages 6-10 respectively.

Accuracy: Errors can be induced in the system either through changing utility values of the error productions (Seran, Pigot, Rialle, 2007) or through inducing more noise in the system (Rehling, Lovett, Lebiere, Reder, Demiral, 2004; Ritter, Schoelles, Klein, Kase, 2007; Jones, Ritter & Wood, 2000). For inducing noise, the settings tried for the ACT-R gain noise parameter were in the range 3 to 6. Also, it is reported in the literature that children tend to make more errors due to distraction from flankers (Mezzacappa, 2004) and hence competing productions with varying utility values were used to model various likelihoods of giving either a correct answer, a random response without checking or purposely giving an incorrect answer. Both methods were applied with similar effects on correlations and RMSD implying that either noise or competing productions might contribute to erroneous behavior; both modifications are equally plausible, however, with good empirical evidence for the latter competing productions were used in the models to simulate errorful performance. The utility values for rules giving a correct, random or incorrect response are 20, 5 and 8 respectively in the adult model. For 6 year olds the random response value with the best fit is 8 and 6 for all other age groups. Incorrect response utilities decremented from 13 to 9 for ages 6-10 respectively. Correct responses held the adult value.

Alerting Network Efficiency: Alerting efficiency is higher up to age 9 reducing around age 10 and further still for adults. Although the overall longer rule firing time has the effect of increasing the latencies of all the networks, in order to fit the data the alerting network needs to be slowed down further in the younger age groups indicating there is poorer alerting efficiency at this age. This is modeled by increasing the rule firing time for the production P1 responsible for giving rise to the effect of surprise when a stimulus appears without an alerting signal. The specific firing time for P1 is set to 55 ms for age groups 6-9 compared to 40 ms in the 10 year olds and adult models.

Orienting Network Efficiency: The overall increase in rule activation time matched the orienting network score of the model with the human data; therefore no other change was required. Also the production that gives the effect of delay in the centre cue condition is not increased and takes the same time as the adult model (*notice-stimulus-with-centercue-and-shift* [P5]. This leads us to infer that not only is the orienting network well developed in the age groups modeled but also there is no effect on the capacity of shifting attention from the neutrally cued location.

Validity and Disengaging Effect: Researchers have suggested that it would be interesting to assess the effect of invalid cueing in children (Mezzacappa, 2004). Though this is not tested in the child study (Rueda et al, 2004 our adult model includes the invalid cueing extension to task (Hussain & Wood, 2009) and so by default do the child models; the invalid cueing condition was run for each age group and the effect of disengaging on validity efficiency calculated using equation 4.

Executive Control Network Efficiency: In Reuda et al's study, 6 year olds are uniquely poor compared to other age groups. This age difference was investigated by changing the utility values of the two conflicting productions [P3] and [P4] that handle incongruency to increase the likelihood of choosing the slower, less efficient P4 rule; however this did not achieve the desired result. An alternative approach is to set the rule [P4], which requires the model to refocus every time a flanker is encountered, with a longer firing time. For the model of 6 year olds only, the rule firing time for production P4 was increased to 60 ms reflecting a slightly slower capacity to refocus compared to all other productions.

Results and Evaluation

The latency data, accuracy data, efficiencies and the possible interactions of the networks are given in detail below. A series of models were run for 12 subjects each to simulate the ages 6, 7, 8, 9 and 10. Adult human for ANT-C (Rueda et al, 2004) and model data (Hussain & Wood, 2009) is also reported for baseline values (see figure 2). Results from running the same model for the invalid cueing condition are also reported.

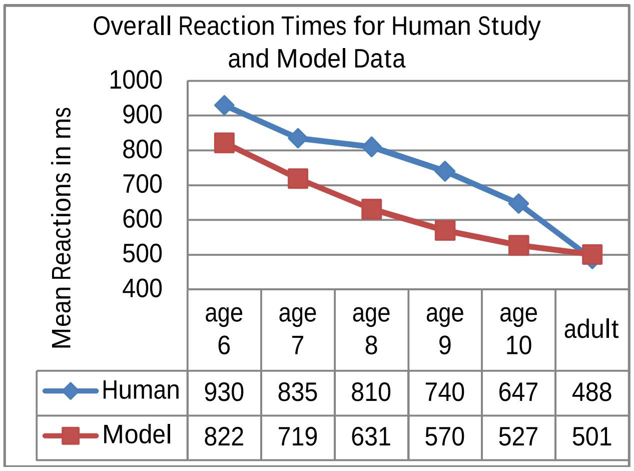
Latency Data As observed by the human study, the model response times incrementally improve for each age group. The statistics of correlation on the mean response times over all model runs shows good correlations and RMSDs, as reported in table 1. Figure 2 shows the mean reaction times (RT) for the human study in each age group along with the simulated results from the ACT-R models.

Accuracy Data As observed in the human study, the model error rate incrementally improved for each age group. However, when the results for each individual age group from the human study were observed closely it was found that for ages 7 and 8 the errors were higher in the neutral and congruent conditions as compared to the incongruent condition (Rueda, et al, 2004) which was not the case in the model data and therefore for ages 7 and 8 there were negative correlations with the human data. The models incrementally show improvement in accuracy and a higher chance of error in the case of the incongruent condition. The models could have been fitted to simulate this anomaly; however, it did not seem logical to do so. The model is in line with child development literature which shows that children make more errors in the case of incongruency (Ahktar & Enns, 1989; Mezzacappa, 2004). Further support for the model comes from a third experiment by Rueda et al (2004) involving 7 year olds. Table 1 reports child data from experiment 3 for age 7.

Age	Latency data		Accuracy data	
	r	RMSD	*r*	RMSD
6	0.79	34.7	0.93	1.28
7	0.92	34.4	0.86	1.02
8	0.88	52.5	-0.11	1.24
9	0.93	38.3	0.58	1.15
10	0.93	35	0.72	0.68

Table 1: Correlations and RMSD are used to show statistical fit of the model to the human data for age groups 6-10 years.

Efficiencies of Attentional Networks The efficiencies of the networks for each age group were calculated using equations 1-4. The efficiency data further validates the models by simulating similar values. As reported in the child study, alerting is much higher in the models for age groups 6-9; orienting scores do not show any significant difference across various age models; whereas executive control shows a high value for the model for age 6. The added finding using invalid cueing is that the validity effect is higher up to age group 10 with this increase mainly accounted for by a poorer ability to disengage from an uncued location. Correlations of the efficiencies of the networks of alerting, orienting and executive control of the model and human study for age groups 6-10 and adult data is 0.9, 0.8 and 0.9 respectively.



	age 6	age 7	age 8	age 9	age 10	adult
Human	930	835	810	740	647	488
Model	822	719	631	570	527	501

Figure 2: Mean RTs for all age groups for human data and simulation showing decreased mean reaction times.

Interaction of Attentional Networks Once the models were shown to be veridical simulations of child performance the interactions of the networks on each other were explored. Rueda and colleagues (2004) reported no interaction effects in their paper. However, studies exploring interactions of networks in adults (Callejas et al, 2004; Fan et al, 2009) show the alerting network has an inhibitory effect on congruency (in line with Posner's idea of "clearing of consciousness" (Posner, 1994, p7401)); in contrast orienting may have a facilitating effect (Callejas et al, 2004; Fan et al, 2009). So applying the formulae in equation 6 and 7, the effect of alerting on congruency was also explored for the child models. Similar equations measured the affect of cueing on congruency.

Effect of alert on cong = (alert-incong – alert-cong)(6)
Effect of un-alert on cong=(nocue-incong–nocue-cong)(7)

The simulation of children's performance produced an inhibitory effect of alerting on congruency although of variable magnitude. This suggests that although the networks of alerting and congruency have slower efficiencies in the child models the interactions are similar to those produced in adult human studies.

General Discussion and Conclusion

The work reported in this paper is based on a reimplementation of Wang & Fan's (2004) model of attentional networks (Hussain & Wood, 2009) to simulate child performance in a study by Rueda et al, (2004), measuring various age groups on a child-friendly version of ANT (ANT-C) and projecting the trajectory of development of various attentional networks. The sequence of models simulates the child study findings well. The model fitting process in the light of relevant child development literature helps explain some of the observed age differences: (1) the overall increased latencies are accounted for by slowing down the rule firing times of all productions, which means that children take more time to process in general and tend to make more mistakes; children make more commission errors, the ones due to confusion and distraction (2) alerting network efficiency is slower than that found in healthy adult studies simulated by slowing down the firing time of the rule which induces an element of "surprise", so the ability to get alerted in the absence of a signal is slower in children under 10; (3) both orienting network efficiency and the ability to shift from center cue and move to the target location are at adult levels; (4) however, by simulating child performance after introducing an invalid cueing condition, a higher validity effect was found, improving up to age 10. This high validity efficiency was accounted for mainly due to slow disengaging ability, a component of orienting; (5) poor conflict resolution ability in age group 6 is due to a non-optimal refocusing ability when a distractor is selected; and (7) from the model results we conclude there is an inhibiting effect of alerting and facilitating effect of cueing on congruency in children as in adults (Callejas, et al, 2004; Fan et al, 2009).

References

Ahktar, N. & Enns, J.T. (1989). Relations between covert orienting and filtering in the development of visual attention. *J. of Exp. Child Psychology*, 48, 315-344.

Anderson, J. R. & Lebiere, C. (1998). The atomic components of thought. Mahwah, NJ: Lawrence Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C. & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review* 111, (4). 1036-1060.

Callejas, A., & Lupianez, J. & Tudela, P, (2004). The three attentional networks: on their independence and interactions. *Brain Cognition*. 54, 225– 227.

Eriksen BA, Eriksen CW. (1974). Effects of noise letters upon the identification of a target letter in a non search task. *Perception and Psychophysics*, 16:143-149

Fan J., McCandliss B.D., Sommer T., Raz M. & Posner M.I. (2002).Testing the efficiency and independence of attentional networks. *J. of Cog. Neurosc.* 3(14):340–47.

Fan, J, Xiaosi, G, Kevin GG, Xun, L, Fossella, J, Wang, H, Posner, MI (2009). Testing the behavioral interaction and integration of attentional networks. *Brain and Cogn.*

Fernandez-Duque, D., Black, S.E., (2006). Attentional networks in normal aging and Alzheimer's disease. Neuropsychology. 20:2, 133-143.

Hussain, F. & Wood, S. (2009). Modeling the Efficiencies and Interactions of Attentional Networks, In L. Paletta & J.K. Tsotsos, Eds. *Attention in Cognitive Systems. LNAI 5395,* Springer-Verlag, Berlin, Germany.

Hussain, F. & Wood, S. (2009a). Computational Modeling of Deficits in Attentional Networks in mild Traumatic Brain Injury: An Application in Neuropsychology. Proceedings of 31th Annual Conference of the Cognitive Science Society.

Jones, G., Ritter, F. E. & Wood, D. J. (2000). Using a cognitive architecture to examine what develops. *Psychological Science*, 11(2), 1-8.

Mezzacappa, E., (2004). Alerting, orienting, and executive attention: Developmental properties and socio-demographic correlates in an epidemiological sample of young, urban children. *Child Development,* 75: 1-14.

Pashler, H. (1998). The Psychology of Attention. Cambridge, MA: MIT Press.

Posner M.I. & Fan J. (2007). Attention as an organ system. In Neurobiology of Perception and Communication: From Synapse to Society. De Lange Conference IV, Ed. J Pomerantz. London: Cambridge Univ. Press.

Posner M.I. (1980). Orienting of Attention. Quarterly *Journal of Experimental Psychcology*. 32, 3-25.

Posner, M.I. & Petersen, S.E. (1990). The Attention system of the Human Brain. *Ann. Rev of Neuroscience*, 13: 25-42.

Posner, M.I. (1994). Attention: The mechanisms of consciousness. *Proc. Nat. Acad. Sc, USA*, 91, 7398-7403.

Posner, M.I., Walker, J.A., Fredrich, F.A., & Rafal, R.D. (1984). Effects of parietal injury on covert orienting of attention. Journal of Neuroscience, 4(7), 1863-1874.

Raz, A. & Buhle, B., (2006). Typologies of attentional networks. *Nature Review Neuroscience* 7, 367-379.

Rehling, J., Lovett, M., Lebiere, C., Reder, L. M., & Demiral, B. (2004) Modeling complex tasks: An individual difference approach. In proceedings of the *26th Annual Conference of the Cognitive Science Society* (pp. 1137-1142) . August 4-7, Chicago, USA

Ritter, F. E., Schoelles, M., Klein, L. C., & Kase, S. E. (2007). Modeling the range of performance on the serial subtraction task. In Proceedings of the 8th International Conference on Cognitive Modeling. Lewis, R. L., Polk, T. A., Laird, J. L., (eds.). 299-304. Oxford, UK: Taylor & Francis/Psychology Press.

Rueda, M.R., Fan, J., McCandliss, B.D., Halparin, J.D., Gruber, D.B., Lercari, L.P. & Posner, M.I. (2004). Development of attentional networks in childhood. *Neuropsychologia, 42,* 1029-1040.

Serna, A., Pigot, H., & Rialle, V. (2007). Modeling the progression of Alzheimer's disease for cognitive assistance in smart homes. *User Modeling and User-Adapted Interaction, 17*, 415-438.

Van Rijn, H., van Someren, M., & van der Maas, H.L.J. (2003). Modeling developmental transitions on the balance scale task. *Cognitive Science*, 27(2), 227-257

Wang H., Fan J. & Johnson T. R. (2004). A symbolic model of human attentional networks. *Cog. Sys. Res.,* 5:119–34.

URL 01: [http://act-r.psy.cmu.edu/actr6/reference-manual.pdf]

A Formal Comparison of Model Variants for Performance Prediction

Tiffany S. Jastrzembski (tiffany.jastrzembski@us.af.mil)
Warfighter Readiness Research Division, Air Force Research Laboratory
6030 S. Kent St., Mesa, AZ 85212 USA

Kevin A. Gluck (kevin.gluck@us.af.mil)
Warfighter Readiness Research Division, Air Force Research Laboratory
6030 S. Kent St., Mesa, AZ 85212 USA

Abstract

In the field of cognitive science, the primary means of judging a model's viability is made on the basis of goodness-of-fit between model and human empirical data. Recent developments in model comparison reveal, however, that other criteria should be considered in evaluating the quality of a model. These criteria include model complexity, generalizability, predictive capability, and of course descriptive adequacy. The current investigation seeks to formally compare three variants of a mathematical model for performance prediction. The results raise the issue of how to go about selecting a model when formal comparison methods reveal equivalent values. A possibility briefly proposed at the end of the paper is that cognitive/neural plausibility is an appropriate tiebreaker among otherwise equivalent functional forms.

Keywords: Mathematical Model, Performance Prediction, Model Selection, Model Comparison, Cognitive Plausibility

Introduction

As common practice in the field of cognitive modeling, most modelers judge the explanatory power and descriptive adequacy of their models on the basis of goodness-of-fit measures comparing model predictions to human empirical data in each highly specialized task environment for which those models had been developed. It is far less typical to assess the generalizability or predictive power of a single model across multiple sets of data, tasks, or domains. It is also atypical for modelers to investigate substantive variations in the implementation of a single model, where multiple mechanisms could potentially achieve equivalent values in goodness-of-fit. Thus, the common practice of basing model performance on the goodness-of-fit criterion alone may lead a modeler to erroneously conclude that true underlying process regularities have been captured (Roberts & Pashler, 2000), which could in turn lead to faulty theoretical claims.

To minimize this probability and to effectively evolve cognitive theory, the modeling community must conduct more thorough investigations of model instantiations, whereby selection should be based on formal comparison criteria. The most widely used means of model comparison is quantitative in nature, and is referred to as goodness-of-fit, or descriptive adequacy. Assessment in this criterion includes optimizing model parameters to first find the best fit, and then choosing the model that accounts for the most variance in the data (typically calculated as root mean square deviation (RMSD) or sample correlation (R^2). This practice is a critical component of model selection, but simply selecting a model that achieves the best fit to a particular set of data is critically insufficient for determining which model truly captures underlying processes in the human system. In fact, basing model selection on this criterion alone will always result in the most complex model being chosen, whereby overfitting the data and generalizing poorly could be very real problems, and interpreting how implementation ties to underlying processes may be all but impossible (Myung, 2000).

The inclusion of additional *qualitative* model selection criteria (i.e., weighing the necessity of added parameters) helps overcome these pitfalls and improves our chances of selecting models that offer more insight into how human memory functions. Because complex models are more likely to have the ability to capture a particular set of data well, including the possibility of capturing noise, it is necessary to embody the principle of Occam's Razor (William of Occam, ca. 1290-1349) in model selection tools by balancing parsimony with goodness-of-fit. This translates into accounting for both the number of parameters included in a model, and the model's functional form, defined as the interplay between model factors and their effect on model fit.

Take for example the following models, which include the same number of parameters, but differ drastically in their functional form:

Model 1: $y = ax + \mathrm{b}$

Model 2: $y = ax^b$

Model 3: $y = \sin(\cos ax)^a e(-bx)/x^b$

In this scenario, Model 3 should incur a greater penalty than Models 1 or 2 because of its functional complexity. Further, in order to justify the addition of parameters or the additional complexity in functional form, it must be shown that the inclusion of added parameters is necessary to explain the data and add substance to the underlying theoretical rationale.

Additional helpful criteria for model selection are generalizability and predictive capability. These concepts refer to the ability for a model to make valid and accurate predictions outside the task or domain for which it was originally developed, thereby tapping into some meaningful account of true underlying processes (e.g., Cutting, 2000). These criteria have been shown to have an inverse relationship to model complexity, where more complex models tend to generalize to new data sets poorly because parameters were optimized to fit one set of data, resulting in an overfit to the data and absorption of random error (Myung, 2000). Thus, simpler, more parsimonious models often perform better in generalization and predictive capability evaluations.

In the current investigation, we examine and evaluate three variations of a mathematical account of a Performance Prediction Model (Jastrzembski, Gluck, & Gunzelmann, 2006). The model is an extension of the General Performance Equation (Anderson & Schunn, 2000), and accounts for learning stability by balancing true time passed with training opportunities amassed. Given that no one model comparison technique incorporates all of the quantitative and qualitative inclusion criteria previously mentioned, we compare our model instantiations using the (1) Bayesian Information Criterion, which is sensitive to the number of parameters but insensitive to functional form, (2) Minimum Description Length, which is sensitive to both the number of parameters and their functional form, and (3) Cross-Validation, which provides a good measure of a model's ability to generalize but has no sensitivity to the number of parameters or functional form. We have previously compared one instantiation of this mathematical model of the spacing effect with a computational model of the spacing effect (Pavlik & Anderson, 2005) using these comparison techniques, and found that the more parsimonious mathematical account should be selected on the basis of all of these evaluation techniques (Jastrzembski, 2008).

This current work extends previous research to investigate manipulations to the mathematical model itself, to evaluate the necessity of parameters with different functional forms as they relate to goodness-of-fit measures, model complexity, and predictive power. We elucidate the issue of which model to choose when goodness-of-fit, model complexity, generalizability, and predictive capability of competing models are equivalent, and additionally bring to bear the issue of cognitive and neurological plausibility – a more abstract, currently unquantifiable construct in the model selection literature, but no less important than any of the criteria used in formal model comparisons. In sum, this work discusses the quantitative and qualitative differences across model instantiations, and argues that such thorough examinations are useful for evolving cognitive theory.

Performance Prediction Model

The model builds upon the strengths of the General Performance Equation (Anderson & Schunn, 2000), which handles effects of recency and frequency very well. However, we sought to extend the equation to capture effects of spacing, while also providing flexibility and the additional capability for predicting performance at later extrapolated points in time. This equation is expressed as:

$$Performance = S \cdot St \cdot N^{c} \cdot T^{-d};$$

(Equation 1a)

where free parameters include *S*, a scalar to accommodate any variable of interest, *c*, the learning rate, and *d*, the decay rate. Fixed parameters include *T*, defined as the true time passed since training began, and *N*, defined as the discrete number of training events that have occurred over the training period. The term *St*, defined in Equation 1b below, is short for Stability Term and is responsible for capturing effects of spacing by calculating experience amassed as a function of temporal training distribution and true time passed.

$$St = \left\lfloor \frac{\sum lag}{P} \cdot \frac{P_i}{T_i} \cdot \frac{\sum_i^j \left(lag_{max\ i,j} - lag_{min\ i,j} \right)}{N_i} \right\rfloor;$$

(Equation 1b)

where *lag* is defined as the amount of true time passed between training events and *P* is defined as the true amount of time amassed in practice. In the equation's current form, experience and training distribution attenuate performance by affecting knowledge and skill stability at the macro-level of analysis.

In the upcoming model comparison it is the *St* term that will be moved to different places in the equations to change their functional forms, and perhaps their theoretical implications. Before we move to the comparison, however, it is first necessary to illustrate the model's viability as it appears in Equation 1a.

Descriptive Adequacy across Test Harness of Data

We have validated the descriptive adequacy and predictive validity of this mathematical model across multiple types of previously published datasets from the cognitive/experimental psychology literature. This includes studies of knowledge acquisition, knowledge retention, skill acquisition, and skill retention. We also have validated the Performance Prediction Model with more recent applied data coming out of a team coordination Unmanned Air Systems (UAS) Predator reconnaissance task from the Cognitive Engineering Research Institute, and finally, with F-16 simulator air-to-air combat data coming from the highly complex Distributed Missions Operations testbed at the Air Force Research Laboratory's Mesa Research Site. Figures 1-4 provide a subset of our test harness data sets with model goodness-of-fit measures.

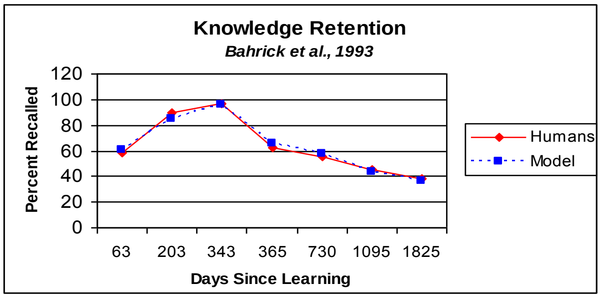


Figure 1. Task deals with the study of foreign language vocabulary and long-term retention. The model achieved an *RMSD* of 1.2% and R^2 = 0.98.

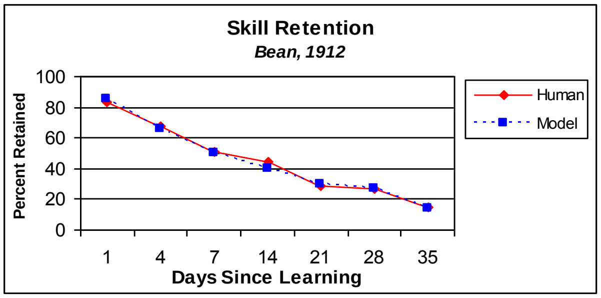


Figure 2. Task deals with retention of typing skills over periods of non-practice. The model achieved an *RMSD* of 1.34% and R^2 = 0.99.

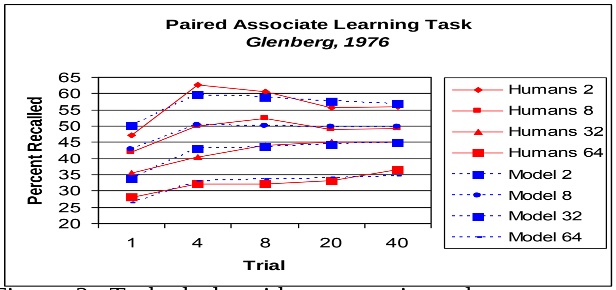


Figure 3. Task deals with monotonic and nonmonotonic effects across four retention intervals (2, 8, 32, or 64 days), and five levels of spacing (repetition every1, 4, 5, 20, or 40 trials). The model achieved an *RMSD* of 1.55% and R^2 = 0.96.

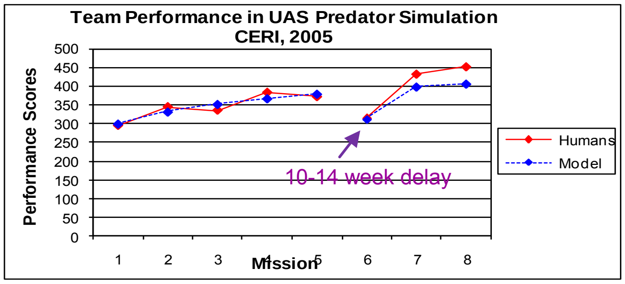


Figure 4. Task deals with a team of three individuals coordinating to complete five missions on the first day of training, then return 10-14 weeks later to perform an additional three missions, with the goal of flying a UAS and attaining pictures of targets. The model achieved an *RMSD* of 12.7 and R^2 = 0.94.

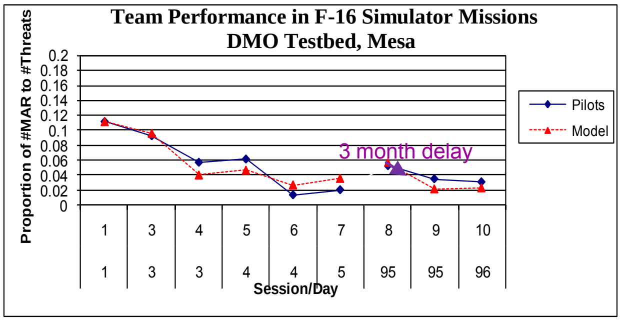


Figure 5. Task deals with a team of four pilots flying F-16 simulators who fly missions for a week of baseline training and return three months later for an additional two days of training. Objective measurements of the number of times they violated enemy airspace were taken. The model achieved an *RMSD* of 0.004 and R^2 = 0.96.

In sum, the current instantiation of the mathematical model achieved excellent goodness-of-fit across tasks. Given the placement of the stability term in this model's functional form, experience and training distribution may arguably attenuate learning and decay at the macro-level of performance analysis. We will next turn our attention to the relative descriptive adequacy of competing model instantiations, by shifting the stability term to other, theoretically-motivated locations.

Goodness-of-Fit Comparisons Across Model Variations

Pavlik and Anderson (2005) developed a computational model of the spacing effect in the ACT-R architecture, wherein they argued for an activation-based decay mechanism to variably adjust decay rates as a function of the activation value at the time of the presentation. This limits long-term benefits from further practice at higher levels of activation, and produces effects of spacing in tasks that are declarative memory dependent.

The second instantiation of the Performance Prediction Model is inspired by Pavlik and Anderson's model, and inserts the stability term directly into the decay parameter to approximate the activation-based decay mechanism (see Equation 2).

$$Performance = S \cdot N^{c} \cdot T^{-d \cdot St};$$

(Equation 2)

The third instantiation of the Performance Prediction Model receives its inspiration from the neurobiological literature, in which the timing and frequency of learning input determine whether long-term potentiation (LTP) or long-term depression (LTD) of neurons will occur (Dudek & Bear, 1992), which translates into stable or unstable knowledge, respectively. To approximate this theoretical perspective in our model, we distribute the stability term into both the learning and decay rate, as shown in Equation 3.

$$Performance = S \cdot N^{c \cdot St} \cdot T^{-d \cdot St};$$

(Equation 3)

Interestingly, goodness-of-fit measures across all three models and data are equivalent across the empirical datasets shown in Figures 1-3 (average R^2 for Equation 1a = 0.977, Equation 2 = 0.971, Equation 3 = 0.975). Differences arose, however, when examining the cases of the UAS Predator task and the F-16 DMO mission simulation. In those contexts, model descriptive adequacy was considerably worse for Equation 2 (activation-based decay instantiation), revealing a loss in explanatory power of 12% (see Figures 6 and 7). The nature of the discrepancy is that the model produces more forgetting during the lag periods than was observed in the human subjects and the model produces a greater degree of subsequent re-learning than was observed in the human subjects. Goodness-of-fit measures in these contexts were statistically equivalent for Equations 1 and 3 however (R^2 *for Equation 1a = 0.928, and Equation 3 = 0.925*).

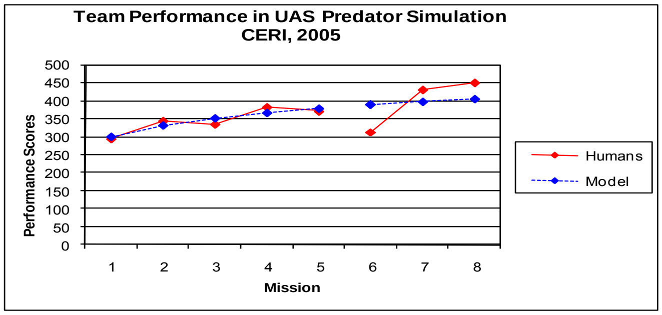


Figure 6. Activation-based decay model instantiation fit to UAS Predator Simulation task. The model achieved an *RMSD* of 30.6 and R^2 = 0.75.

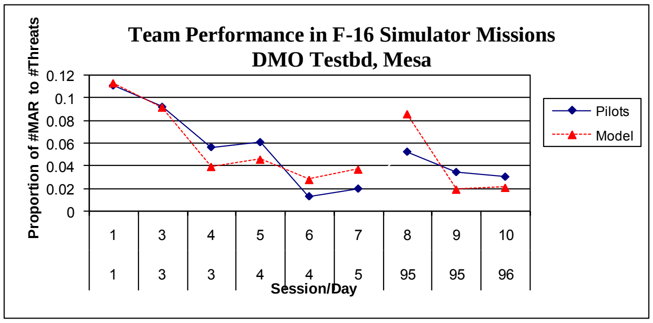


Figure 7. Activation-based decay model instantiation fit to F-16 team training in the DMO testbed. The model achieved an *RMSD* of 0.018 and R^2 = 0.91.

This exercise reveals a very interesting finding. Had the model instantiations only been compared across the first three sets of data, all model instantiations would have been deemed equivalent as far as descriptive adequacy goes. Only when the models were fit to the more applied data, entailing longer periods of delay, were weaknesses in Equation 2 revealed. In the next section, we will take our model comparisons to the next level, and compare them using three formal methods commonly used in the mathematical psychology community. Given the unacceptable level of descriptive adequacy in applied and relevant domains for Equation 2, we will omit this model from evaluation with the following comparison techniques.

Additional Qualitative Comparisons Across Model Variations

Bayesian Information Criterion (BIC) The goal of this comparison technique is to estimate a model's ability to predict all future data samples from the same underlying process by penalizing added parameters weighed against goodness-of-fit across all datasets of interest. The algorithm for evaluation with this criterion is provided in Equation 4:

$$BIC = -2\ln f\left(y\middle|\hat{\theta}\right) + k\ln(n); \qquad \text{(Equation 4)}$$

where the first term of the equation refers to the maximum likelihood function of the model given its optimized parameters, and the latter term of the equation refers to the number of free parameters included in the model (see Table 1 for breakdown of model parameters). The model that results in the lower BIC value is deemed the more parsimonious model to be selected.

Parameter	Symbol	Free Parameter?
Scalar	*S*	Yes
Stability Term Composition	*lag* *P*	No
Practice Amassed	*N*	No
Learning Rate	*c*	Yes
Time	*T*	No
Decay Rate	*d*	Yes
Total	**7**	**3**

Table 1. Breakdown of parameter information fed into formal comparison techniques.

With this comparison technique, both Equation 1a and Equation 3 reveal statistically equivalent values ($BIC_{Equation1}$ = 26.72, $BIC_{Equation3}$ = 26.15), due to statistically equivalent goodness-of-fit values and an equal composition of free parameters. Therefore BIC adds nothing to our ability to make an informed decision concerning model selection in this particular case.

Cross-Validation (CV) The motivation behind this technique is to select a model on its ability to capture behavior of unseen or future observations from the same underlying process (Browne, 2000). The method for evaluating the predictive accuracy of the model is to divide the available data into two subsets. The first subset is used for parameter calibration and the second subset of data is used for predictive evaluation. To conduct this analysis, half of the data points in each data set of our test harness were eliminated, and the models were calibrated with the remaining points. The algorithm for evaluation with this criterion is given in Equation 5, and the summary of the CV comparison is shown in Table 2:

$$CV = -\ln f\left(y_{validation}\ \middle|\hat{\theta}(y_{cali\,bration}\right); \qquad \text{(Equation 5)}$$

Experiment	Number of Data Points (Calibration/Validation)	Equation 1		Equation 3	
		RMSD	R^2	RMSD	R^2
Bahrick (1993)	4/3	2.83	0.92	2.53	0.93
Bean (1917)	4/3	3.16	0.94	3.09	0.94
Glenberg (1976)	10/10	4.05	0.89	3.98	0.90
CERI (2005)	8/8	18.7	0.91	17.46	0.92
DMO Testbed	5/4	0.011	0.92	0.011	0.93
Totals/Averages	**31/28**	**5.75**	**0.916**	**5.414**	**0.924**

Table 2. Cross-validation *RMSD* and R^2 values across model variants, data sets, and summary measures.

As revealed in Table 2, both Equation 1a and Equation 3 generalized quite well, predicting the unseen or future data to a high degree of precision and achieving statistically equivalent correlations to human data of 0.916 and 0.924, respectively. Based on this criterion, the decision to select one model over the other is again unresolved. We now turn to the final formal model comparison technique to evaluate our competing models.

Minimum Description Length (MDL) This measure of complexity evaluates a given model on the basis of the encoding length necessary to fit or predict observed data (Grünwald, 2000), and identifies the model that provides reasonable fits to data most parsimoniously. The algorithm for calculating this criterion is shown in Equation 6:

$$MDL = -\ln f\left(y\middle|\hat{\theta}\right) + \frac{k}{2}\ln\frac{n}{2\pi} + \ln\int d\theta\sqrt{det\lfloor I(\theta)\rfloor};$$

(Equation 6)

where both number of free parameters and the model's functional form are penalized.

Using this evaluation technique, Equation 1a results in a value of 8.07 and Equation 3 results in a value of 9.52. This is because Equation 3 distributes the stability term through both the learning and decay rate, whereas Equation 1a only incorporates the stability term in one location. Though Equation 3 resulted in a slightly worse value due to the added length of the equation, there were no added free parameters penalizing the model, so the MDL equation results in only a slightly higher score than Equation 1a. Thus, once again, the question of which model is the best selection remains unresolved.

Discussion

We investigated model viability on the basis of goodness-of-fit, model complexity, generalizability, and predictive capability. We argue that all of these criteria are essential in helping guide the decision-making process for selecting among competing models and objectively determining which model most succinctly captures true underlying cognitive processes.

We also argued that comparing different instantiations of a single model against itself can elucidate whether proposed mechanisms are necessary or viable. In this exercise, we shifted one parameter (the stability term) to theoretically-motivated locations in our mathematical model, and discussed the potential ramifications on cognitive plausibility that could be made as a function of that single change.

We found that one model variation (activation-based decay instantiation) was deemed to be descriptively inadequate when tested in applied domains over long lag periods, and we additionally found that the remaining two model variations, though different in functional form, were equivalent using criteria of descriptive adequacy, predictive power, and generalizability across tasks and domains.

The issues that are raised by these findings include how to select a model when formal comparison methods reveal equivalent values, and additionally, how to bring the unquantifiable construct of cognitive plausibility into the decision-making process when all else is equal.

The ultimate goal of a cognitive modeler is to push the science and advance cognitive theory, but if two models are objectively equivalent, provide theoretically plausible explanations of underlying processes, and provide good approximations of human learning, then where should a modeler turn?

This is precisely our conundrum with Equations 1 and 3. We believe strong theoretical claims can be made for each model variation, so our future work will include identifying one or more critical experiments, perhaps incorporating longer lags between training events or even multiple blocks of training across repeated, extended lags, to systematically discern whether one equation will prove to win out and provide greater descriptive adequacy for explaining a broaer range of empirical data.

Finally, we mentioned earlier that a motivation for the implementation of Equation 3 is the neurobiological literature on long-term potentiation and long-term depression at the neural level. As cognitive science continues its inexorable march toward clearer elucidation of the mind/brain relationship, it may very well be that cognitive/neural plausibility will prove to be an appropriate tiebreaker among otherwise equivalent functional forms.

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was sponsored by the Air Force Research Laboratory's Warfighter Readiness Research Division. The authors would like to thank the Cognitive Engineering Research Institute (CERI) and researchers from Mesa's Distributed Missions Operations testbed, particularly Dr. Nancy J. Cooke and Dr. Wink Bennett, respectively, for providing data for model evaluation.

References

Anderson, J. R. & Schunn, C. D. (2000). Implications of the ACT-R learning theory: No magic bullets. In R. Glaser (Ed.), *Advances in instructional psychology: Educational design and cognitive science, Vol. 5*. Mahwah, NJ: Erlbaum.

Bahrick, H. P., Bahrick, L. E., Bahrick, A. S., & Bahrick, P. O. (1993). Maintenance of foreign language vocabulary and the spacing effect. *Psychological Science, 4*, 316-321.

Bean, C. H. (1912). The curve of forgetting. *Archives of Psychology, 2*, 1-47.

Browne, M. W. (2000). Cross-validation methods. *Journal of Mathematical Psychology, 44*, 108-132.

Cutting, J. E. (2000). Accuracy, scope, and flexibility of models. *Journal of Mathematical Psychology, 44*, 3-19.

Dudek, S. M., & Bear, M. F. (1992). Homosynaptic long-term depression in area CA1 of hippocampus and effects of N-methyl-D-aspartate receptor blockade. *Proceedings of the National Academies of Sciences, 89*, 4363-4367.

Glenberg, A. M. (1976). Monotonic and nonmonotonic lag effects in paired-associate and recognition memory paradigms. *Journal of Verbal Learning & Verbal Behavior, 15*, 1-16.

Grünwald, P. (2000). Model selection based on minimum description length. *Journal of Mathematical Psychology, 44*, 133-152.

Jastrzembski, T. S. (2008). Cognitive model comparisons of the spacing effect: Criteria that account for complexity, generalization, and prediction capabilities. *Society for Mathematical Psychology annual meeting*, Washington, D. C.

Jastrzembski, T. S., Gluck, K. A., & Gunzelmann, G. (2006). Knowledge tracing and prediction of future trainee performance. *I/ITSEC annual meetings*, Orlando, December 4-7.

Myung, J. (2000). The importance of complexity in model selection. *Journal of Mathematical Psychology, 44*, 190-204.

Pavlik, P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science, 29*, 559-586.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review, 107*, 358-367.

What's in an error? A detailed look at SRNs processing relative clauses.

Lars Konieczny[1] (lars@cognition.uni-freiburg.de)
Nicolas Ruh[2] (nruh@brookes.ac.uk)
Daniel Müller[1] (daniel@cognition.uni-freiburg.de)

[1]Center for Cognitive Science, Friedrichstr. 50
79098 Freiburg i. Br., Germany

[2]Oxford Brookes University

Abstract

This paper responds to MacDonald and Christiansen's (2002) experience-based account of subject vs. object relative clause processing based on Simple Recurrent Network simulations. They found that object-extracted relative clauses exhibit performance penalties that are absent in subject relative clauses, and more so in less trained networks. Whereas MC argue that their finding reflects a differential amount of word order regularity in subject- vs. object-extractions, a detailed analysis of the word-by-word output-activation pattern suggests that it is caused by the network failing to distinguish verbs from the relative pronoun *that* during early training epochs. This interpretation is supported by other aspects of the activation pattern that indicate incomplete grammar acquisition. Nevertheless, the results point at a viable source of complexity in sentence processing.

Introduction

Relative Clauses and working memory

The contrast between Subject-extracted (2) and Object-extracted relative clauses (1) is the poster child of working-memory oriented psycholinguistics.

(1) The reporter *who the senator attacked* admitted the error (ORC)

(2) The reporter *who attacked the senator* admitted the error (SRC)

Subject-relative clauses (SRCs, 1) are generally easier to process than object-RCs (2), and more notably so for readers with a low reading span (King & Just, 1991). Among the multitude of models, two fundamentally opposing frameworks have been most prominent: retrieval-based working memory models, (eg. Just & Carpenter, 1992, Gibson, 1998, Gordon et al., 2004, Vasishth & Lewis, 2005), and experience-based models, such as probabilistic parsers (Hale 2001, Levy, 2005) and connectionist models, most notably that of MacDonald and Christiansen (2002). Their model is based on Simple Recurrent Networks (Elman, 1990). SRNs acquire implicit grammatical knowledge when they are trained on linguistic corpora. Crucially, they lack a clear distinction between linguistic knowledge, processing, and a knowledge-free notion of a working memory and its capacity. In MCs' SRN-based approach, the complexity difference between Subject and Object-RCs can be attributed to the differential degree of word-order regularity exhibited by SRCs and ORCs. Subject-RCs match the predominant subject-verb-object (SVO) word order of simple main clauses. Object-RCs, on the other hand, show an irregular O-S-V order. Processing SRCs hence benefits from "regular" word order expectations being transferred from main clauses, whereas no such transfer is made for ORCs. Therefore, SRCs are easier to process than ORCs despite the relatively low frequency of relative clauses in general. These predictions were – in principle at least – confirmed by MC's simulations with Simple Recurrent Networks. These networks were trained in three epochs of 10000 random sentences each. Because SRCs were easy even in the earliest training epoch, only ORCs benefited from more training. The resulting grammatical error pattern shows striking resemblance to the reading times of the different span groups of King and Just (1991). MacDonald and Christiansen (2002) hence attribute the differential performance of span-groups to their respective amount of linguistic experience rather than differences in working memory capacity. Basically, they reveal a – this time word-order-based – frequency (amount of training) x regularity (i.e. transfer from predominant order) interaction comparable to what has been demonstrated for other connectionist models in a variety of domains (e.g. Seidenberg & McClelland, 1989).

In this paper, we will show that MC's critical results can be attributed to a fundamental part of speech classification error due to insufficient learning in early epochs. We will argue however that the underlying mechanism of interference by locally coherent predictions might very well be a valid predictor for processing complexity.

SRNs and sentence processing

SRNs have successfully been demonstrated to be capable of implicitly acquiring limited recursive "grammars" (e.g. Elman, 1991; Christiansen & Chater, 1999). They do so by learning to predict the next word when presented with sentences word-by-word at the input. In the SRN architecture, there is a hidden layer that receives combined activation from the input layer and the context layer, which holds the content of the hidden-layer at the previous cycle. Using the standard back-propagation algorithm, the prediction-error, reflecting the deviance of the predicted activation pattern from the actual next word pattern, is used to adjust connection weights throughout the network back to the input layer. Eventually, after thousands of learning cycles, the SRN performs reasonably well even on sentences that it has never seen before. At this point, SRNs can be demonstrated to have classified words into their syntactic categories and possibly even into more fine-grained semantic distinctions (Elman, 1990). SRNs have repeatedly been demonstrated to be able to acquire an implicit recursive grammar (Elman, 1991, Christiansen & Chater, 1999).

As a measure of the grammatical viability of the network's predictions, output vectors are compared to grammaticality vectors calculated from the underlying context free grammar used to generate the training set. Each unit corresponds to a lexicon entry (word) and carries its grammatical probability in the context of the previous words in the sentence. For instance, if there are two grammatical continuations, both equally likely, the corresponding units both have a probability of 0.5 and should hence receive 50% of the output activation each.

Deviation from this activity pattern increases the *grammatical prediction error* (GPE). The GPE is a global error measure (i.e. the specific errors on each output unit are collapsed into a single value) ranging from zero to one, with zero meaning a perfect prediction of all grammatical continuations, and one meaning that all activation is on ungrammatical units. To achieve this, the GPE is computed from *hits* (summed activation on correctly predicted, grammatical nodes), *false alarms* (summed activation on incorrectly predicted, ungrammatical nodes) plus *misses* (sum of differences of desired and actual activity on grammatical nodes, if positive, weighted by the amount of total output activation), as specified in (3).

$$(3) \quad GPE = 1 - \frac{hits}{hits + falsealarms + misses}$$

A GPE decreasing over several training epochs reflects the network's ongoing acquisition of implicit grammatical knowledge.

MC used the GPE to predict on-line processing load, with GPEs being directly proportional to reading times. Unfortunately, they restricted their analyses to global error (GPE) patterns. However, the GPE as a global measure can reflect two independent properties of the networks: *i.* how well the networks have learnt the grammar underlying the training corpora, and *ii.* on-line processing difficulty. MC clearly focused on the second aspect, implicitly presuming that grammar acquisition even after the earliest training epoch has reached a mature enough stage to be compared to adult participants in reading studies. However, until more fine-grained analyses have been carried out, the source of the errors remains obscure.

What's in an error?

False alarm activation can indicate a. the lack of adequate knowledge about word categories and the constructions they can appear in, or b. the interference induced by locally coherent continuations, ignoring the global context they appear in. We will show that strong but globally inconsistent local dependencies can distract from globally grammatical predictions, even in networks that have sufficiently learnt to classify words along syntactic categories.

We present detailed analyses of a. the output activation patterns in our replication of MacDonald and Christiansen's SRNs, and b. multi-dimensional scaling results of average hidden layer activations[1].

SRN simulation

The SRNs were built from thirty-one units each in the input and the output layer, and sixty units each in the hidden and the context layer. Like MC (2002), we trained ten SRNs with ten different corpora. The corpora were generated from a 30 word vocabulary plus the *end of sentence* marker (EOS) fed into a probabilistic context free grammar. Ten percent of the NPs were modified by relative clauses[2], regardless of their position in the sentence. Half of the RCs were SRCs (25% transitive and 25% intransitive) and the other half ORCs (transitive only). RCs were both center-embedded or right branching. One half of the verbs were in the present tense, the other half in the past tense. The present tensed verbs agreed in number (singular or plural) with their clausal subject, past tensed verbs fit with both singular and plural subjects.

[1] We did not have access to MCs networks and data except for the summarized output activities. We therefore had to replicate their results before we could start analyzing hidden layer activities.

[2] The probabilities differ slightly from those published in the article, because we rather used the numbers of the actual original grammar generator that M. Christiansen has provided to us. Our test revealed the same basic activation patterns with either set of values.

Each training corpus contained 10,000 sentences, resulting in an epoch of about 55,000 sweeps (words) on average. The learning rate was set to .1, and there was no momentum. Cross-entropy was used to calculate the error used by the backpropagation learning algorithm. The test sentences were not included in the training corpus.

Results

There are two positions of interest with high GPEs: the embedded verb in ORCs and the matrix verb in both ORCs and SRCs. The most interesting spot in ORCs is the embedded verb, where the largest portion of experience-based variance was obtained in MC's networks, motivating the *frequency x regularity* interpretation.

Embedded verb in ORCs

In ORCs, the embedded verb follows a "NP–that–NP" sequence. After the first training epoch, the element most active here, quite surprisingly, is the *end of sentence* (EOS, see figure 1). This prediction is clearly ungrammatical, because neither the matrix clause nor the RC received a verb yet. In the second epoch, the prediction of an EOS has been strongly reduced, while the correct predictions of verbs with the right number marking were increased. This trend continues until the third epoch, where there is virtually no activity left for EOS. As for the verbs, it should be easy to establish the agreement between the NP and the verb, since both are adjacent in ORCs, as they are in main clauses. Surprisingly, it takes three epochs to learn this dependency to an adequate extent.

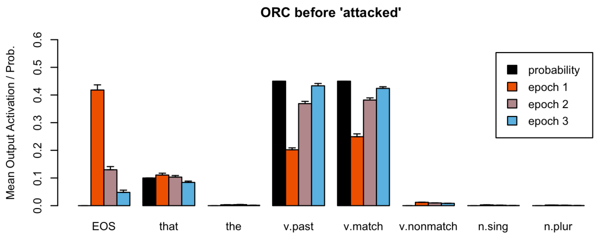


Figure 1: Mean output activations and grammatical probabilities at the embedded verb in ORCs, for three training epochs. Whiskers indicate standard errors.

Matrix verb

The second position at which a *sentence-type* x *experience* interaction was established in MCs simulations is the matrix verb. Moreover, GPEs on the matrix verb were high for both SRCs and ORCs. The results seem to fit King and Just's (1991) reading data in as much as reading times were also highest at this point in both SRCs and ORCs, with a slight advantage for SRCs. Nevertheless, while reading times at the matrix verb after ORCs showed the highest variability for readers of different span groups, the GPEs for ORCs in the network simulations varied not nearly as much at the matrix verb as on the embedded verb[3].

We examined the activation patterns at the matrix verb after both SRCs and ORCs, since both exhibit extremely high GPEs (between about .55 and .88).

SRCs. The detailed output vector analysis revealed that the GPE is based on one major false alarm component. In SRCs (figure 2), after a *verb-NP* sequence, the high GPE was based on false activation of the EOS, which did not change substantially over epochs.

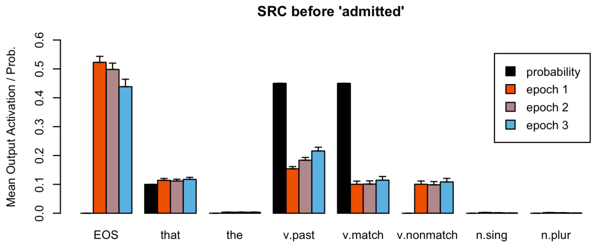


Figure 2: Mean output activations and grammatical probabilities at the matrix verb after SRCs, for three training epochs.

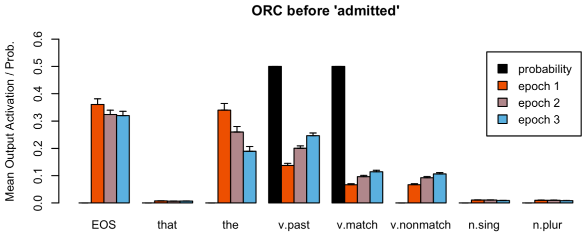


Figure 3: Mean output activations and grammatical probabilities at the matrix verb after ORCs, for three training epochs.

ORCs. After ORCs, following a *NP-verb* sequence, the only grammatical continuation is the matrix verb. Activation on all other words is a false alarm. Note that in the first epoch, the sum of false alarms is about 80%. The activation pattern reveals that the high GPE was due to one of the following two major false alarm components:

1. The false prediction of a determiner, indicating the prediction of another NP following the verb. This error dramatically decreased over the three epochs, but was still present even in the third epoch.
2. The false activation of EOS, which even grew slightly in the third epochs.

[3] However, the reading data on the matrix verb can be explained by a spill-over from the embedded verb, something that can quite regularly be observed in reading data. This dissimilarity between reading and simulation data should therefore not be taken too seriously.

Discussion

Embedded verbs

The activation patterns reveal that the high GPE at the embedded verb in ORCs during the first and the second epochs is mainly due to an ungrammatical prediction of an EOS. The remaining activation of the verbs shows that the networks have, at the same time, learned intra-RC number agreement, if not perfectly. How can this pattern of results be explained?

The EOS prediction is also high after SRCs following the sequence *NP-that-verb-NP*. Note that about half of the sentences end after the RC, namely when the RC modifies the Object-NP in transitive main clauses. The high false EOS prediction might thus be due to locally predicting the sentence ending, despite the context of a Subject-NP modifying center-embedded sentence.

Back to the embedded verb in ORCs. Here, *NP-that-NP*, and *...that-NP* is certainly not a good sentence ending. Two simple hypotheses can be ruled out fairly quickly. First, since about half of the sentences end with an NP, it might be just the NP that makes a good EOS in early training. Secondly, the prediction of the EOS might just reflect that with each additional word, the likelihood of an EOS increases.

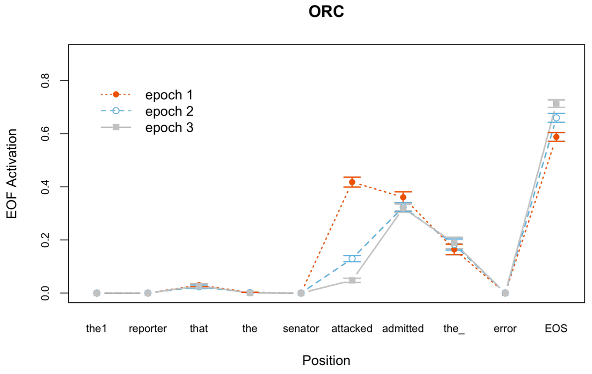


Figure 4: Output activaton of EOS at each position in sentences with ORCs, for three training epochs.

Figure 4 shows the activation of an EOS throughout the entire sentence. There is clearly little activation after the first NP, ruling out the first hypothesis. Moreover, there is a clear peak at the embedded verb, the matrix verb and the following determiner, whereas the subsequent noun shows almost zero EOS activation. An implicit counting mechanism that predicts increasing EOS activity with each step further downstream can hence be ruled out.

We want to pursue a third hypothesis: The network has not yet classified the relPro *that* correctly after the first epoch and confuses it with verbs. Note that the sequence *NP-that-NP* shares some distributional properties with regular transitive main clauses. The training corpora contained both simple main clauses and sentences with one or more RCs, most of which were center-embedded, i.e. modifying the first NP. All sentences started with an NP. The next word could either be a verb, or the relative pronoun *that*. Both were often followed by another NP, as *i.* transitive verbs in main clauses are followed by the direct object, and *ii.* the relative pronoun is followed by the subject-NP in ORCs. Due to this distributional resemblance, it seems reasonable that in early epochs, the networks are bad in distinguishing *NP-verb-NP* sequences from *NP-that-NP* sequences, or, to put it more simply, they confuse the relative pronoun with transitive verbs, at least in the local context of one NP to the left and one NP to the right. Hence, at the acquired level of grammatical knowledge, the EOS appears to be a feasible continuation for *NP-that-NP*, since it appears to mark the end of a simple transitive SVO main clause[4].

With more training, the networks slowly adapt to the fact that the relPro and verbs are not distributionally equivalent when a wider context is taken into account.

To substantiate this claim, we analyzed the hidden layer activities for all words in the corpus. There have been several proposals for analyzing distributed representations in neural networks, such as cluster analysis (Hinton, 1988), principle component & phrase state analysis (Elman, 1989), skeleton analysis (Mozer & Smolensky, 1989), contribution analysis (Sanger, 1989), which make the networks' representations and behavior more transparent. Since we are interested in how the SRNs have classified words, we analyzed hidden layer activities for each word averaged over test runs of one thousand random sentences. We present *multi-dimensional scaling* (MDS) data illustrating the internal grouping of words and indicating scaled euclidean distances between individual words (word groups). All stress values were below 0.1.

If the confusion of relPros and verbs is responsible for false EOS prediction, the hidden layer activations of relPros and verbs should be more similar in the first epoch than in later epochs.

Results

As figures 5 and 6 illustrate, euclidean distances between the relPro *that* and transitive verbs change considerably between epochs. The relPro is thus much more similar to verbs, especially transitive verbs, after the first training epoch than it is after the third, where

[4] There is even more distributional overlap between relpros and verbs: In SRCs, the relative pronoun *that* is immediately followed by a verb. However, even this local sequence is locally consistent with the verb classification of *that*, since in sentences with ORCs, the matrix verb immediately follows the embedded verb (it even follows a NP-verb sequence!). NP-that-verb-NP sequences are hence locally consistent with both the verb reading of *that*, since there is a NP-verb-verb-NP sequence contained in sentences with ORCs, and with the correct relative pronoun reading of *that*.

that builds an outlier categorie of its own. The hidden layer activities support the confusion hypothesis: After the first epoch, average activities of relPros resemble those of verbs much more than after the second and the third epoch.

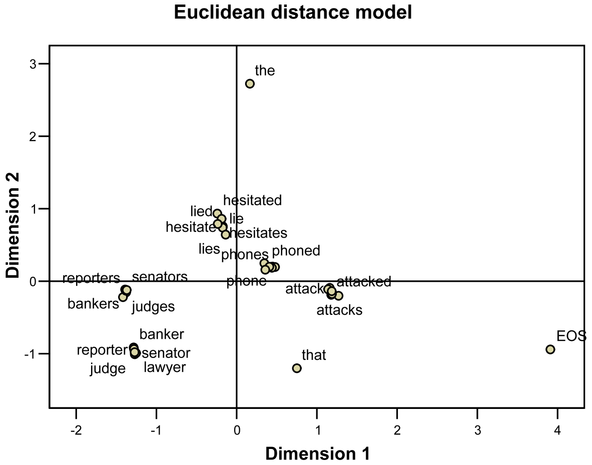


Figure 5: MDS plot of average hidden layer activations after epoch 1

In fact, relPros resemble transitive verbs more than intransitive verbs. These data clearly suggest that the biggest part of what is gained from training is the substantially better classification of the relPro. On the other hand it is also clear that relPros are not generally classified as verbs even in the first epoch. However, the hidden layer analyses reflect averaged hidden layer activities at the moment when the word is at the input, not after the entire NP-that-NP sequence.

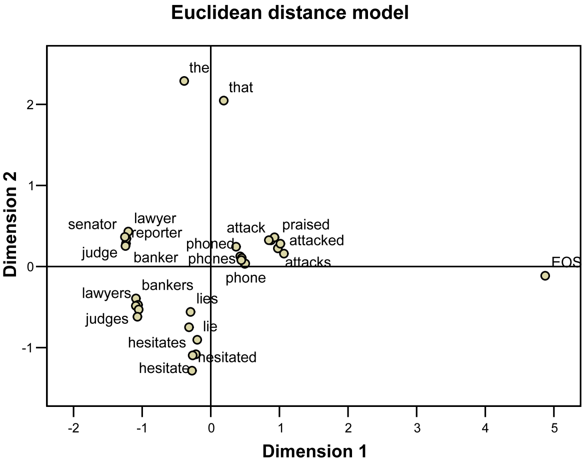


Figure 6. MDS plot of average hidden layer activations after epoch 3

Discussion of matrix verb results

The results on the matrix verbs strongly suggest that the GPE is mainly based on one or two false alarm components for SRCs and ORCs, respectively. In both sentences, the EOS is a major false alarm component.

In SRCs, the EOS-prediction follows a …-verb$_{transitive}$-NP sequence. Expecting an EOS here is locally legitimized by the word order in transitive main clauses, which end here in the majority of the cases. The false EOS prediction appears to be stable, and would probably survive even more training epochs, even though the activation of correct verbs is continuously growing throughout the epochs. These data suggest that the main reason for long reading times on matrix verbs in center embedded sentences is that readers, even the most experienced ones, expect the sentence to end here about as much as they expect a correct matrix verb. In the absence of further empirical data, we resort to questioning this empirical prediction on the grounds of plausibility. We are convinced that adult readers, even less experienced ones, would be quite surprised if the sentence ended after a simple center-embedded RC.

In ORCs, both false alarm predictions of the determiner and the EOS prediction follow a …-NP-verb$_{transitive}$ sequence. In this local context, the prediction of the determiner is legitimized by the word order in simple transitive main clauses, where verbs are followed by an NP. As in SRCs, this prediction indicates that, to a substantial degree, the networks ignore the fact that the RC is sub-ordinate. Contrary to the stable EOS prediction in SRCs however, the determiner prediction shrinks over time, indicating that the networks learned to widen their contextual window. The decreasing amount of false alarm activation is responsible for the global GPE reduction at the matrix verb. Although it appears odd that adult readers would run into this local trap, this result is modestly consistent with MCs *frequency* x *regularity* interpretation.

The false prediction of an EOS at this position seems a bit puzzling at first glance. The embedded verbs used here are transitive, as they have to combine with an object-NP in the test sentences. Even if the networks pursue a main clause analysis, they should predict a NP, but rule out an EOS. However, half of the transitive verbs used (*phones, phone, phoned, understands, understand, understood*) were also used as intransitives. It seems likely that the averaged GPEs are based on false predictions due to these verbs. A more detailed analysis, distinguishing strictly transitive and optional transitive verbs could clarify this issue. Also note that the false prediction of an EOS increases with experience. So the most experienced networks, and hence high span readers, are predicted to not really be surprised if the sentence ends after a center-embedded ORC. Once again, we are skeptical about this hypothesis.

In all cases, locally coherent continuations have distracted the network from the global necessity of a matrix verb at this position. More generally speaking, locally consistent false alarms were identified as the main source of processing difficulty.

Conclusion

We have argued that when predictions derived from connectionist models are presented, global error measures must be accompanied with detailed analyses of the output activation vectors to understand the source of the errors in the networks. A detailed analysis of false alarm components can hint at substantial acquisition deficits at the current stage of learning and at simulation artifacts caused by the choice of the grammar that the training corpora are generated from. In the present case, MC's networks were shown to make unrealistic continuation predictions based on classification errors (the relative pronoun *that* is considered a verb). However, identifying a flaw in a particular simulation hardly renders a general hypothesis invalid. Experience is a likely source of both construction specific complexity and inter-individual variation, and empirical support is beginning to materialize. For instance, Wells, Christiansen, Race, Acheson, and MacDonald (2009) showed that processing of relative clauses, and especially of ORCs, can be improved by training with RCs.

The activation analyses also revealed that the main source of complexity is the distraction induced by *locally coherent* continuations. Are adult language processors distracted by such false alarm predictions? Again, empirical support is beginning to surface. Tabor, Galantucci and Richardson (2004) provided data indicating that locally coherent but globally incoherent fragments can distract attention from the globally valid analysis in ambiguities. Konieczny (2005) revealed that syntactic errors produced by adding locally coherent words to a sentence were harder to detect than errors induced by locally incoherent words. Konieczny, Müller, Hachmann, Schwarzkopf and Wolfer (2009) showed in visual-world eyetracking experiments that local coherences are being interpreted during speech processing. Despite their misleading results, MC's approach helped identifying a fundamental processing phenomenon: interference by local coherences. Empirical data showing local coherence effects in real language processers provides support for the connectionist framework as a whole.

References

Christiansen, M. H. & Chater N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science, 23,* 157-205.

Daneman, M. & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Behavior*, *19*, 450-466.

Elman, J.L. (1989). *Representation and structure in connectionist models.* Technical Report CRL-8903. Center for Research in Language, University of California, San Diego.

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, *14*, 179-211.

Elman, J. L. (1991).Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7*, 195-224.

Gibson, E. (1998). Linguistic complexity: locality of syntactic dependencies. *Cognition*, *68,1*, 1-76.

Gordon, P. C.; Hendrick, R.; Johnson, M. (2001). Memory interference during language processing. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, *27(6)*, 1411-1423.

Hale, J. (2001) *A Probabilistic Earley Parser as a Psycholinguistic Model.* Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics. 159-166.

Hinton, G.E. (1988). *Representing part-whole hierarchies in connectionist networks.* Technical Report CRG-TR-88-2, Connectionist Research Group, University of Toronto.

Just, M. A. & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, *99*(1), 122-149.

King, J., & Just, M. A. (1991). Individual differences in syntactic processing: The role of working memory. *Journal of Memory & Language, 30,* 580-602.

Levy, R (2005). *Probabilistic models of word order and syntactic discontinuity.* Stanford University.

Konieczny, L. (2005). *The psychological reality of local coherences in sentence processing.* Proceedings of the 27th Annual Conference of the Cognitive Science Society. Stresa, Italy, August 2005.

Konieczny, L., Müller, D., Hachmann, W., Schwarzkopf, S., & Wolfer, S. (2009). *Local syntactic coherence interpretation. Evidence from a visual world study.* Paper presented at the 31st Annual Conference of the Cognitive Science Society.

Lewis, R.L. & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. Cognitive Science, 29, 375-419.

MacDonald, M. C. & Christiansen, M. H. (2002). Reassessing working memory: A comment on Just & Carpenter (1992) and Waters & Caplan (1996). *Psychological Review*, *109*, 35-54.

Mozer, M. C. and Smolensky, P. (1989). Skeletonization: A technique for trimming the fat from a network via relevance assessment. In: Touretzky, D. S. (ed.), *Advances in Neural Information Processing Systems 1*, 107-115. San Mateo, CA.

Sanger, D. (1989*). Contribution analysis: a technique for assigning responsibilities to hidden units in connectionist networks.* Boulder, Colorado.

Seidenberg, M. & McClelland, C. (1989). A distributed model of word recognition and naming. *Psychological Review, 96,* 523-568.

Tabor, W., Galantucci, B., Richardson, D. (2004). Effects of Merely Local Syntactic Coherence on Sentence Processing. *Journal of Memory and Language*, *50(4)*, 355-370.

Wells, J. B., Christiansen, M. H., Race, D. S., Acheson, D. J., MacDonald, M. C. (2009). Experience and sentence processing: Statistical learning and relative clause comprehension. *Cognitive Psychology 58*, 250-271

Multiple Object Manipulation: is structural modularity necessary? A study of the MOSAIC and CARMA models

Stéphane Lallée (stephane.lallee@inserm.fr)
INSERM U846, Stem-Cell and Brain Research Institute
Bron, France

Julien Diard and Stéphane Rousset
Laboratoire de Psychologie et NeuroCognition CNRS-UPMF
Grenoble, France

Abstract

A model that tackles the Multiple Object Manipulation task computationally solves a higly complex cognitive task. It needs to learn how to identify and predict the dynamics of various physical objects in different contexts in order to manipulate them. MOSAIC is a model based on the modularity hypothesis: it relies on multiple controllers, one for each object. In this paper we question this modularity characteristic. More precisely, we show that the MOSAIC convergence during learning is quite sensitive to parameter values. To solve this issue, we define another model (CARMA) which tackles the manipulation problem with a single controller. We provide experimental and theoretical evidence that tend to indicate that non-modularity is the most natural hypothesis.

Keywords: motor control; MOSAIC; CARMA; modularity; internal representation; neural network.

Introduction

There is, in the world, an infinity of objects with different physical behaviors. Despite this variability, humans can manipulate them with ease, from light origamis to heavy cups. For a given goal position, how do they select the correct force to apply? How are they able to accurately predict the displacements resulting from the applied forces? These two questions are central in object manipulation: control and prediction, respectively. If the physical characteristics of objects and their identity are known, or if there is a single object, this problem is easy to model and solve. Indeed, the dynamics of physical bodies are well described by Newton's equations. Given the starting position, and the applied forces, it is straightforward to compute the resulting trajectory.

The problem becomes much more difficult if the objects are numerous, and of unknown physical parameters. We call this the Multiple Object Manipulation task (MOM).

It is thought that natural cognitive systems are able to solve this problem because they are capable of good predictions in uncertain and unstable environments. Modeling this ability can provide insights and a better understanding of the possible brain structures involved in the process (Kawato, 2008). This has lead Gomi and Kawato to propose the MOSAIC model (MOdular Selection And Identification for Control) (Gomi & Kawato, 1993). This model solves both problems of object identification and object control simultaneously. The key feature of MOSAIC is that it uses neural networks in a modular way. In other words, the system has multiple distinct neural controllers, one for each object. It is able to choose which controller to use in order to manipulate an object, even without knowing the object identity explicitly.

The structural matching between object and controller, in MOSAIC, is a very strong hypothesis, that we question in this paper. To do so, we developed another model, CARMA (Centralized Architecture for Recognition and MAnipulation) which solves the same problem as MOSAIC but in a non-modular way. By comparing the properties of MOSAIC and CARMA, we study the object–controller coupling in both a theoretical and experimental manner. More precisely, we show how object specialization in MOSAIC is actually quite sensitive to the learning parameters, and how CARMA avoids this issue.

The rest of the paper is organized as follows. We first describe the experimentation framework, as well as the MOSAIC and CARMA models. Our experiments begin with a validation of the capacity of both models to solve the MOM task. We then study the way it is solved in more detail, particularly regarding the controller specialization in MOSAIC. We finally show how the notion of object is encoded as part of the network activation structure in CARMA.

Experimental framework

We replicate the task defined by Gomi and Kawato (Gomi & Kawato, 1993) and applied in their subsequent papers (Wolpert & Kawato, 1998; Haruno, Wolpert, & Kawato, 2001), as closely as possible. It is the simulation of an arm that moves an object on a one-dimensional axis. The task is to move the object according to a given trajectory: the simulation has to choose, at each time step, a force to apply to the object.

The task becomes a MOM task when the object to be moved is changed at a fixed frequency during the simulation, and this change occurs in a single time-step. The amount of information available to the system is quite limited: the physical characteristics of the various objects, and the change frequency are unknown. This turns a simple linear equation system into a difficult cognitive task. Fig. 1 shows an example of the task.

In the simulation, any physical object is treated as a damped spring-mass system. Each object is thus defined by 3 parameters (M,B,K), with M the mass, B the viscous damping coefficient and K the spring constant.

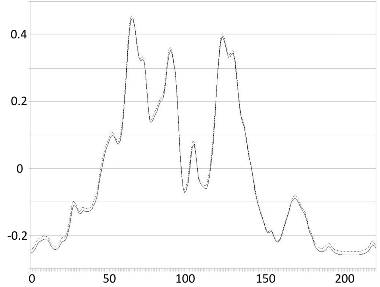


Figure 1: A sample desired trajectory to be followed (light) and the actual trajectory (dark) plotted against time.

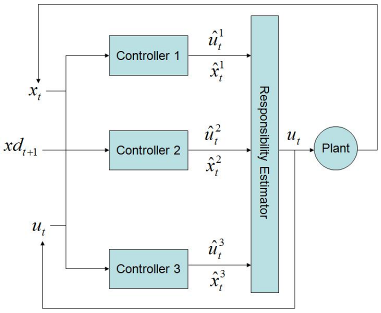


Figure 2: Global structure of MOSAIC with three controllers.

Time is treated as a discrete variable. We will use the following notations: for a given time index t, x_t is the object position, $\hat{x}_t$ the estimated object position, $\dot{x}_t$ is the object speed, O_t is the object identity, xd_t is the desired position, u_t is the applied force, and $\hat{u}_t$ is the estimated applied force.

A very simple plant model is used to compute, at each time step, the actual movement of the presented object O_t under the applied force u_t:

$$x_{t+1} = \frac{dt}{M}\left(u_t + \left(\frac{M}{dt} - B\right)\dot{x}_t - Kx_t\right) . \tag{1}$$

It is the only part of the simulation that actually knows the characteristics (M, B, K) of the objects O_t.

The MOSAIC model

The main idea of the MOSAIC model is to use multiple parallel controllers, each one suited for each particular object. Each controller is divided into three modules: the first encodes a Direct model, the second encodes an Inverse model and the last is a Responsibility Predictor, and is used to take into account visual information. Each of these modules is implemented using artificial neural networks (ANN).

Multiple controllers

At the highest level, the MOSAIC architecture is illustrated Fig. 2. Each controller is designed to predict the behavior of a particular object, but the actual control is done by all of them. At each time step, the responsibility of each controller is estimated. These responsibilities reflect the controllers' abilities to predict adequately the behavior of the current object. They are used in 2 ways: first, they weigh the contribution of each controller towards the final command (the controllers that predict well have greater control over the object), and second, they weigh the learning rate of each controller (the best predictors learn more and learn faster).

The responsibility λ_t^i of the i-th controller is computed by comparing the current position x_t with the position $\hat{x}_t^i$ estimated by controller i, as follows (Wolpert & Kawato, 1998):

$$\lambda_t^i = \frac{e^{-(x_t - \hat{x}_t^i)^2/\sigma^2}}{\sum_{j=1}^{n} e^{-(x_t - \hat{x}_t^j)^2/\sigma^2}} , \tag{2}$$

with σ a scaling parameter of this soft-max function.

Because the sum of all responsibilities is 1, they can be interpreted as probabilities: λ_t^i is the probability that the i-th controller is the best one to control the current object, according to the prediction errors. The σ parameter then regulates the competition between controllers.

Since the responsibilities gate both learning and forces, they are the heart of MOSAIC. The controller which was the best at predicting the object trajectory will have the highest responsibility, will learn more about controlling this object, which will help it predict more accurately, etc. Theoretically, it is supposed to make every controller specialize and converge to being the controller of a specific object.

Controller architecture

Each controller includes a Direct and Inverse model for a given object. The Direct model F predicts the future position given the current position, current speed and last applied control, while the Inverse model G computes the command to apply to go from a current position and speed to a desired future position:

$$\hat{x}_{t+1} = F(x_t, \dot{x}_t, u_t) , \tag{3}$$

$$\hat{u}_t = G(x_t, \dot{x}_t, xd_{t+1}) . \tag{4}$$

Each is implemented using linear ANNs (without hidden layers), with 4 nodes each. Indeed, for a single object, they approximate very simple equations with two unknown quantities and three parameters (M, B, K). The task of the Backpropagation algorithm is to adapt the weights of the network to give an implicit approximation of these parameters.

Visual modality

So far, the responsibilities are only based on the controllers' prediction error: it is feedback of a purely motor nature.

To more closely approximate the cognitive task of object manipulation, a third module is added to each controller, the Responsibility Predictor (RP), which simulates feedforward visual information. A visual representation is added to each object, in the form of a 3×3 matrix M_v of boolean pixels. The role of the RP is, given this visual representation, to predict the responsibility $\hat{\lambda}_t^i$ of its controller before any motion is performed. This feedforward responsibility estimation is merged

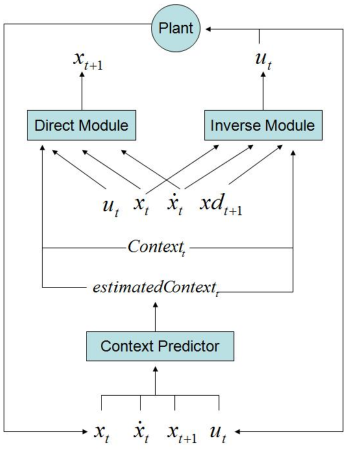


Figure 3: Global structure of the CARMA model. The direct, inverse and context predictor modules are four-layer MLPs.

with the motor feedback, and the responsibilities of Equation (2) are replaced by:

$$\lambda_t^i = \frac{\hat{\lambda}_t^i \times e^{-(x_t-\hat{x}_t^i)^2/\sigma^2}}{\sum_{j=1}^n \hat{\lambda}_t^j \times e^{-(x_t-\hat{x}_t^j)^2/\sigma^2}} \,. \tag{5}$$

The CARMA model

The global CARMA model takes the same inputs as MOSAIC $(x_t, \dot{x}_t, x_{t+1}, u_t, M_v)$, and is essentially structured like a single controller of MOSAIC (see Fig. 3): it is made of three modules, which are a Direct model, an Inverse model and a Contextual Predictor (CP). Each of these thus encodes knowledge relevant to several objects: therefore, they are more computationally complex than in MOSAIC. Whereas in MOSAIC, each module of a controller could be a linear ANN, in CARMA, each module is a four-layer Multi-Layer Perceptron (MLP), with an input layer, an output layer, and two hidden layers (with 10 and 2 nodes for the Direct and Inverse models, 10 and 5 nodes for the RP; the full CARMA model we used thus had 96 nodes).

Direct and Inverse models

The Direct and Inverse models have the same outputs as in MOSAIC, and the same inputs, augmented with two 3×3 matrices, which represent the actual visual input (real context) and the estimated visual input (estimated context). The real context input is the visual representation M_v of the manipulated object. Given the same input position, speed and force, this enables the Direct and Inverse models to output different values for different objects, according to this contextual input.

Context Predictor

The purpose of the CP is to identify the manipulated object, based on its dynamics. It uses motor feedback information to predict what should be the visual representation of the manipulated object. This estimated context can then be compared with the actual visual context; this comparison and the resulting difference drives the learning phase of CARMA. After convergence, this difference becomes very close to zero: the estimated and real context are almost always equal to one another.

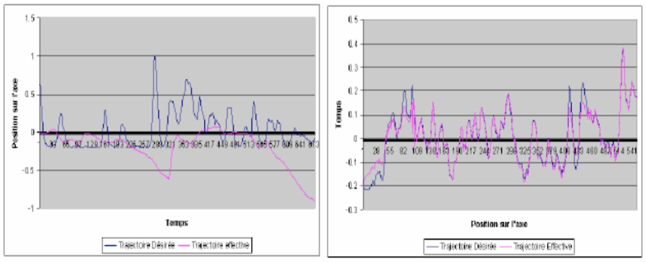


Figure 4: Solving the manipulation task with CARMA, before learning (left) and after learning (right). Three objects are switched every 20 time steps.

In some experiments, we also used the difference between the real and estimated contexts as a mechanism to handle illusions, where the system was fed a visual input which corresponded to a different object than the one actually manipulated. However, the details of these experiments are beyond the scope of this paper.

Experiments

In this section we first show that both systems can handle and solve the MOM task. We then analyze in more detail the way MOSAIC solves it. In particular, we show that MOSAIC's controllers do not become specialized for specific objects, except in special cases. We then study the mechanisms involved in CARMA for solving the MOM task.

Solving the task: experimental validation

MOSAIC and CARMA can both solve the MOM task without any problem. In Fig. 4 the results were recorded from CARMA controlling a set of 3 different objects, before and after the learning phase. Similar plots, obtained with MOSAIC, are not shown.

In order to prove that learning how to manipulate one object is not sufficient to manipulate all of them, we trained both systems on one given object, and, after convergence, gave them a different object (test object). We observed very low performance overall, as expected.

However, two cases could clearly be identified. If the system was trained on a lighter object than the test object, it would subsequently generate insufficient forces during test, which would not displace sufficiently the test object: the general trends of the trajectory would be followed, with large errors, large delays and slow convergence to the trajectory (Fig. 5, top). On the other hand, if the training object was heavier than the test object, the system would subsequently generate excessive forces, which would lead to overshoots and oscillating behaviors(Fig. 5, bottom). This was observed both in MOSAIC and CARMA.

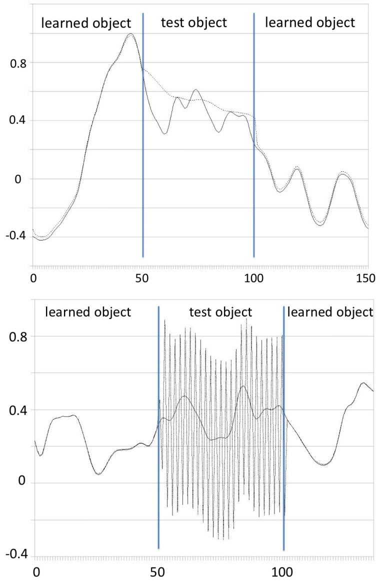


Figure 5: Using the learned controllers for an unknown, test object (from time step 50 to 100) either leads to damped and delayed control (top) when the test object is lighter, or oscillations (bottom) when the test object is heavier.

MOM with multiple controllers: MOSAIC

We then studied the behavior of MOSAIC for a MOM task.

The MOSAIC model relies on the property that, after convergence, each controller is specialized, in the sense that it should be responsible for the control of one and only one object. This property is well described (Haruno et al., 2001), but, unfortunately, we were not able to replicate it reliably. Indeed, according to our simulations, this specialization is not systematic: most of the time, it does not occur.

In typical cases, we observed that one controller acquires a large responsibility over all the objects, even if they have widely different physical characteristics (M,B,K). For instance, we presented the system with four objects (A to D), with different dynamics, and trained a MOSAIC system with four controllers (0 to 3). Despite the variability in the objects, we usually observed that one of the controllers was mainly responsible for most of the output commands, with marginal specialization in the remaining controllers. One typical case is shown Fig. 6.

Conditions for object specialization in MOSAIC

We discovered that object specialization in MOSAIC was quite sensitive to the values of the learning algorithm's parameters. We now detail them and explain their influence.

Learning rate The learning algorithm for the Inverse and Direct modules of each controller is the Backpropagation algorithm. If a controller is given a high responsibility for a short time, it learns a lot more than the other controllers and then already has an accurate control on all objects; we therefore used a low value (0.001).

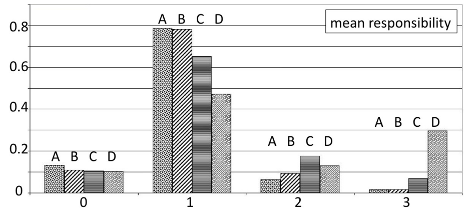


Figure 6: Mean responsibilities along a typical trajectory for a 4-controller MOSAIC (0 to 3) with 4 objects (A to D): here, controller 1 (second block of bars) takes care of most of the control for all objects, while controllers 2 and 3 are marginally specialized for objects C and D, respectively. Object A: $(M = 1, B = 2, K = 8)$, object B: $(M = 1, B = 8, K = 1)$, object C: $(M = 3, B = 1, K = .7)$, object D: $(M = 8, B = 2, K = 1)$.

Object switching frequency This frequency has a crucial importance during the learning phase. If the frequency is too low, the situation is similar to sequential training: large training on one object, then on another one. In this case one controller becomes perfect for one object, and is also better for the other objects than untrained controllers: this is the property we illustrated previously (see Fig. 5). In our simulations we switched objects frequently, every 20 time steps.

Controller competition parameter σ This parameter seems to be key for controller specialization. Unfortunately, the way it is defined in MOSAIC is unclear: it is only said to be "tuned by hand over the course of the simulation " (Haruno et al., 2001, 2211). We therefore investigated three cases.

If σ is set to a low value, the competition is strong between controllers: as soon as one controller specializes for one object, as it is also better than untrained controllers on the other objects (see Fig. 5), it wins control over all objects. Moreover, only one controller is active at a time: the system becomes similar to a mixture of experts system (Jacobs, Jordan, Nowlan, & Hinton, 1991).

If σ is set to a high value, the cooperation is strong between controllers: the responsibilities are so well distributed that almost no specialization appears. All controllers have nearly the same responsibilities so they share the control of the objects. Despite this, the manipulation error remains small.

The last case is to have σ vary during training, and more precisely, decrease over the training period. Indeed, with an initial cooperation and shared control between controllers, they all quickly learn the main characteristics of the motion equation, and the main aspect of control: apply a positive force when the object needs to go up, a negative force otherwise. When this is trained into all controllers, then σ can be slowly decreased so that controllers, in turn, pick more precise characteristics of the physical behaviors of the objects. Finally, σ should decrease over time, but not in a linear way since the convergence of the ANNs is not linear. When it is correctly tuned, a specialization can be observed (Haruno et al., 2001). Unfortunately, the function $\sigma(t)$, according to our

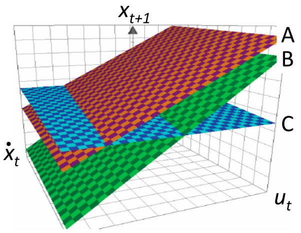


Figure 7: Dynamics for 3 objects (A to C) with different characteristics (M,B,K). The axes are: speed $\dot{x}_t$, applied force u_t and next position x_{t+1}. A fixed starting position is assumed. Each plane corresponds to one object.

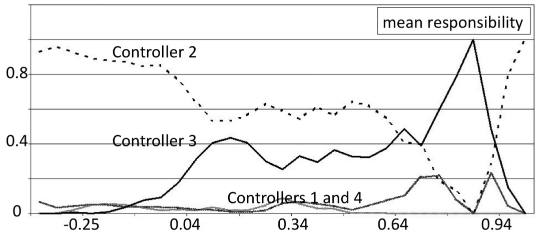


Figure 8: Responsibilities for 4 controllers in MOSAIC, plotted against the desired position xd_t.

simulations, also appears to be dependent of problem specific factors, including the number of presented objects and their characteristics (M,B,K); we do not foresee an easy way in which $\sigma(t)$ could be automatically defined in order to be suitable for a given instance of the MOM task.

What is learned by controllers in MOSAIC?

A close inspection of the physical manipulation problem shows that some structural properties are the same for all objects: for instance, discrimination between the pulling cases (negative force) and the pushing cases (positive force).

Further investigations also show that some objects with different physical characteristics become indistinguishable for some trajectories. For instance, consider two objects with the same mass M and spring constant K but with different damping factors B_1 and B_2: when the speed along the trajectory is small, these two objects behave similarly, and a single controller can easily control both. On the other hand, when the speed is high, the forces to output are different, and two controllers are needed. This is illustrated Fig. 7: we plotted the motion equation (1) for three different objects. In order to represent it on a 3D plot, we set a fixed starting position x_t. In this projection of the Space Of Dynamics (SOD), we can observe that objects are intersecting planes. At the intersections, the objects are indistinguishable.

Because objects are indistinguishable for some trajectories, we hypothesized that controllers in MOSAIC would not become specialized for specific objects, but rather, for object–trajectories combinations. We thus plotted the controller responsibilities after learning against the trajectory characteristics. We show Fig. 8 the responsibilities for 4 controllers (0 to 3) manipulating 3 objects (A to C) as a function of the desired position xd_t: we observe that when $xd_t < 0$, controller 2 takes almost full control, that there is a shared control between controllers 2 and 3 for $xd_t \in [.1,.7]$, and that controller 3 is specialized for $xd_t \in [.7,.9]$, *independently of the object being manipulated*. Therefore, it appears that MOSAIC controllers indeed specialize for motion subspaces.

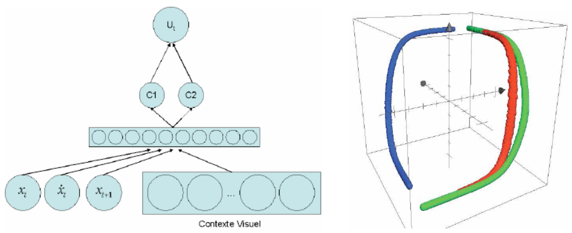


Figure 9: On the left, the CARMA Inverse module with an additional two-node layer for investigating the structure of the learned network. On the right, activity plot of the Inverse module; the axes are the values in the 2D added layer (X and Y) and the output value of the network (Z).

MOM with a single controller: CARMA

Since CARMA solves the MOM task with a single controller, and because it does not encode objects in its structure, we studied the way different objects were represented in the Direct model and Inverse model ANNs after learning.

We first quickly describe a new method of plotting the activation of a MLP neural network, and then use it on CARMA to investigate its internal representation of objects.

In a MLP, each layer is a transformation of the input space that can have a different dimensionality. If we add a two-node layer to the network, it is possible to extract a two-dimensional transformation of the input space and plot it. To generate the plotting data, we first train the network, then disable learning, submit to the network a random input activity on the nodes of interest, propagate it through the network, and, finally, record the activity of the two-dimensional hidden layer. This process is iterated until enough data is collected to have a good representation of the input space.

We used this method on CARMA's Inverse module (Fig. 9, left). By logging the activity of the two-dimensional hidden layer activity and the one-dimensional output layer we can draw a 3D plot of the function approximated by the whole module. In the case of a network trained on multiple objects, this representation gives information about the internal representations of objects. The most interesting result is that the function is fragmented: multiple long shapes are partially merged (Fig. 9, right). The number of shapes is equal to the number of objects learned by the system. With this method we get a clear representation of what an object is for CARMA: the concept of object is no longer a structural property of the model, it a contiguous set of activities in the set of possible activations in the network.

In the activity plot, the proximity of the shapes provides useful information. Some of them are merged, meaning that the objects are indistinguishable based on their dynamics. Furthermore, only one factor modifies the shape positions in the plot: the visual representation of the object. In other words, CARMA produces different outputs for different visual representation. This means that CARMA learns the motion equation, and uses the visual representation of the object as its parameters. Thus, the characteristics (M,B,K) are encoded in the internal visual representation. This encoding is the key point of the Context Predictor module.

Recall that the Context Predictor inputs are from the space of object dynamics, and its output is an estimated visual context: the CP learns a mapping from object dynamics to the visual representation space. In other words, it associates physical behaviors of objects with their appearances. The module never computes the (M,B,K) parameters explicitly, but encodes these in a visual space.

There is an interesting analogy with the way humans are not able to exactly ascertain the mass of an object. It is easy to know that one object is heavier than another, but very difficult to provide a precise estimation of a mass. Indeed, humans probably encode mass in a non-numeric space which would be a mixture of volume, aspect, dynamic experienced by motor experience, etc.

To further study this analogy, it would be fruitful to train the system and verify whether and how similar visual representations are associated with objects with similar dynamics; in other words, study the metrics of the transformation between the visual space and the space of the (M,B,K) parameters. With manually designed visual representations (*e.g.* objects with very similar visual representations but very different dynamics), it would be possible to test the predictions made by the Contextual Predictor.

Discussion

We presented and tested two systems, MOSAIC and CARMA, designed to solve the Multiple Object Manipulation task. The main difference between them is the modularity hypothesis: MOSAIC assumes that objects are encoded in a spatial way, into the model structure; whereas CARMA builds a function which handles all objects. Our experimental study has shown that, in MOSAIC, the controller–object association is not systematic and mainly relies on a human tuned parameter. Most of the time, controllers specialize on complex mixtures of trajectory, motion and object, which we have shown to be a central property of the CARMA model.

The non-modular approach was criticized by the authors of MOSAIC. According to them, for instance, a single controller would be too computationally complex. Actually, for the same problem, CARMA uses less neurons than MOSAIC. Indeed, in CARMA, the computational power comes from the number of nodes in hidden layers, while in MOSAIC, complete Direct and Inverse models are duplicated for each additional object. For instance, our CARMA implementation, with 10 hidden nodes in the Direct and Inverse models, and a total of 96 nodes, solves the MOM task with 10 objects. In MOSAIC, it would require 8*10 objects + 20 (for the RP) = 100 neurons. We believe that the difference would grow for additional objects, as CARMA with a few more nodes would treat a large number of additional objects (as we illustrated experimentally but did not expose in detail here).

They also suspected a slow adaptation to context variation; however, there is no delay in CARMA since the context is what defines the output of the system. The last point is the sensibility to catastrophic unlearning, which we did not study in this paper, but which has been solved elsewhere on similar single controllers, by a method that can easily be adapted to CARMA (Ans & Rousset, 2000).

Studying MOSAIC has implications beyond the scope of pure mathematical modeling. Indeed, the modularity hypothesized in MOSAIC – one controller for one object, and therefore, a spatial, structural encoding of objects in the global controller – is taken as a starting point of some recent brain imagery studies (Imamizu et al., 2000; Imamizu, Kuroda, Miyauchi, Yoshioka, & Kawato, 2003; Ito, 2000). Therefore, equivalents of this structural object endoding are looked for in the biological substrate; there is here the risk of an interpretation bias, resulting from taking for granted a model which is too specific.

References

Ans, B., & Rousset, S. (2000). Neural networks with a self-refreshing memory: Knowledge transfer in sequential learning tasks without catastrophic forgetting. *Connection science*, *12*, 1–19.

Gomi, H., & Kawato, M. (1993). Recognition of manipulated objects by motor learning with modular architecture networks. *Neural Networks*, *6*, 485–497.

Haruno, M., Wolpert, D. M., & Kawato, M. (2001). MOSAIC model for sensorimotor learning and control. *Neural Computation*, *13*(10), 2201–2220.

Imamizu, H., Kuroda, T., Miyauchi, S., Yoshioka, T., & Kawato, M. (2003). Modular organization of internal models of tools in the human cerebellum. *Proceedings of the National Academy of Science (PNAS)*, *100*(9), 5461–5466.

Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., Pütz, B., et al. (2000). Human cerebellar activity reflecting an acquired internal model of a new tool. *Nature*, *403*, 192–195.

Ito, M. (2000). Neurobiology: Internal model visualized. *Nature*, *403*, 153–154.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*(1), 79–87.

Kawato, M. (2008). From 'understanding the brain by creating the brain' towards manipulative neuroscience. *Phil. Trans. R. Soc. B*, *363*, 2201–2214.

Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*(7-8), 1317–1329.

“Lateral Inhibition” in a Fully Distributed Connectionist Architecture

Simon D. Levy (levys@wlu.edu)
Department of Computer Science, Washington and Lee University
Lexington, VA 24450 USA

Ross W. Gayler (r.gayler@gmail.com)
Department of Psychology, La Trobe University
Victoria 3086 Australia

Abstract

We present a fully distributed connectionist architecture supporting lateral inhibition / winner-takes all competition. All items (individuals, relations, and structures) are represented by high-dimensional distributed vectors, and (multi)sets of items as the sum of such vectors. The architecture uses a neurally plausible permutation circuit to support a multiset intersection operation without decomposing the summed vector into its constituent items or requiring more hardware for more complex representations. Iterating this operation produces a vector in which an initially slightly favored item comes to dominate the others. This result (1) challenges the view that lateral inhibition calls for localist representation; and (2) points toward a neural implementation where more complex representations do not require more complex hardware.

Keywords: Lateral inhibition; winner-takes-all; connectionism; distributed representation; Vector Symbolic Architecture

Introduction

Connectionist representations are typically classified as localist, distributed, or some combination of both. In a localist representation each node corresponds to a single item or concept. In a distributed representation each node participates in the representation of every concept, and each concept is “spread out” (distributed) among every node. Proponents of localist representation cite simplicity and transparency as benefits of localist coding. Proponents of distributed representations argue that the robustness of such representations in the presence of noise makes them more plausible and appealing, and cite related impressive work on modeling neuropsychological disorders using distributed connectionist representations. For a review see Olson & Humphreys (1997). A comprehensive argument for distributed representations is of course beyond the scope of this article. We will focus here instead on a particular capability that appears to be exclusive to localist representations, and will provide an alternative analysis using a distributed representation.

In a 2000 target article in *Behavioral and Brain Sciences*, Page (2000) argues for a “generalized localist model” with a localist representation on one layer and general (distributed) o representations on the others. Each node in the localist layer is associated with a category, and a lateral inhibition (winner-takes-all competition) function is used, allowing the localist layer to act as a classifier for (distributed) patterns on an incoming layer. Indeed, the ability of localist representations to support competitive classification seems to be the main appeal of localism, as suggested by the remarks of the commentators who supported Page’s position (*e.g.* Phaf & Wolters, 2000).

In this article we will argue that localist representations are not necessary to support winner-takes-all competition or lateral inhibition in general. We will present a fully distributed connectionist architecture supporting lateral inhibition / winner-takes all behavior, in which all items (individuals, relations, and structures) are represented by high-dimensional distributed vectors, and (multi)sets of items as the sum of such vectors. Unlike a localist representation, such representations are based on a fixed neural architecture that does not need to grow as new representational categories are added.

Problems with Localism

The greatest challenge to connectionist accounts of cognition continues to be the problem of compostionality, that is, the problem of how to put simpler items like words and concepts together to make more complex structures like sentences and propositions (Fodor & Pylyshyn, 1988; Jackendoff, 2002). Localist connectionism addresses this challenge by assigning one neuron or pool of neurons to each item, and employing additional (pools of) neurons as higher-order elements for organizing the simpler items via physical connections or temporal synchrony. For example, the Neural Blackboard Architecture of van der Velde (2006) builds sentences out of words via “structure assemblies” corresponding to traditional syntactic categories like Noun Phrase and Verb Phrase. Hummel and Holyoak’s LISA model of analogical mapping (Hummel & Holyoak, 1997) uses higher-order assemblies to represent the bindings of individuals to semantic roles like agent and patient.

In a forthcoming article, Stewart and Eliasmith (Stewart & Eliasmith, forthcoming) provide a detailed analysis of the computational complexity entailed by localist accounts of composition. This analysis suggests that the need to have physical connections between all pairs of items causes localist representations lead to a combinatorial explosion when applied to realistically-sized item inventories, such as the vocabularies of natural languages. An alternative approach, which dates back to the work of Pollack (1990) and others, attempts to encode structures of arbitrary complexity on a fixed-size connectionist architecture.[1] Commenting on Pol-

[1] A serious limitation of Pollack’s Recursive Auto-Associative (RAAM) network was the need to *learn* representations (via backpropagation). The work presented here avoids the need for learning,

lack's results, Hammerton (1998) notes that it is important to consider whether the representations produced by such architectures can be manipulated holistically, or whether they require "functional localism", such as serial extraction of components, in order to support useful computations. Even if it is neurally plausible, which seems unlikely, functional localism strikes us as essentially an implementation of classical symbol processing (cf. Fodor & Pylyshyn, 1988), foregoing much of the appeal of connectionism.

Another problem with localist implementation of lateral inhibition is that the system can only implement winner-takes-all; that is, the result is the choice of 1 out of k alternatives. For some problems, however, it would be more appropriate to have a *set* of answers returned. It is not clear how this would be achievable with localist winner-take-all implementation. What is needed is a new type of network that exhibits the attractor dynamics of localist winner-takes-all networks, but which can converge simultaneously to a *set* of items, rather than a single item.

The work presented here addresses these issues, providing a holistic implementation of an operation previously thought to require localist coding.

Vector Symbolic Architectures

Vector Symbolic Architecture is a name that we coined to describe a class of connectionist models that use high-dimensional vectors (with as few as 1000 dimensions, but more typically around 10,000) of low-precision numbers to encode structured information as distributed representations. That is, VSAs can represent complex entities such as trees and graphs; and every such entity, no matter how simple or complex, is represented by a pattern of activation distributed over all the elements of the vector. This general class of architectures traces its origins to the tensor product work of Smolensky (1990), but avoids the exponential growth in dimensionality of tensor products. The currently available VSAs employ three types of operation on vectors: a multiplication-like operator, an addition-like operator, and a permutation-like operator. The multiplication-like operation is used to associate or bind vectors. The addition-like operation is used to superpose vectors or add them to a set. The permutation-like operation is used to quote or protect vectors from the other operations.

The use of hyperdimensional vectors to represent symbols and their combinations provides a number of mathematically desirable and biologically realistic features. A hyperdimensional vector space can contain as many mutually orthogonal vectors as there are dimensions, and exponentially many almost-orthogonal vectors (Hecht-Nielsen, 1994), thereby supporting the representation of astronomically large numbers of distinct items. Such representations are also highly robust to noise: a significant fraction of the values in a vector can be randomly changed before it becomes more similar to another vector than to its original form. To cite a result from a forthcoming paper by Kanerva (in press): *When meaningful entities are represented by 10,000-[element] vectors, many of the bits can be changed more than a third by natural variation in stimulus and by random errors and noise, and the resulting vector can still be identified with the correct one, in that it is closer to the original "error-free" vector than to any unrelated vector chosen so far, with near certainty.* It is also possible to implement such vectors in a spiking neuron model (Eliasmith, 2005), lending them a further degree of biological plausibility.

The main difference among types of VSAs is in the kind of numbers used as vector elements and the related choice of multiplication-like operation. Holographic Reduced Representations (Plate, 2003) use real numbers and circular convolution. Binary Spatter Codes (Kanerva, 1994) use binary (Boolean) values and elementwise exclusive-or. MAP (Multiply, Add, Permute) coding (Gayler, 1998) uses bipolar (-1/+1) values and elementwise multiplication. A useful feature of BSC and MAP is that every vector is its own multiplicative inverse: multiplying a vector by itself elementwise yields the multiplicative identity vector ($A*A = \mathbf{1} = B*B$, where $\mathbf{1}$ is the identity vector, but $A+A = 2A$). As in ordinary algebra, multiplication and addition are associative and commutative, and multiplication distributes over addition.

We used MAP in the work described here. In MAP, properties are *accumulated* through vector addition; hence, it is trivial to have multiple, self-reinforcing copies of the same property (vector) in a single representation. For example, given a vector representation A of the property *affluent* and a vector representation B of the property *brave*, the representation $A+A+B = 2A+B$ could represent being very affluent and somewhat brave. Second, the *association* of two representations through elementwise multiplication produces a third representation that is completely dissimilar from both elements. If C represents an individual, say, Charlie, the proposition that Charlie is brave could be represented as $B*C$, whose similarity (vector cosine) with both B and C is close to zero. Together, these facts mean that a given entity can be associated with a large number of properties (and vice versa): $C*(2A+B)$, etc.

Without an additional mechanism, self-cancellation *would* pose a challenge when copies of structures are embedded in themselves recursively. For example, if D_1 and D_2 represented the semantic roles *doubter* and *(thing) doubted*, then one possible way to represent the proposition *Bill doubted that Charlie doubted that Ed is affluent* as

$$D_1*B+D_2*(D_1*C+D_2*A*E)$$

Without further modification, the two copies of D_2 would have the undesired affect of canceling each other out. As mentioned above and discussed at length in (Levy, to appear), the permutation operator of the MAP architecture provides a neurally plausible mechanism for quoting or protecting vectors in these situations.

by relying on fixed, constant-time mechanisms for associating and composing vector representations.

As an example of holistic computation in MAP, consider the common task of retrieving a set of items associated with a given property. We imagine three individuals: A and B having property P and C having property Q. In a MAP encoding, each individual and property would be encoded in a hyper-dimensional vector, and the association of properties with individuals would be the vector sum of the elementwise products between each individual and its property:

$$V = A*P + B*P + C*Q$$

To retrieve the set of individuals having property P, we multiply the "knowledge-base" vector V by P. The self-inverse property of MAP produces a representation of the individuals A and B, as well as a "noise" component not corresponding to any individual or property "known" to the system:

$$\begin{gathered} P*V = \\ P*(A*P + B*P + C*Q) = \\ A*P*P \; + \; B*P*P \; + \; C*Q*P = \\ A \; + \; B \; + \; C*Q*P = \\ A \; + \; B \; + \; noise \end{gathered}$$

Comparing this resulting vector to the vectors for each of the individuals will yield a high similarity (dot product, cosine) between the result vector and both A and B, but not C. In other words, a single holistic computation on two vectors (P and V) has retrieved structurally sensitive information about distinct individuals, without (1) the need for explicit physical connections among the individuals (and the concomitant additional representational hardware) or (2) a functionally localist decomposition. This power comes at the cost of of noise in the retrieved representation, which is not a dealbreaker for this example. If noise becomes a problem (as it can in recurrent circuits like the analogy-mapping circuit described below, where noise accumulates), the noise can be removed from the result vector by passing the vector through a "cleanup memory" that stores only the meaningful items, or vector directions: here, A, B, and C.

The issue of noise in VSA is rather subtle. In a localist representation there are distinguished directions in the vector space that correspond to the individual units, because individual units represent individual concepts. In VSA there are no inherently distinguished directions. For example, the vector X might represent the concept A, but it could just as well represent $A+B$ or $C*D+E$, etc. The functional equivalent of distinguished directions is provided by the contents of the cleanup memory, which are initialized for a particular problem. Noise is then any pattern which is not stored in cleanup memory. Unlike localist representations, which require reconfiguring the "hardware" for each new problem, VSA reuses the same fixed cleanup hardware (e.g. an autoassociative Hofpield network) for every problem.

Lateral Inhibition as Self-Intersection

Consider a situation in which three categories A and B, and C are competing with one another on a given neural layer L_2, to classify input patterns on an incoming layer L_1. An example localist implementation is shown in Figure 1. Each node in L_2 has an inhibitory connection to every other node in that layer. The connections from L_1 to L_2 can be interpreted as setting the state of L_2 to reflect the initial evidence for each of the categories. The inhibitory connections within L_2 implement a recurrent process that increases the differences between the most supported category and the other categories.

We can interpret the state of L_2 as a multiset - a set of weighted elements. Each category X_i (here, A, B, or C) is weighted by a non-negative real-valued coefficient k_i that reflects the importance of X_i in the multiset, with $0 \leq k_i \leq 1$. Given this interpretation of the L_2 state as a multiset we need a multiset operation that increases the differences between the most supported category and the other categories. We do this with multiset intersection (multiplication of the corresponding category weights) and normalization (constraining the sum of the category weights to be constant).

To see what we mean by multiset intersection, consider multisets $X = \{k_1A, k_2B, k_3C\}, Y = \{k_4A, k_5B, k_6C\}$. Intersecting X and Y would would yield a multiset $\{k_1k_4A, k_2k_5B, k_3k_6C\}$. Intersecting X with itself would yield $(k_1)^2A \; + \; (k_2)^2B \; + \; (k_3)^2C$, magnifying the differences between the k_i. Normalization of the result forces the smaller k_i towards zero. The repeated application of self-intersection with normalization yields a similar dynamic to lateral inhibition thereby implementing winner-takes-all competition.

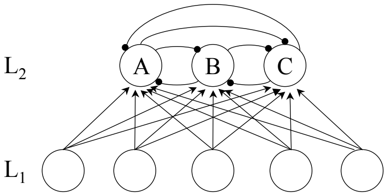


Figure 1: Lateral inhibition in a localist network

In a localist network like the one in Figure 1, the multiset coefficients k_i correspond to the activations of the nodes in the L_2 layer. In VSA, a multiset is represented as a single vector, for example, $k_1A \; + \; k_2B \; + \; k_3C$ where A, B, and C are hyperdimensional vectors and k_1, k_2, and k_3 are non-negative scalars. Note that this network can *only* represent a choice between A, B, and C. No other category can be considered without modifying the physical structure of the network.

How are we to perform the multiset intersection of two such vectors? Because of the self-cancellation property of the MAP architecture, simple elementwise multiplication (the standard MAP product operator) of the two vectors will not implement this operation. We could extract the k_i by iterating through each of the vectors A, B, and C and dividing x and y elementwise by each mapping, but this is the very kind of functionally localist approach that we are trying to avoid.

To implement this intersection operator in a holistic, distributed manner we exploit the third component of the MAP architecture: permutation. For explanatory purposes we can conceive of our solution as a simple register-based machine, where (as in a traditional von Neumann architecture), each register holds a temporary stage of the computation. (In our version, of course, the register contents are hyperdimensional vectors.). As depicted in Figure 2, our solution works as follows: 1: and 2: are registers loaded with the vectors representing the multisets to be intersected. $\mathrm{P}_1()$ computes some fixed permutation of the vector in 1:, and $\mathrm{P}_2()$ computes a different fixed permutation of the vector in 2: (randomly chosen permutations are sufficient). Register 3: contains the product (via elementwise multiplication) of these permuted vectors. Register 4: is another variety of "cleanup" memory (a constant vector value) pre-loaded with each of the principal vectors transformed by multiplying it with permutations of itself; *i.e.*, $4 := \Sigma_{i=1}^{n} X_i * \mathrm{P}_1(X_i) * \mathrm{P}_2(X_i)$. In other words, register 4: indicates the items of interest to the system and is functionally analogous to the L_2 units in the localist network; however, the contents of the register can be changed at any time without modifying the underlying hardware. Note so that each of these registers contains a high-dimensional vector representing an arbitrarily complex multiset, and each arrow in Figure 2 represents the transfer of a high-dimensional vector.

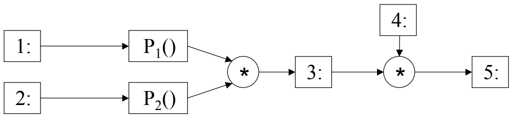


Figure 2: A neural circuit for vector intersection.

In brief, the circuit in Figure 2 works by guaranteeing that the permutations will cancel for only the subset of X_i present in both input registers, with the other X_i being rendered as random noise. In order to improve noise-reduction it is necessary to take the sum over several such intersection circuits, each based on different permutations. This sum over permutations has a natural interpretation in the synaptic connections between neural layers of sigma-pi units. Each unit (neuron) in one layer calculates the sum over many products of a few inputs from units in the prior layer. The apparent complexity of Figure 2 is a consequence of drawing it for explanatory clarity rather than computational complexity. The intersection network could be implemented in a single layer of sigma-pi units.

To see how this circuit implements intersection, consider again the simple case of a system with three meaningful vectors A, B, and C where we want to compute the intersection of $x = k_1A + k_2B + k_3C$ with $y = k_4A + k_5B + k_6C$. The vector x is loaded into register 1:, y is loaded into 2:, and the sum

$$A * \mathrm{P}_1(A) * \mathrm{P}_2(A) + B * \mathrm{P}_1(B) * \mathrm{P}_2(B) + C * \mathrm{P}_1(C) * \mathrm{P}_2(C)$$

is loaded into 4:. After passing the register contents through their respective permutations and multiplying the results, register 3: will contain

$$\begin{gathered}
\mathrm{P}_1(k_1A + k_2B + k_3C) * \mathrm{P}_2(k_4A + k_5B + k_6C) = \\
(k_1\mathrm{P}_1(A) + k_2\mathrm{P}_1(B) + k_3\mathrm{P}_1(C)) * (k_4\mathrm{P}_2(A) + k_5\mathrm{P}_2(B) + k_6\mathrm{P}_2(C)) = \\
k_1k_4\mathrm{P}_1(A) * \mathrm{P}_2(A) + k_2k_5\mathrm{P}_1(B) * \mathrm{P}_2(B) + k_3k_6\mathrm{P}_1(C) * \mathrm{P}_2(C) + \\
\mathit{noise}
\end{gathered}$$

where *noise* represents terms not corresponding to a meaningful component of the intersection. Multiplying this sum in register 3: by the contents of register 4: will then result in the desired intersection (plus additional noise), via the self-cancellation property:

$$\begin{gathered}
[k_1k_4\mathrm{P}_1(A) * \mathrm{P}_2(A) + k_2k_5\mathrm{P}_1(B) * \mathrm{P}_2(B) + k_3k_6\mathrm{P}_1(C) * \mathrm{P}_2(C)] * \\
[A * \mathrm{P}_1(A) * \mathrm{P}_2(A) + B * \mathrm{P}_1(B) * \mathrm{P}_2(B) + C * \mathrm{P}_1(C) * \mathrm{P}_2(C)] = \\
k_1k_4\mathrm{P}_1(A) * \mathrm{P}_2(A) * A * \mathrm{P}_1(A) * \mathrm{P}_2(A) + \\
k_1k_4\mathrm{P}_1(A) * \mathrm{P}_2(A) * B * \mathrm{P}_1(B) * \mathrm{P}_2(B) + \\
k_1k_4\mathrm{P}_1(A) * \mathrm{P}_2(A) * C * \mathrm{P}_1(C) * \mathrm{P}_2(C) + \\
k_2k_5\mathrm{P}_1(B) * \mathrm{P}_2(B) * A * \mathrm{P}_1(A) * \mathrm{P}_2(A) + \\
k_2k_5\mathrm{P}_1(B) * \mathrm{P}_2(B) * B * \mathrm{P}_1(B) * \mathrm{P}_2(B) + \\
k_2k_5\mathrm{P}_1(B) * \mathrm{P}_2(B) * C * \mathrm{P}_1(C) * \mathrm{P}_2(C) + \\
k_3k_6\mathrm{P}_1(C) * \mathrm{P}_2(C) * A * \mathrm{P}_1(A) * \mathrm{P}_2(A) + \\
k_3k_6\mathrm{P}_1(C) * \mathrm{P}_2(C) * B * \mathrm{P}_1(B) * \mathrm{P}_2(B) + \\
k_3k_6\mathrm{P}_1(C) * \mathrm{P}_2(C) * C * \mathrm{P}_1(C) * \mathrm{P}_2(C) = \\
k_1k_4A + \\
k_1k_4\mathrm{P}_1(A) * \mathrm{P}_2(A) * B * \mathrm{P}_1(B) * \mathrm{P}_2(B) + \\
k_1k_4\mathrm{P}_1(A) * \mathrm{P}_2(A) * C * \mathrm{P}_1(C) * \mathrm{P}_2(C) + \\
k_2k_5\mathrm{P}_1(B) * \mathrm{P}_2(B) * A * \mathrm{P}_1(A) * \mathrm{P}_2(A) + \\
k_2k_5B + \\
k_2k_5\mathrm{P}_1(B) * \mathrm{P}_2(B) * C * \mathrm{P}_1(C) * \mathrm{P}_2(C) + \\
k_3k_6\mathrm{P}_1(C) * \mathrm{P}_2(C) * A * \mathrm{P}_1(A) * \mathrm{P}_2(A) + \\
k_3k_6\mathrm{P}_1(C) * \mathrm{P}_2(C) * B * \mathrm{P}_1(B) * \mathrm{P}_2(B) + \\
k_3k_6C = \\
k_1k_4A + k_2k_5B + k_3k_6C + \mathit{noise}
\end{gathered}$$

Note that this apparently complex calculation is actually a single elementwise vector product operation. The circuit does not "see" the complexity of the vectors it operates on. The same holds true for the normalizing operator mentioned above: normalization is implemented as a scalar multiplier applied to the entire vector to keep the sum of the element activations approximately constant.

Experimental Results

As a proof-of-concept for our distributed lateral inhibition architecture, we ran several experimental trials using the circuit from Figure 2. We started with an initial vector $x_0 = 1/N \sum_{i=1}^{N} k_i X_i$, with $k_j = 1.02$ for one arbitrarily chosen j and $k_i = 1$ for $i \neq j$. We then iterated the operation $x_{t+1} = \mathit{normalize}(x_t \wedge x_t)$, where $\wedge$ is the intersection operator in Figure 2, and $\mathit{normalize}(x) = x / \max_l(|x_l|)$. (The initial conditions thus represent a temporary violation of the constraints given above for k_i that are immediately rectified by the normalizing operation.) We stopped iterating when the Euclidean distance between x_t and x_{t-1} fell below 0.01.

Figure 3 shows a typical result, for $N = 3$ a vector x of 2000 dimensions, and 100 permutations. The system quickly converges to an x in which a single X_i dominates. We have reproduced these results for larger values of N, using vectors

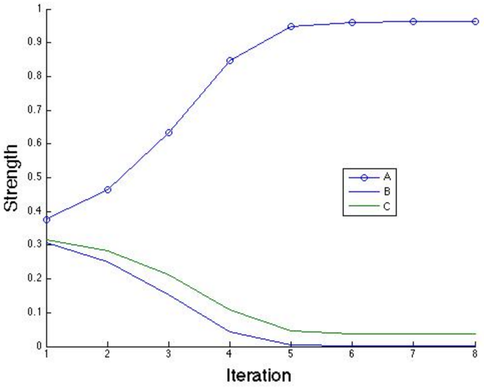


Figure 3: Winner-takes all in VSA implementation

with more realistically large dimensions and more permutations (connectivity). It is important to emphasize that *all representations and operations in this and the next experiment are fully distributed.* The figure was produced by serial extraction of the strengths of the three principal vectors of this system, but this was done only for purposes of illustration. There is nothing in the system that requires the intervention of a localist "homunculus" at any stage.

Application to Analogical Mapping

Analogical mapping has long been a focus of efforts in cognitive modeling. There are several successful connectionist cognitive models of analogy (Holyoak & Thagard, 1989; Hummel & Holyoak, 1997; Eliasmith & Thagard, 2001). These models vary in their theoretical emphases and the details of their connectionist implementations. However, they all share a problem in the scalability of the amount of computational resources or effort required to construct the connectionist mapping network. We contend that this is a consequence of using localist connectionist representations or using distributed representations in a localist manner.

To address this issue, we have recently developed a model that treats analogical mapping as a special case of graph isomorphism; that is, the solution of finding an optimal mapping between two structures (graphs) consisting of individuals (vertices) and their relations (edges). For example, in the simple graphs in Figure 4, the maximal isomorphism is {A=P, B=Q, C=R, D=S} or {A=P, B=Q, C=S, D=R}. Our model builds on the work of Pelillo (1999), who uses replicator dynamics (originally developed in evolutionary game theory) to solve the problem with a localist representation. In Pelillo's solution, iterated multiplication of a localist edge-consistency matrix *w* by a localist vertex-mapping vector *x* produces a localist "payoff" vector π expressing the quality of the solution. Elementwise multiplication of *x* with π produces an updated *x* representing an improved set of vertex mappings. This elementwise multiplication can be construed as a multiset intersection.

In our VSA implementation of this model, all entities (vertices, edges, and *w*, *x*, and π) are represented as high-dimensional MAP vectors. Vertex mappings in *x* are represented as the sums of the corresponding pairwise edge-mapping products (A*P + A*Q + ... + C*S), and the winner-takes-all intersection circuit of Figure 2 supports competition among mutually inconsistent mappings (C=R, D=S vs. C=S, D=R), without decomposing *x* into its constituent edge mappings. As shown in Figure 5, the VSA implementation can exhibit dynamic convergence to a solution in a way that is qualitatively similar to the localist implementation. Here, each curve corresponds to the level of support for a specific node mapping; *e.g.*, AP represents the support for the correspondence between nodes A and P. Notice that the components corresponding to the correct node mappings compete with and suppress the components corresponding to incorrect node mappings. As in the previous experiment, the convergence takes place without decomposition into localist components, the figure being a localist presentation for illustration only.

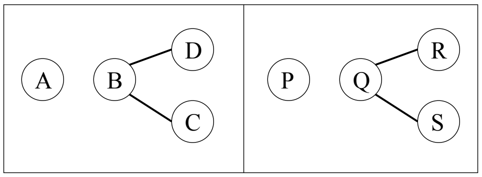


Figure 4: A simple graph isomorphism problem

Conclusion

We have presented a fully distributed connectionist architecture supporting lateral inhibition / winner-takes all competition. The architecture uses a neurally plausible permutation circuit to support a multiset intersection operation without decomposing the summed vector into its constituent items. This approach compares favorably with a localist approach when applied to the task of analogical mapping. Our results thus challenge the commonly-accepted view that lateral inhibition calls for localist representation. More profoundly, our model points toward a neural implementation where more complex representations do not require more complex or dynamically rewired hardware, a long-standing goal of connectionist cognitive modeling.

Acknowledgments

The authors thank two anonymous reviewers for the most helpful comments.

Software Download

Matlab code implementing the experiments described in this paper can be downloaded from **`tinyurl.com/lidemo`**.

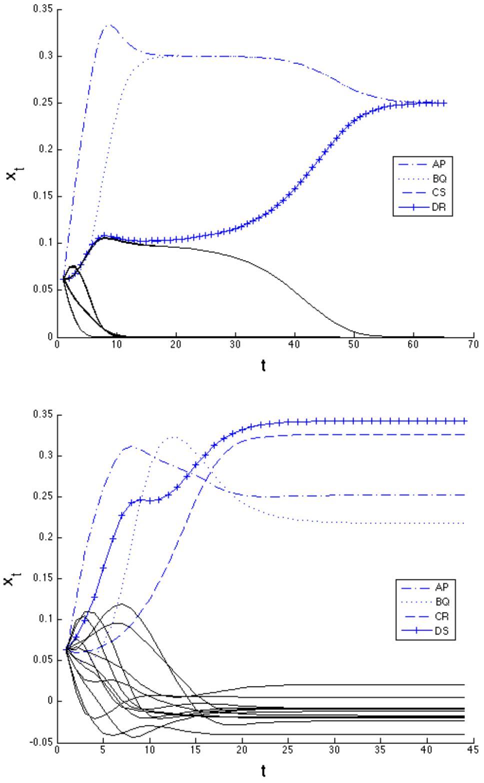


Figure 5: Convergence of localist (top) and VSA (bottom) implementations of graph isomorphism.

References

Eliasmith, C. (2005). Cognition with neurons: A large-scale, biologically realistic model of the Wason task. In G. Bara, L. Barsalou, & M. Bucciarelli (Eds.), *Proceedings of the 27 th annual meeting of the cognitive science society.*

Eliasmith, C., & Thagard, P. (2001). Integrating structure and meaning: A distributed model of analogical mapping. *Cognitive Science*, *25*, 245-286.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3-71.

Gayler, R. (1998). Multiplicative binding, representation operators, and analogy,. In K. Holyoak, D. Gentner, & B. Kokinov (Eds.), *Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences* (p. 405). Sofia, Bulgaria: New Bulgarian University.

Hammerton, J. A. (1998). Holistic computation: Reconstructing a muddled concept. *Connection Science*, *10*, 10-1.

Hecht-Nielsen, R. (1994). Context vectors: general purpose approximate meaning representations self-organized from raw data. In J. Zurada, R. M. II, & C. Robinson (Eds.), *Computational intelligence: Imitating life* (p. 4356). IEEE Press.

Holyoak, J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, *13*, 295-355.

Hummel, J., & Holyoak, K. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427-466.

Jackendoff, R. (2002). *Foundations of language: Brain, meaning, grammar, evolution.* Oxford University Press.

Kanerva, P. (1994). The binary spatter code for encoding concepts at many levels. In M. Marinaro & P. Morasso (Eds.), *Icann '94: Proceedings of international conference on artificial neural networks* (Vol. 1, p. 226-229). London: Springer-Verlag.

Kanerva, P. (in press). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*.

Levy, S. D. (to appear). Becoming recursive: Toward a computational neuroscience account of recursion in language and thought. In H. van der Hulst (Ed.), *Recursion in human language.* The Hague: Mouton de Gruyter.

Olson, A. C., & Humphreys, G. W. (1997). Connectionist models of neuropsychological disorders. *Trends in Cognitive Sciences*, *1*(6), 222 - 228.

Page, M. (2000). Connectionist modelling in psychology: A localist manifesto. *Behavioral and Brain Sciences*, *23*, 443512.

Pelillo, M. (1999). Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, *11*, 1933-1955.

Phaf, R. H., & Wolters, G. (2000). A competitive manifesto. *Behavioral and Brain Sciences*, *23*, 487-488.

Plate, T. A. (2003). *Holographic reduced representation: Distributed representation for cognitive science.* CSLI Publications.

Pollack, J. (1990). Recursive distributed representations. *Artifical Intelligence*, *36*, 77-105.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159-216.

Stewart, T., & Eliasmith, C. (forthcoming). Oxford handbook of compositionality. In W. Hinzen, E. Machery, & M. Werning (Eds.), *Compositionality and biologically plausible models.* Oxford: Oxford University Press.

van der Velde, F. (2006). Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, *29*, 1-72.

Flexible Spatial Language Behaviors: Developing a Neural Dynamic Theoretical Framework

John Lipinski (2johnlipinski@gmail.com)
Yulia Sandamirskaya (yulia.sandamirskaya@neuroinformatik.rub.de)
Gregor Schöner (gregor.schoener@neuroinformatik.rub.de)
Institut für Neuroninformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany

Abstract

To date, spatial language models have tended to overlook process-based accounts of building scene representations and their role in generating flexible spatial language behaviors. To address this theoretical gap, we implemented a model that combines spatial and color semantic terms with neurally-grounded scene representations. Tests of this model using real-world camera input support its viability as a theoretical framework for behaviorally flexible spatial language.

Keywords: dynamical systems; neural networks; spatial cognition; spatial language.

Introduction

Spatial language is an incredibly flexible tool whose capabilities range from generating and comprehending directions (Tom & Denis, 2004) to facilitating coordinated action (Bangerter, 2004). Yet, despite this broad behavioral scope, implemented spatial language models which seek to uncover processes underlying basic spatial communication (e.g. object location description) have tended to focus on a limited range of behaviors, namely relational judgment tasks. These models have successfully accounted for a complex array of empirical data including the influence of landmark shape (Regier & Carlson, 2001) and functional object features (Coventry et al., 2005). The neural processing aspects underlying these accounts, however, remain underdeveloped. Consequently, a number of critical questions that bear directly on spatial language and its linkage to supporting sensory-motor processes have gone unaddressed. For example, how does a neural scene representation evolve on the basis of sensory information? How might complex higher-level behaviors like spatial language emerge from these lower-level dynamic processes? How are the time courses of spatial language behaviors structured by their roots in scene representations?

Behavioral flexibility in the spatial language system becomes a central issue once one addresses the neural processes that link spatial language to the sensory-motor system. Fundamentally, we do not yet understand how the sensory-motor foundations of scene representations and spatial language work to support the broad array of spatial language behaviors. The absence of process-based accounts for the generation of spatial scene representations and the behaviors derived from these representations is a significant barrier to developing a more comprehensive, integrative spatial language model.

As a step to overcoming this barrier, we were led to consider three elements underlying behavioral flexibility in spatial language. First, the spatial language system uses both spatial and non-spatial characteristics. Second, it integrates the graded sensory-motor representations with symbolic, linguistic terms. Finally, the spatial language system combines these numerous elements continuously in time according to the specific behavioral context.

To develop a behaviorally flexible theoretical framework for spatial language that satifies these constraints, one needs a representational language that links to both the sensory-motor and linguistic worlds. The Dynamic Field Theory (Erlhagen & Schöner, 2002), a neuronally based theoretical language emphasizing attractor states and their instabilities, is one viable approach. Recent applications of the DFT have extended beyond spatial working memory development (Spencer, Simmering, Schutte, & Schöner, 2007) to include a theoretically generative account of signature landmark effects in spatial language (Lipinski, Spencer, & Samuelson, in press). Critically, this latter work integrated a connectionist-style localist spatial term network into the model. This suggests that the DFT can provide the requisite, integrative representational language.

The present work incorporates this hybrid approach to implement a new model integrating spatial language semantics with real-world visual input. Our goal is to qualitatively test the model's core functionality and, thus, its viability as an initial theoretical framework for flexible spatial language behaviors. To rigorously test our model, we implement it on a robotic platform continously linked to real-world visual images of everyday items on a tabletop workspace. Our model extracts the categorical, cognitive information from the low-level sensory input through the system dynamics, not through neurally ungrounded preprocessing of the visual input. Models which do not directly link cognitive behavior to lower-level perceptual dynamics risk side-stepping this difficult issue. Our demonstrations specifically combine visual space, a selected subset of basic English spatial semantic terms, and color. These demonstrations serve as an initial proof of concept that takes an early step towards modeling more complex, natural spatial language behaviors.

Modeling neurons and dynamical neural fields

This section briefly reviews the mathematics of our model (see also (Erlhagen & Schöner, 2002)).

Dynamical fields

The dynamical neural fields are mathematical models first used to describe cortical and subcortical neural activation dynamics (Amari, 1977). The dynamic field equation Eq. (1) is a differential equation describing the evolution of activation u defined over a neural variable(s) $\mathbf{x}$. These neural variables represent continuous perceptual (e.g. color) or behavioral (e.g. reaching amplitude) dimensions of interest that can be naturally defined along a continuous metric.

$$\begin{aligned}\tau\dot{u}(\mathbf{x},t) = -u(\mathbf{x},t) + h + \int f(u(\mathbf{x}',t))\omega(\Delta\mathbf{x})d\mathbf{x}' + \\ + I(\mathbf{x},t)\end{aligned} \tag{1}$$

Here, $h < 0$ is the resting level of the field; the sigmoid non-linearity $f(u) = 1/(1+e^{-\beta u})$ determines the field's output at suprathreshold sites with $f(u) > 0$. The field is quiescent at subthreshold sites with $f(u) < 0$. The homogeneous interaction kernel $\omega(\Delta x) = c_{exc}e^{\frac{-(\Delta x)^2}{2\sigma^2}} - c_{inh}$ depends only on the distance between the interacting sites $\Delta x = \mathbf{x} - \mathbf{x}'$. This interaction kernel is a Bell-shaped (Gaussian), local excitation/global inhibition function. The short-range excitation is of amplitude c_{exc} and spread σ. The inhibition is global, as we are not interested in multipeak solutions here, and has an amplitude c_{inh}. $I(\mathbf{x},t)$ is the summed external input to the field; τ is the time constant.

If a localized input activates the neural field at a certain location, the interaction pattern ω stabilizes a localized "peak", or "bump" solution of the field's dynamics. These activation peaks represent the particular value of the neural variable coded by the field and thus provide the representational units in the DFT (Spencer & Schöner, 2003).

In our model, all entities having "field" in their name evolve according to Eq. (1), where $\mathbf{x}$ is a vector representing the two-dimensional visual space in Cartesian coordinates. The links between the fields are realized via the input term $I(\mathbf{x},t)$, where only sites with $f(u) > 0$ propagate activation to other fields or neurons.

Discrete nodes

The discrete (localist) neural nodes in the model representing spatial and color semantic terms can be flexibly used for either user input or response output. Their activation evolves according to the dynamic equation (2).

$$\tau_d\dot{d}(t) = -d(t) + h_d + f(d(t)) + I(t). \tag{2}$$

Here, d is the activity level of a node; the sigmoidal nonlinearity term $f(d)$ shapes the self-excitatory connection for each discrete node and provides for self-stabilizing activation. The negative resting level is defined by h_d. The $I(t)$ term represents the sum of all external inputs into the given node. This summed input is determined by the input coming from the connected neural field, the user interface specifying the language input, and the competitive, inhibitory inputs from the other discrete nodes defined for that same feature group (color or space); τ is the time constant of the dynamics.

The spatial language framework

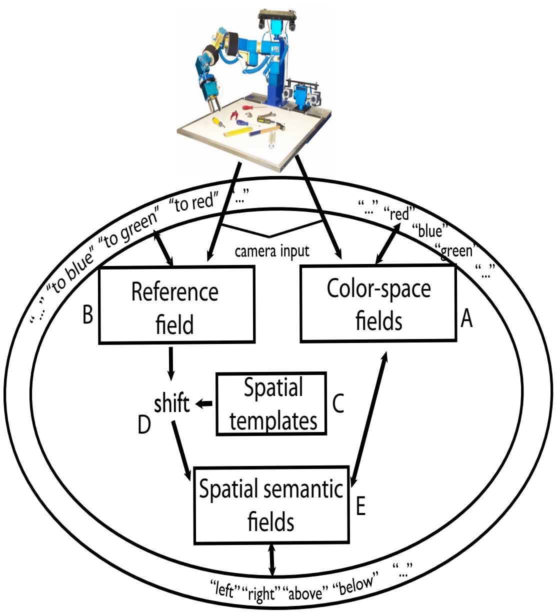


Figure 1: Overview of the architecture

This section outlines the overall structure (see Fig. 1) of our integrative model and explains how it operates in two scenarios fundamental to spatial language: describing *where* an object is (Demonstration 1) and describing *which* object is in a specified spatial relation (Demonstration 2).

Color-space fields

The color-space fields (Fig. 1A) are an array of several dynamical fields representing the visual scene. Each of the fields is sensitive to a hue range which corresponds to a basic color. The resolution of color was low in the presented examples because only a few colors were needed to represent the used objects. In principle, the color (hue) is a continuous variable and can be resolved more finely. The stack of color-space fields is therefore a three-dimensional dynamic field that represents colors and locations on the sensor surface. The camera provides visual input to the color-space field, which is below the activation threshold before the task is defined. The field is thus quiescent to this point.

Once the language input specifying the *color* of the object activates the respective color-term node, however, the resting levels of all sites of the corresponding color-space field are raised homogeneously. Because the color-space fields receive localized camera input, this uniform activation increase is summed with that input to enable the development of an instability and, ultimately, the formation of a single-peak solution. This peak is centered over the position of the object with that specified color.

The *spatial* language input also influences the color-space field's dynamics through the aligned spatial semantic fields (see below).

Reference field

The reference field (Fig. 1B) is a spatially-tuned dynamic field which also receives visual input (Fig 1B). When the user specifies the reference object color, the corresponding ”reference-color” node becomes active and specifies the color in the camera image that provides input into the reference field. A peak of activation in the reference field evolves at the location of the reference object. The reference field continuously tracks the position of the reference object. Its dynamics also filters out irrelevant inputs and camera noise and thus stabilizes the reference object representation. Having a stable, but updatable reference object representation allows the spatial semantics to be continuously aligned with the visual scene.

Spatial semantic templates

The spatial semantic templates (Fig. 1C) are represented as a set of synaptic weights that connect spatial terms to an abstract, ”retinotopic” space. The particular functions defining ”left”, ”right”, ”below”, and ”above” here were two-dimensional Gaussians in polar coordinates and are based on a neurally-inspired approach to English spatial semantic representation (O'Keefe, 2003). When viewed in Cartesian coordinates, they take on a tear-drop shape for these terms.

Shift

The shift mechanism (Fig. 1D) aligns these retinotopically defined spatial semantics with the current task space. The shift is done by convolving the ”egocentric” weight matrices with the outcome of the reference field. Because the single reference object is represented as a localized activation peak in the reference field, the convolution simply centers the semantics over the reference object. The spatial terms thus become defined relative to the specified reference object location (for related method see (Pouget & Sejnowski, 1995)).

Aligned spatial semantic fields

The aligned spatial semantic fields (Fig. 1E) are arrays of dynamical neurons with weak lateral interaction. They receive input from the spatial alignment or ”shift” mechanism which maps the spatial semantics onto the current scene by ”shifting” the semantic representation of the spatial terms to the reference object position. The aligned spatial semantic fields integrate the spatial semantic input with the summed outcome of the color-space fields and interact reciprocally with the spatial-term nodes. Thus, a positive activation in an aligned spatial semantic field increases the activation of the associated spatial-term node and vice versa.

Demonstrations

We here detail two exemplar demonstrations (from a set of thirty conducted) which address two behaviors fundamental to spatial language. In the presented scenarios, three objects were placed in front of the robot: a green stack of blocks, a yellow plastic apple, and a blue tube of sunscreen. The visual input was formed from the camera image and sent to the reference and color-space fields. The color-space field input was formed by extracting hue value (”color”) for each pixel in the image and assigning that pixel's intensity value to the corresponding location in the matching color-space field. The input for the reference field was formed in an analogous fashion according to the user-specified reference object color. When the objects are present in the camera image, the reference and color-space fields receive localized inputs, corresponding to the three objects in view (marked with arrows, see Fig. 2 and Fig. 3). This was the state of the system before the particular task was set.

In Demonstration 1 we ask ”Where is the yellow object relative to the green one?” and the robot must select the correct descriptive spatial term. In Demonstration 2 we ask ”Which object is to the right of the yellow one?” and the robot must select the color term that describes the target object. Both examples were performed with exactly the same visual scene and parameter set. Thus, the only difference for the system was the user-specified task input. If our model functions properly, the interactive dynamics should select the correct spatial or color term according to the task details.

Due to the graded representation of space and color in the neural fields, being able to solve these two tasks means accessing hundreds of scenarios with multiple objects and object positions in the image. More fundamentally, these different tasks both require the integration of visual and symbolic input as well as the autonomous selection of a descriptive spatial term. Such integration and decision processes are a core capacity of the human spatial language system and underlie the full range of real-world spatial language behaviors. Accounting for these core processes in different tasks in a single, neurally-grounded model provides a strong foundation for scaling up to more complex spatial language scenarios.

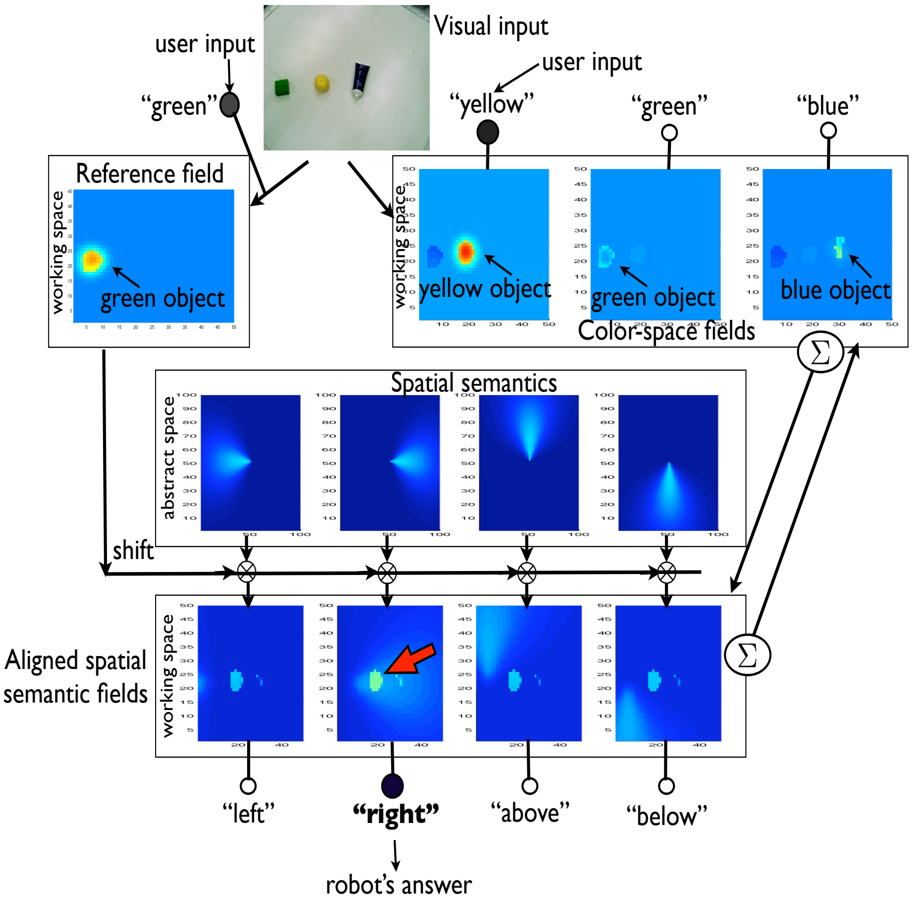


Figure 2: Demonstration 1 activations just before answering "Where".

Demonstration 1: Describing "Where"

Demonstration 1 asks "Where is the yellow object relative to the green one?" To respond correctly, the robot must select "Right". Fig. 2 shows the neural fields' activation just before the answer is given. The task input first activates two discrete neurons, one representing "green" for the user-specified reference object color and the other "yellow" for the user-specified object color (see user inputs, top Fig. 2). The reference object specification "green" leads to the propagation of the green camera input into the reference field, creating an activation bump in the reference field at the location of the green item (see Reference field, Fig. 2). The specification of the target color "yellow" increases the activation for the "yellow" node linked to the "yellow" color-space field (see yellow activation time course line, top Fig. 4a), which raises the resting level of the associated "yellow" color-space field. This uniform activation boost coupled with the camera input from the yellow object induces an activation peak in the field (see "yellow" Color-space field, Fig. 2).

This localized target object activation is then transfered to the aligned semantic fields. In addition to receiving this target-specific input, the aligned semantic fields also receive input from spatial term semantic nodes. Critically, these semantic profiles are shifted to align with the reference object position. In the current case, the yellow target object activation therefore overlaps with the aligned "right" semantic field (see red arrow in the "right" Aligned spatial semantic field, Fig. 2). This overlap ultimately drives the activation and selection of the "right" node (see spatial-term neuron activation time course, bottom Fig. 4a).

Demonstration 2: Describing "Which"

Demonstration 2 asks "Which object is to the right of the yellow one?". To respond correctly, the robot must select "Blue". As indicated in Fig. 3, the task input first activates two discrete nodes, one representing the reference object color "yellow" and the other representing "right".

The reference object specification "yellow" creates an activation bump in the reference field location matching that of the yellow item (see Reference field, Fig. 3). The specification of "right", in its turn, increases the activation for

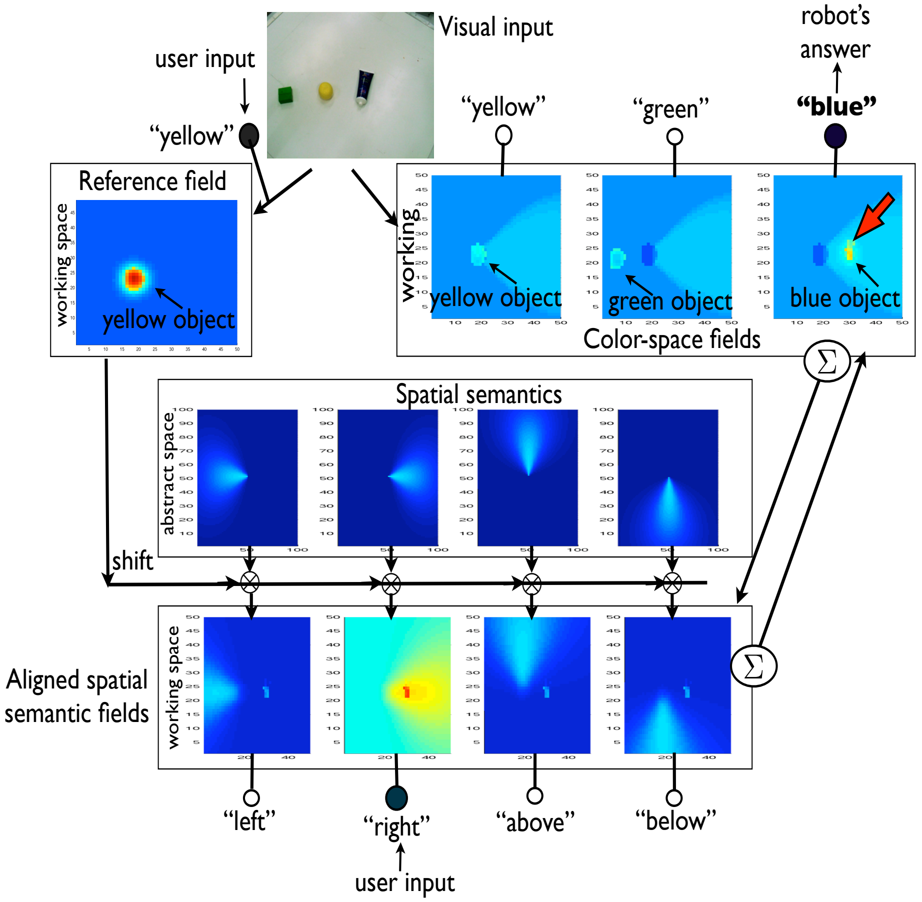
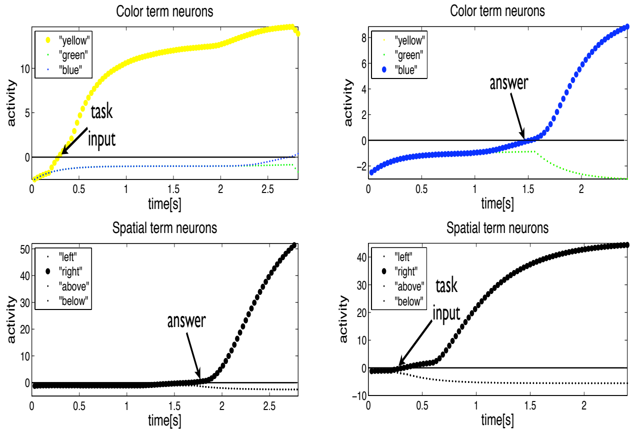


Figure 3: Demonstration 2 activations just before answering ”Which”.

that spatial-term node (see activation time course, bottom Fig. 4b), creating a homogeneous activation boost to the ”right” semantic field. This activation boost creates a positive activation in the field to the right of the yellow reference object (see ”right” Aligned spatial semantic field, Fig. 3). This spatially-specific activation is then input into the color-space fields and subsequently raises activation at all those color-space field locations to the right of the reference object (see lighter-blue Color-space fields' regions, Fig. 3). This region overlaps with the localized input of the blue object in the ”blue” color-space field and an activation peak develops in that field (see red arrow in the ”blue” Color-space field, Fig. 3). This increases the activation of the associated ”blue” color-term node, triggering selection of the correct answer, ”blue” (see color-term node's activation profile, top Fig. 4b).



(a) Demo 1: ”Where”

(b) Demo 2: ”Which”

Figure 4: Activation time courses for the spatial and color terms neurons

Discussion

Together, these demonstrations reveal the model's ability to localize the specified target object in the visual scene and to extract the required spatial or non-spatial target information. These different behaviors emerged from the autonomous dynamics integrating the low-level camera input and the categorical user input and are thus truly context-dependent. In assessing this framework it is also important to note that precisely the same parameter setting was used in all tasks; only the context input changed. Thus, the behaviors are autonomously structured simply by the symbolic and visual input. Even with our initially limited range of spatial and color terms, the framework can be immediately applied to a broad range of real-world objects and locations without modification. This novel system therefore provides a contextually adaptive framework for the flexible application of spatial semantics. More fundamentally, because of its focus on integrative dynamic processes modelled in accordance with neural principles, it also provides a foundation for modeling more complex human spatial language behaviors.

References

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, *27*, 77-87.

Bangerter, A. (2004). Using pointing and describing to achieve joint focus of attention. *Psychological Science*, *15*, 415-419.

Coventry, K., Cangelosi, A., Rajapakse, R., Bacon, A., Newstead, S., Joyce, D., et al. (2005). Spatial prepositions and vague quantifiers: Implementing the functional geometric framework. In C. Freksa (Ed.), *Spatial cognition iv* (Vol. LNAI 3343, p. 98-110). Heidelberg: Springer-Verlag.

Erlhagen, W., & Schöner, G. (2002). Dynamic field theory of movement preparation. *Psychological Review*, *109*, 545-572.

Lipinski, J., Spencer, J. P., & Samuelson, L. (in press). It's in the eye of the beholder: Spatial language and spatial memory use the same perceptual reference frames. In L. B. Smith, M. Gasser, & K. Mix (Eds.), *The spatial foundations of language.* Oxford University Press.

O'Keefe, J. (2003). *Vector grammar, places, and the functional role of the spatial prepositions in english* (E. van der Zee & J. Slack, Eds.). Oxford: Oxford University Press.

Pouget, A., & Sejnowski, T. J. (1995). Spatial representations in the parietal cortex may use basis functions. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems 7.* MIT Pres, Cambridge MA.

Regier, T., & Carlson, L. (2001). Grounding spatial language in perception: An empirical and computational investigation. *Journal of Experimental Psychology: General*, *130*(2), 273-298.

Spencer, J. P., & Schöner, G. (2003). Bridging the representational gap in the dynamical systems approach to development. *Developmental Science*, *6*, 392-412.

Spencer, J. P., Simmering, V. R., Schutte, A. R., & Schöner, G. (2007). What does theoretical neuroscience have to offer the study of behavioral development? insights from a dynamic field theory of spatial cognition. In J. Plumert & J. P. Spencer (Eds.), *The emerging spatial mind.* Oxford University Press.

Tom, A., & Denis, M. (2004). Language and spatial cognition: Comparing the roles of landmarks and street names in route instructions. *Applied Cognitive Psychology*, *18*, 1213-1230.

An Account of Model Inspiration, Integration, and Sub-task Validation

Christopher W. Myers
cmyers@cerici.org

Cognitive Engineering Research Institute
5810 S. Sossaman Rd. Ste. 106
Mesa, AZ 85212 USA

Abstract

The promise of reuse is a motivator for, and benefit from, developing cognitive models. Another benefit is the integration of previously developed models into a single model capable of making predictions across different tasks than either of the contributing models could make alone. In the current paper, I explicate the development of a model through the integration of, and inspiration from, two previously published models. The composite was developed for a context different from the constituent models' original contexts, and demonstrates a success of inspiration and integration.

Keywords: model reuse, integration, synthetic teammate

Introduction

The promise of model reuse is a boon to, and motivator for, developing models. A related benefit is to integrate different models into a composite capable of making predictions in different contexts than the contributing models could make alone. To take advantage of these potential benefits, developers of large-scale, complex models must seek out published models for integration rather than *reinventing the wheel*. This paper provides an account of the integration of, and inspiration from, two previously reported models of distinct cognitive processes.

The context for model inspiration and integration is the development of a synthetic teammate (Ball et al., 2009). The constituent models provided portions of the synthetic teammate component responsible for interacting with its task environment, the *task behavior component*.

In the remaining sections of the introduction, I first provide background on the synthetic teammate project. Second, I describe the task and goals specific to the synthetic teammate's task behavior component. Third, I provide results from a task analysis on the goals critical to the task behavior component.

The Synthetic Teammate Project

The Cognitive Engineering Research Institute and the Performance & Learning Models team at the Air Force Research Laboratory are collaborating to develop a synthetic agent capable of coordinating with human teammates to complete an unmanned aerial vehicle (UAV) reconnaissance task. The far-term goal of the project is to reduce the number of human operators in team trainers while maintaining training effectiveness. The near-term goal of this project is to develop a cognitively plausible synthetic teammate within the ACT-R 6 cognitive architecture (Anderson, 2007). Achieving the near-term goal will facilitate accomplishing far-term goals, through the identification of cognitive capacities necessary for operating as a teammate (e.g., memory, language, etc.), and demonstrate how to integrate relevant cognitive capacities.

The synthetic teammate is being developed to operate within a UAV Synthetic Task Environment (UAV-STE; see Cooke & Shope, 2005) used to study teams for the better part of the past decade. In the UAV-STE, teammates coordinate to successfully complete a reconnaissance task. The synthetic teammate has been developed to operate as the UAV Air Vehicle Operator (AVO), and to interact with a photographer that takes pictures of ground targets, or *waypoints*, and a mission planner responsible for planning the UAV's route. Communication among teammates occurs through a text-based instant messaging system.

Four cognitive components have been identified as the basis of the synthetic teammate: 1) language comprehension, 2) language generation and dialog management, 3) situation assessment, and 4) task behavior. The current paper is focused on goals associated with the task behavior component (see Ball et al., 2009, for details on the other components).

Task Goals for the Task Behavior Component

To fly the UAV, the AVO must complete six goals: 1) set the airspeed, 2) altitude, 3) course, 4) waypoint, and 5) send and 6) receive text messages. Of these six, the solution for the first four is covered in the current paper. A typing model associated with the last two goals, sending and receiving messages, is currently being integrated (John, 1996).

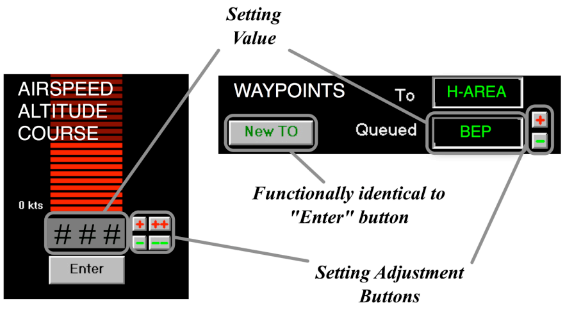


Figure 1. The left box is an example of the interface used to enter the airspeed, altitude, and course. The right box is used for setting waypoints.

The UAV-STE was designed to simulate a team task; consequently the user interface was not designed as a high fidelity representation of any existing UAV system in use by the military. To maneuver the UAV from one location to another, the AVO uses a point-and-click interface to enter settings (see Figure 1). To set the waypoint, the user toggles through a list of 109 alphabetically organized waypoints by pressing the setting adjustment buttons (see Figure 1). Each time an adjustment button is pressed, a new waypoint value is queued (e.g., BEP in Figure 1). The waypoint list operates as a continuous loop (i.e., *A* comes after *Z*). When the user has queued the next waypoint to visit, she presses the "New TO" button, changing "H-AREA" to "BEP" in Figure 1.

There are three *flight parameters* (altitude, course, and airspeed). Each flight parameter has a separate user interface, though they are identical (see Figure 1). To set the flight parameters, the AVO adjusts the setting value by using the small (+ | –) and large (++ | – –) setting adjustment buttons. These buttons have different increments and decrements depending on the setting (see Table 1). Similar to the waypoint list, course values are a continuous loop, returning to 1° after 360°. The airspeed and altitude values are infinite number lines, beginning at 0 and ending at infinity. When the desired setting value is reached, the user presses the "Enter" button to complete the setting goal.

Table 1. Setting adjustment buttons for each task setting goal. The waypoint buttons either increment to the next (+), or decrement (–) to the previous waypoint in an alphabetical list. The other button increments increase or decrease setting values, accordingly.

Task Goals	Large ++ | – –	Small + | –
Airspeed	20 | –20	2 | –2
Altitude	1000 | –1000	100 | –100
Course	10 | –10	1 | –1
Waypoint	Not applicable	1 | –1

There are five differences between setting a waypoint and setting the flight parameters. First, the adjustment buttons for setting the flight parameters have small and large adjustments, whereas there are only small adjustments for setting the waypoint (see Figure 1 and Table 1). Second, there is an "Enter" button for setting the flight parameters and a "New TO" button for setting the waypoint. Although these buttons have different names, there functions are identical. Third, the values of flight parameters are integers, whereas waypoints are strings of numbers and letters (e.g., WP8, BEP). The fourth difference is the addition of the queued value for setting the waypoint, and the fifth and final difference is the spatial arrangements of the user interfaces.

Although there are interface differences between setting flight parameters and waypoints, there is considerable overlap of methods for setting altitudes, courses, airspeeds, and waypoints. In the following section I provide results from a task analysis of the four goals.

Task Analysis Results

A hierarchical GOMS (i.e., goals, operators, methods, and selection rules) analysis was conducted on setting waypoints and flight parameters. The purpose of the analysis was to reveal commonalities and differences between the goals.

The task analysis revealed a consistent three-step subgoal structure across each of the four goals, composed of 1) *obtaining* the desired setting value, 2) c*omparing* the desired setting value against the current value, and 3) *Changing* the current setting value to the desired value. The following methods <*m*> and selection rules <*sr*> are identical across the four goals:

```
Obtain subgoal <sr>:
    Either  Retrieve the desired information from memory
    Or      Request the information from a teammate.
Compare subgoal <m>:
    1. <m>Visually encode one of the adjustment buttons
    2. <m>Move mouse to, and click on, button
    [system-event]:= setting value appears
    3. <m> Visually encode setting value
    4. <sr> IF button adjustment values are unknown,
       THEN retrieve them from memory
    5. <sr> Given the current setting value, desired
       setting value, and adjustment button values, select
       adjustment button
Change subgoal <m>:
    1. <sr> IF mouse is at the selected adjustment button,
       THEN goto <m> 4; ELSE continue
    2. <m> Visually encode button
    3. <m> Move mouse to button
    4. <m> Click mouse
    [system-event]:= setting value changes
    5. <sr> IF not attending to setting value, THEN
       visually encode setting value; ELSE continue
    6. <sr> IF the current setting equals the desired
       setting, THEN visually encode "Enter"/"New TO"
       button and goto change subgoal <m>7; ELSE IF
       large adjustment clicked, THEN goto compare
       subgoal, <sr>5; ELSE goto change subgoal,
       <m>4.
    7. <m> Click mouse–return with goal accomplished.
    [system-event]:= setting value disappears
```

Although setting flight parameters and waypoints follow the same subgoal structure, methods for completing steps in the subgoal methods presented above diverge. The divergence results from different value types between flight parameters and waypoints and the absence of large setting adjustment buttons for setting waypoints. These differences specifically affect methods for completing <*sr*>5 of the *compare* subgoal. In the following section, candidate models for setting the flight parameters and waypoints are selected from the cognitive modeling literature.

Candidate Models

As the science of developing quantitative process models of cognitive activities matures, many models become available with which to take whole cloth or draw inspiration from when tackling large, complex models that must be capable of completing many different tasks. Rather than possibly reinventing the wheel, published models were sought as candidates for integration into the task behavior component of the synthetic teammate. To be a candidate, models had to be compatible with the subgoal methods and selection rules detailed above. The current section covers a strategy selection model (Lovett, 1998) with implications for setting flight parameters, and a letter recall and comparison model (Klahr et al., 1983) with implications for setting waypoints.

Strategy Selection

Lovett (1998) demonstrated that ACT-R's choice mechanism can account for changes in strategy selection with experience from the task environment. Lovett identified two strategies for obtaining a solution in a spatial problem-solving task (i.e., the building-sticks task): *overshoot* or *undershoot*. Generally, the overshoot strategy results in passing the desired state, and then backtracking to it. The undershoot strategy incrementally approaches the desired state without passing it. Strategy selection was based on a strategy's likelihood of success within the environment, such as sets of problems where the overshoot strategy produced a solution a majority of the time and vice versa.

In the building-sticks task, the choice of which strategy to use was not obvious, requiring experience to determine which strategy was most successful. Lovett's model used the production utility mechanism in ACT-R 5 to learn which of the two strategies was best suited for different problem sets. With experience, the model learned to choose a strategy on a proportion of trials that was similar to humans.

Lovett's approach to strategy selection is perfectly suited for selecting between possible strategies for setting flight parameters for two reasons. First, Lovett's model was originally developed in an earlier version of ACT-R. Second, her undershoot and overshoot strategies are similar to strategies that can be brought to bear on setting flight parameters.

When setting a flight parameter, the AVO has four possible adjustment buttons to choose from. From the four options come two strategies: *difference reduction* and *meandering*. The difference reduction strategy involves moving from the current setting to the desired setting, reducing the difference between the two values at each step, and can be achieved *efficiently* or *inefficiently*.

The efficient difference reduction strategy comes as close as possible to the desired setting using the large adjustment buttons, then switching to the small adjustment buttons to reach the desired setting. Indeed, Lovett's overshoot and undershoot strategies are efficient difference reduction strategies.

The inefficient difference reduction strategy involves only using the small adjustment buttons. This strategy will succeed, but in many cases take substantially longer to complete than the efficient difference reduction strategy.

Finally, the meandering strategy is a mix of difference reduction and periodic interventions to randomly select and use a different adjustment button. This strategy will eventually select the desired setting value, but could take months. Hence, only the two difference reduction strategies are considered further.

The efficient difference reduction strategy can be developed as independent undershoot and overshoot strategies, similar to those described by Lovett. Because the structure of the flight parameter setting environment does not contain any bias leading to differential success between an efficient undershoot difference reduction strategy or an efficient overshoot difference reduction strategy, there is little use in developing models of each strategy and letting ACT-R's choice mechanism demonstrate equivalency. Furthermore, the inefficient difference reduction strategy is a "straw man" strategy–participants will arguably use the large increment buttons simply because of their availability.

Letter Recall and Comparison

When setting a waypoint, the AVO can either advance (+) or retreat (–) through the list of waypoints one waypoint at a time. I assumed that participants come to the task with extensive knowledge and experience in the English alphabet. I also assumed that the choice to advance or retreat through waypoints results from bringing the alphabet knowledge to bear on the waypoint setting goal, and looked to Klahr et al. (1983) as a candidate representation of the English alphabet for a model of letter comparison.

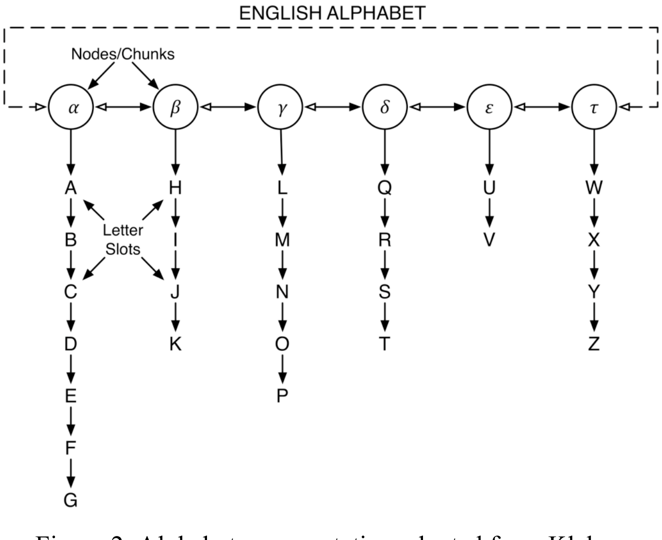


Figure 2. Alphabet representation adapted from Klahr, Chase, and Lovelace (1983). Dashed lines and open arrows represent capabilities added to their model.

In the Klahr et al. (1983) model of letter retrieval and comparison, letters were stored as hierarchical subgroups in a link-node structure (e.g., α to τ in Figure 2). Letters within a node (e.g., D in node α) can only be reached through node *entry points*. Entry points for each node are the first letter of the node (e.g., A for α, H for β, etc., see Figure 2). Node contents are based on empirical evidence of entry point consistency with phrasing in "Twinkle, Twinkle Little Star," used to teach the alphabet (Klahr et al., 1983).

Klahr et al.'s letter retrieval model is a serial, self-terminating search across and within alphabet nodes. Letters and nodes were linked only to their successors. Thus, to backtrack through nodes the previous node must be maintained in working memory.

Klahr et al. validated their model with response time data collected from human participants that were shown letters of the alphabet and asked to respond with the name of the letter that either occurs before or after the probe letter. However, determining whether to advance or retreat through the waypoint list in the UAV-STE is quite different. Rather than responding with an adjacent letter, the model must determine whether the desired waypoint (e.g., BEO) occurs before or after the queued waypoint (BEP in Figure 1). This requires determining if a letter occurs before or after another letter in the alphabet, and these comparisons can occur between and within letter nodes. Even so, the Klahr et al. (1983) model is a good candidate for representing the English alphabet. In the following section I cover the development and integration of the candidate models within ACT-R.

Development & Integration in ACT-R

ACT-R is a computational cognitive architecture for developing cognitive models (Anderson, 2007). In ACT-R, cognition revolves around the interaction between a central production system and several modules. There are modules for vision, motor capabilities, memory, storing the model's intentions for completing the task (i.e., the control state), information retrieved from memory, and a module for storing the mental representation of the task at hand (i.e., the problem state). Each module contains one or more buffers that can store one piece of information, or *chunk*, at a time. Modules are capable of massively parallel computation to obtain chunks. For example, the memory module can retrieve a single chunk from thousands of others and place the chunk into the module's buffer. Module contents are used to guide processing in the central production system.

The central production system is a set of state-action rules that are matched to buffer contents and act on the buffers by removing information from them or adding information to them. Only a single production rule can proceed at a time, and each production rule takes at least 50 ms to complete. The production system acts as a serial bottleneck, as all information passed between the buffers, and interactions with the environment, must go through it.

Developing A Flight Parameter Setting Strategy

The previous section covering Lovett's model of strategy selection revealed that there is not a differential benefit between overshoot and undershoot strategies for setting flight parameters. Not only is there no differential benefit, there are few alternative strategies that would compete in setting flight parameters. Consequently, only the efficient undershoot strategy was developed for setting the flight parameters.

In the ACT-R productions that instantiate <*sr*>5 of the *compare* subgoal, a function was called from a production's action side that selects the appropriate adjustment button given button adjustments for the current flight parameter (e.g., altitude, airspeed, and course) and the current and desired setting values. Button selection was implemented in this fashion to avoid the need of integrating a representation of the number line, integrating models of addition and subtraction, and integrating a model of choosing the appropriate adjustment button based on the button increments and the difference between the desired and current setting values. Hence, the efficient undershoot strategy was perfectly executed by the model when setting the flight parameters. However, the model was not provided knowledge that course values looped back to 1° after passing 360°.

Developing a Waypoint Setting Model

The waypoint adjustment button selection process utilized Klahr et al.'s (1983) model of letter retrieval and comparison. The English alphabet was divided into six *alpha-chunks* that contained letters, instantiating Klahr et al.'s alphabet nodes (see Figure 2). Alpha-chunks were stored in ACT-R's declarative memory, and were based on the Klahr et al. (1983) alphabet division. In addition to letters, the chunk's name and the name of the subsequent alpha-chunk (i.e., the next-node-name slot) were also stored in alpha-chunks. Different from Klahr et al., chunk slots for the chunk name that comes prior to the current chunk in the alphabet (i.e., the previous-node-name slot) and the absolute position of the alphabet chunk in the alphabet (i.e., the position slot with values ranging from 1 to 6) were added to alpha-chunks. The values in the previous-node-name and the next-node-name slots were strings and thus have no effect on memory retrieval in ACT-R.

A two-step process was developed to complete <*sr*>5 of the *compare* subgoal. The process began by comparing the first letter of each waypoint name. If they were equal, subsequent letters were compared until two were different (e.g., O and P from the desired waypoint *BEO* and the queued waypoint *BEP*). At this point the second step began.

The second step began with retrieving alpha-chunks for each of the different letters for comparison (in our example letters O and P). When retrieving alpha-chunks, activation was spread from letters residing in the goal buffer. Thus, alpha-chunks were retrieved independently, without the need to serially traverse the alpha-chunks/nodes until the desired alpha-chunk was reached. This non-serial retrieval of alpha-chunks differs from the Klahr, et al. model, and allows traversing the alphabet nodes in either direction (see open and closed arrows between nodes in Figure 2).

When different alpha-chunks were retrieved, letter comparisons were made using a combination of the previous-node-name, next-node-name, and position slots. However, when retrievals returned the same alpha-chunk, the model had to serially search through the letter slots of the retrieved alpha-chunk until one of the letters was found. To instantiate serial search through slots in the alpha-chunks, *s* x *o* productions were developed, where *s* is the greatest number of letter slots in the alpha-chunk containing the most letters minus one, and *o* is the number of possible outcomes based on comparing two letters. The value for *s* is reduced because if the penultimate slot is reached without finding either of the letters, than the wrong alphabet chunk has been retrieved, searching the last slot becomes useless, and a new retrieval is issued.

The α alpha-chunk had the greatest number of letter slots (i.e., 7), and there were three possible outcomes–the letter from the desired waypoint was reached first in an alpha-chunk, a letter from the queued waypoint was reached first, or the currently checked slot did not contain either letter. Consequently, 6 x 3 = 18 productions were developed to serially search through letter slots of retrieved alphabet chunks. These productions mimicked procedural expertise of iterating through letters within an alpha-chunk. Furthermore, these productions were general enough to apply to any letter comparisons within any of the alpha-chunks.

For example, when the model determines which waypoint occurs alphabetically, BEP or BEO, it determines the first and second letters of the waypoints are identical. Next it determines that O and P are different, and retrieves the γ alpha-chunk. The model then iterates through γ's letter slots, reaching O before P, providing information to the model that BEO comes before BEP in the waypoint list, and to retreat (–) rather than advance (+) through the list.

Although the letter comparison procedure and the declarative structure of the alphabet were based on Klahr et al.'s model, the process differs slightly. For their model to obtain the chunk containing the letter O, it would require retrieving α and β chunks first, then retrieving the γ chunk. Once the γ chunk was retrieved, it would be serially searched for O.

Integration: Sharing Production Rules Across Goals

The methods comprising the subgoal methods and selection rules *obtain*, *compare*, and *change* gleaned from the task analysis suggest that there should be a high proportion of shared production rules to set flight parameters and waypoints when integrating the two models within ACT-R. The similarity in procedures for setting the flight parameters was high, and the only difference was the setting adjustment increments retrieved from declarative memory. Hence, each flight parameter (i.e., airspeed, altitude, and course) shared 100% of its production rules with the other flight parameters. However, production sharing between the setting flight parameters and waypoints was not nearly as high, with a minimum of 32% and a maximum of 44.5%. The minimum value comes from the model not having to serially search through an alpha-chunk, and the maximum value comes from the model having to exhaustively search through the largest alpha-chunk, α.

Both models were successfully integrated into a composite, with a relatively high degree of production rule sharing. In the following section I report the composite's validity.

Composite Model Validation

Model and human participants set the airspeed, course, altitude, and waypoint to determine if the composite model provided valid predictions. Data were collected from three dependent variables: 1) interclick duration, which was operationally defined as the time between clicks beginning after method 2 of the *compare* subgoal, 2) the number of mouse clicks to complete the goals, and 3) the total time to complete the goal. Interclick duration represents temporal dynamics between clicking an adjustment button and determining if the new setting value is the desired setting value (from method 4 through method 6 of the *change* subgoal). The number of clicks and the setting duration reflect the accuracy of the task analysis presented above.

Method

Participants were instructed to set the airspeed, course, altitude, and waypoint 20 times each. There were five human and 10 model participants. Human and model participants interacted with the same environment. Although the model had no knowledge of the course value continuous loop, human participants were instructed that both the waypoint and course setting values were continuous loops.

Base levels for alpha-chunks were set to a high initial value to account for early learning of the alphabet and a lifetime of use. All other ACT-R parameters were set to values necessary for other components of the synthetic teammate. These values were set prior to running the model and remained unchanged. Finally, production compilation and production utility learning were not active during model runs, and the model was reset after setting the flight parameters and waypoint 20 times each.

Twenty randomly selected airspeeds, altitudes, courses and waypoints were randomized for each participant and model run. The model operated as if the desired setting was provided from another teammate through the communication system. Consequently, neither the model nor human participants performed the *obtain* selection rule from the task analysis, presented above.

Results

A comparison between human and model data revealed little deviation between model and human performance, across the dependent variables from each of the four goals (i.e., setting airspeed, course, altitude, and waypoint), *RMSD* = 1.20; r^2 = 0.98.

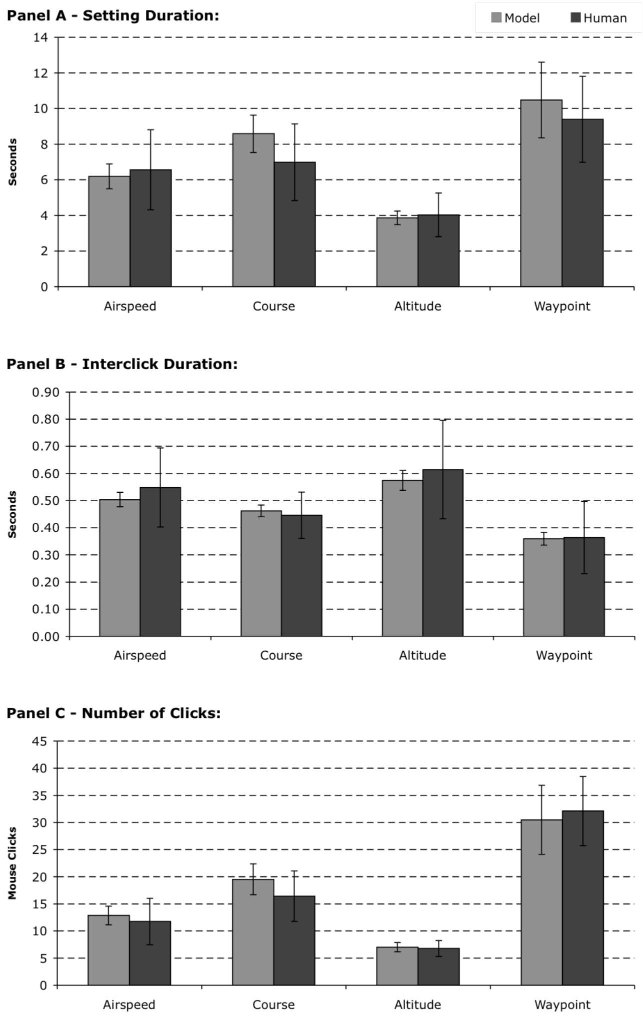


Figure 3. Results from model validation effort for the three dependent variables. Error bars are standard error.

Results indicate a very good fit between the composite model and human data. Interestingly, and not surprisingly, the course flight parameter has the poorest fit to human data, and likely stems from not incorporating knowledge of the setting's continuous loop of setting values.

Discussion and Areas for Improvement

This excellent model fit to human data resulted from performing a detailed task analysis, finding previously published models suitable to perform requisite tasks, and incorporating the models into a composite using a cognitive architecture. Although the model successfully predicts human data, there are clear areas for improvement. First, the selection of the flight parameter setting adjustment buttons is done using a function call external to ACT-R. Incorporating this decision process, while maintaining the model fit to human data is highly desirable. Second, it would be an improvement to enable the model to handle the continuously looping values of the course flight parameter.

The Klahr et al. (1983) model of letter recall and comparison was successfully integrated with other aspects of the synthetic teammate task behavior component. Furthermore, changing Klahr et al.'s serial search across alphabet nodes to a parallel retrieval process using ACT-R's spreading activation mechanism along with the close fit, points to an interesting possible extension to Klahr et al's model. The Lovett (1998) model of choice was less integration–more inspiration. There was also complete sharing of production rules across procedures for setting the different flight parameters, and decent sharing across procedures for setting flight parameters and waypoints. This high degree of production rule reuse reflects success in model integration.

When developing large-scale complex models, such as a synthetic teammate, the model must be capable of completing multiple disparate tasks. Model inspiration and/or integration of existing models provide the developer with the ability to model cognitive activities that may be outside their own area of expertise. The success of the composite model further demonstrates that the development of computational cognitive models has matured enough to draw inspiration from, or integrate, previously published models.

Acknowledgments

The research reported here was supported by grants from the Air Force Office of Scientific Research (AFOSR #FA9550-07-0081) and the Office of Naval Research (ONR #N000140910201) awarded to Dr. Nancy Cooke and Dr. Christopher Myers, respectively.

References

Anderson, J. R. (2007). How Can the Human Mind Occur in the Physical Universe? Oxford: Oxford University Press.

Ball, J. T., Myers, C. W., Heiberg, A., Cooke, N. J., Matessa, M., & Freiman, M. (2009). The Synthetic Teammate Project. Paper presented at the 18th Conference on Behavior Representations in Modeling and Simulation, Sundance, UT.

Cooke, N. J., & Shope, S. M. (2005). Synthetic Task Environments for Teams: CERTT's UAV-STE. In Handbook on Human Factors and Ergonomics Methods (pp. 46-41 - 46-46). Boca Raton: CLC Press, LLC.

John, B. E. (1996). TYPIST: A theory of Performance in Skilled Typing. Human-Computer Interaction, 11(4), 321-355.

Klahr, D., Chase, W. G., & Lovelace, E. A. (1983). Structure and Process in Alphabetic Retrieval. Journal of Experimental Psychology: Learning, Memory, & Cognition, 9(3), 462-477.

Lovett, M. C. (1998). Choice. In J. R. Anderson & C. Lebiere (Eds.), The Atomic Components of Thought (pp. 41). Mahwah: Lawrence Erlbaum Associates.

Applying Occam's razor to paper (and rock and scissors, too): Why simpler models are sometimes better

Antonio Napoli (antonio.napoli@phd.units.it)
Danilo Fum (fum@units.it)
Dipartimento di Psicologia, Università degli Studi di Trieste
via S. Anastasio 12, I-34134 Trieste, Italy

Abstract

A commonly held idea is that people engaged in guessing tasks try to detect sequential dependencies between the occurring events and behave accordingly. For instance, previous accounts of the popular Rock Paper Scissors game assume that people try to anticipate the move an opponent is likely to make and play a move capable of beating it. In the paper we propose that players modulate their behavior by reacting to the effects it produces on the environment, i.e., that they behave exactly as they do in non competitive situations. We present an experiment in which participants play against a computer controlled by different algorithms and develop a procedural model, based on the new ACT-R utility learning mechanism, that is able to replicate the participants' behavior in all the experimental conditions.

Keywords: Rock-Paper-Scissors, reinforcement learning, procedural learning, ACT-R, neural nets.

Introduction

The capability of adapting to changes occurring in the environment and to anticipate future events constitutes a critical factor for organisms' survival, and humans and animals have been tuned by natural selection to become receptive to subtle variations in the external contingencies. Adaptivity and proactivity are realized essentially through a process of selection by consequences—named also law of effect (Thorndike, 1898), operant conditioning (Skinner, 1938) or reinforcement learning (Sutton & Bartho, 1998)—i.e., on the idea that organisms modulate their behavior by reacting to the effects it produces on the environment.

Some predictions organisms routinely make concern the behavior of other organisms. A particular situation in which such predictions are useful is given by competitive games. It's obvious that, if we knew in advance the move our opponent is going to make, our life would become easier. In the paper we deal with Rock Paper Scissors (aka Roshambo), a competitive game that, while being extremely simple to describe and play, presents a series of interesting features when considered from a cognitive point of view.

The following section presents the essentials of the game and describes some strategies that have been suggested to play it effectively. Next, we review previous studies which investigated the behavior of human players in this task and proposed some models to explain it. As it will become apparent in the following, a common theme underlying this work is that people attempt to succeed at the game by trying to anticipate the move the opponent is likely to make and playing a move capable of beating it. We advance, on the other hand, a simpler explanation for the players' performance which relies on the same principle of selection by consequences that explains most of the behavior in non competitive situations. We present an experiment in which participants play against a computer controlled by different algorithms and develop a procedural model based on the new ACT-R utility learning mechanism that is able to replicate the participants' behavior in all the experimental conditions.

Rock Paper Scissors

Rock Paper Scissors (henceforth RPS) is a competitive two-person game which is played through a series of turns in which players make their moves simultaneously. The outcome of each turn is determined as follows: Rock beats Scissors, Scissors beats Paper, but Paper beats Rock. If both players make the same move, the turn is considered as a tie. That's all, as far as the game's rules are concerned.

In RPS no move—no "pure strategy", in terms of Game Theory (Von Neumann & Morgenstern, 1944)—can be considered as the best to play. Concepts like "better", "bigger", "stronger" and similar are possible only referring to sets for which a partial ordering could be established, and this requires the existence of a transitive relation among set members, an eventuality that cannot be realized in RPS where the relation "beats" originates a closed loop.

Considered from the point of view of the Game Theory, RPS is classified as a two-person zero-sum game. For all games of this kind there exists a solution, i.e., a rule or norm that prescribes how the game should be played. Assuming perfectly rational players, the solution coincides with the Nash equilibrium at which neither player could hope to achieve a better performance by modifying their strategy, In case of RPS, the Nash equilibrium is reached by choosing the three possible moves randomly with equal probability, i.e., by playing a mixed strategy through a stochastic combination of the pure strategies.

While game theorists could consider RPS as a trivial game, there are two facts that make it intriguing from a cognitive point of view. First of all, humans are notoriously bad at generating random moves (Rapoport & Budescu, 1997; Wagenaar, 1972), so theorists could not easily practice what they preach. Being unable to play randomly, humans necessarily display sequential dependencies among the moves they make that could be exploited by a clever opponent. Second, the mixed strategy has the advantage that no strategy can beat it but it also has the disadvantage that there is no strategy that it can beat. In other words, it guarantees a break-even result in the long run, regardless of how strong (or how weak) the opponent is, but it does not allow a player to reach consistent wins.

In fact, aficionados consider RPS a game of wit, not a game of chance. Even a cursory look at the web site of the World RPS Society (www.worldrps.com) or a quick skim of *The official rock paper scissors strategy guide (*Walker & Walker, 2004) should convince that RPS experts use their insight to try to anticipate the opponent's move, possibly recurring to particular sequences of moves to try to induce predictable responses in the other player. The problem is that, to exploit a weakness in the opponent's play, you need to make non-random moves, which makes you vulnerable.

A clearer idea about which strategies could succeed at the game may be obtained by looking at the results of the First and Second International RoShamBo Programming Competition—held at the University of Alberta, Canada, in September 1999 and July 2000, respectively—two tournaments between computer programs playing RPS in which each program competed against all others. Because organizers enrolled in the competition some really weak programs that produced easily predictable move sequences, a program that played the optimal strategy without trying to exploit the competitors' deficiencies (running at the same time the risk to expose its owns) could reach only weak results. It should be noted that all programs could store the complete sequence of moves played by themselves and by the opponent, a feature which human players, due to their memory limitations, cannot easily rely upon.

The programs adopted essentially two high-level strategies to choose their moves. The first one was based on pattern-matching and tried to exploit the statistical regularities occurring in the sequence of moves produced by the opponent. The second one relied on some kind of meta-reasoning to determine how the opponent would choose its move. One of the most complicated strategies of this kind was represented by the so called Sicilian-reasoning according to which a program tried to figure out the competitor's move by assuming that it will think like itself, taking however than into account the fact that the competitor was likely to use Sicilian reasoning too, and giving thus raise to a “I know that you know that I know ...” recursive pattern. This approach was very effective and programs adopting it ranked among the best.

While computer programs could shed light on how RPS should be played by perfectly rational agents with unlimited memory, we could ask how individual with bounded rationality, cognitive limits and emotions (i.e., normal people) really play the game.

Previous work

In the last decade Robert West, with Christian Lebiere and coworkers, produced a series of studies (West, 1999; Lebiere & West, 1999; West & Lebiere, 2001; Routledge-Taylor & West, 2004, 2005; West, Stewart, Lebiere & Chandrasekharan 2005) focused on the analysis of human behavior in the RPS and on the attempts to simulate it. These studies present several experiments whose results are explained through models that differ slightly from paper to paper. Through their comparative exam it is possible, however, to extract a unitary view and a coherent story that we are now going to tell.

According to the authors, people engaged in the RPS, and similar guessing tasks, try to detect regularities in the occurring events—in our case in the sequence of moves made by the opponent—and use this information to modulate their behavior. If both players use the same strategy of sequence detection, they enter in a state of reciprocal causation in which each player tries to influence the opponent's behavior while being, simultaneously influenced by it. The result is a dynamic, coupled system capable of generating patterns of interaction that could not be explained by looking at each system in isolation.

The players' behavior could be explained and replicated by a model capable of storing a variable number of previous opponent's moves. Differently from the computer programs playing the same game, the model has a reduced memory buffer whose capacity constitutes a critical factor in determining its behavior. A model which stores only the previous opponent's move is said to be a Lag1 model, if it stores the previous two moves is said to be Lag2, and so on.

The intuitive idea behind the models is that, if players could figure out what an opponent, having made the moves represented in the memory buffer, is going to do, they should make the move capable of beating it. This idea has been realized and implemented in different ways.

West (1999), Lebiere & West (2001) and Rutledge-Taylor & West (2004) used a two layers neural net which received in input the opponent’s moves and gave as output the move made by the player. The input layer comprised a number of node triples (each node representing Rock, Paper or Scissor, respectively) corresponding to the number of opponent's moves the model could store: one three-nodes group for Lag1—storing only the last move—two groups for Lag2— storing the last and the last but one moves—etc. Each input node could have a value of 0 or 1. More particularly, for each input triple, the node corresponding to the move made by the opponent received an activation value of 1 while the remaining two nodes got a 0.

All the nodes of the input layer were linked to the three nodes, one for each move, constituting the output layer. The weights of the links connecting the input and output nodes were initialized to 0 (in West, 1999 and West & Lebiere 2001) or were assigned a value randomly chosen from the set {-1, 0, +1} (in Rutledge-West, 2004). The value of an output node was determined by summing the weights of the links coming from the activated input nodes, i.e., from those input nodes having their activation set to 1. The network returned the move associated with the highest-value output node, possibly making a random choice in case of multiple nodes with the same activation.

After each choice, the link weights were adjusted according to the outcome. Two main policies were followed for updating the links. In the "passive" models, wins were rewarded by adding 1 to the weights of the links coming from the activated nodes (i.e., input nodes with a value of 1) and leading to the output node corresponding to the chosen move, losses were punished by subtracting 1, while ties kept all the links unvaried. "Aggressive" models, on the other hand, treated ties like losses and subtracted 1 to the links connecting the activated input nodes with the non-winning output node.

West & Lebiere (2001) carried out a series of experiments in which human participants played against different versions of the model and compared these results with those obtained by having different models compete against each other. In general, games in which identical versions of the models were pitted against each other ended in a tie. On the other hand, it was found that a broader memory span or a more aggressive attitude provided a competitive advantage: Lag2 models were able to reliably defeat Lag1 models while aggressive versions were superior to passive ones. Interestingly, the advantage provided by an extra lag or a more aggressive attitude were additive and approximately equal in magnitude.

Coming to human players, they were able to win on the average 9.99 turns (after a 300 turns game) more than the Aggressive Lag1 and 11.14 turns (after 287) more than the Passive Lag2 when pitted against these algorithms. According to West & Lebiere (2001), humans perform like the Aggressive Lag2 that constitutes, according to the authors, the best model for their behavior. Humans showed indeed a small but statistically significant trend to lose, instead of tie, against this model, but this effect was attributed to the fact that they were not able, due to lack of motivation and/or fatigue, to play in the same consistent manner as their computerized opponent.

These findings were congruent with those reported in those papers (West, 1999, Lebiere & West, 1999; Rutledge-Taylor & West, 2005, West, Stewart, Lebiere, & Chandrasekharan, 2005) that utilized models based on the ACT-R (Anderson & Lebiere, 1998) cognitive architecture. The idea that RPS is played exploiting the last moves to try to anticipate the next one to is maintained in these models, but in this case the sequence of moves is stored and retrieved not through a neural net but by taking advantage of the ACT-R declarative mechanism, while the model's choices, instead of being driven by the nodes' activation values, are demanded to the ACT-R procedural system.

The ACT-R declarative memory stores chunks containing the opponent's previous patterns and a prediction for the next move. The most important procedure used by the models is the following (slightly adapted from Lebiere & West, 1999, p. 297, and of intuitive significance):

Sequence Prediction

```
IF no move has been played
    and the opponent last played move(s) L2 (and L1)
    and move(s) L2 (and L1) are followed by move L
    and move L is beaten by move M
THEN play move M.
```

For each turn, the model recalls the most active chunk matching the two (for Lag2 models) or the last (for Lag1 models) opponent's move(s). The model notes the move predicted by the chunk and plays the move that beats it. After both players have made their choice, a new chunk containing the update moves and the corresponding prediction is formed, or a previously existing chunk already storing the same information is reinforced.

Using this approach, Lebiere & West (1999) were capable of replicating the results of West (1999) while Rutledge-Taylor & West (2005) constructed several models capable of replicating the results of West & Lebiere (2001).

The Experiment

Our interest for RPS arose after a series of experiments (e.g., Fum & Stocco, 2003; Fum, Napoli, & Stocco, 2008,; Stocco, Fum, & Napoli, 2009) which found the participants' behavior heavily influenced by the principle of selection by consequences. These experiments, however, dealt with non competitive situations, i.e., situations that could be classified as "a *one-person game*, sometimes called a *game against natur*e, wherein a single player makes a decision in the face of an environment assumed to be indifferent or neutral" (Zagare, 1984, p. 11). To investigate whether the same principle could hold in competitive situations like RPS, we established the following experiment.

Participants played three rounds of RPS against a computer controlled in each round by a different program. A first group of participants (in the "Classic" condition) interacted with the computer through an interface which adopted the usual symbols, i.e. clinched fist for Rock, flat hand for Paper, and closed hand with extended index and Scissors. Participants were instructed about the rules of the game and were told that the computer was following in each round a different strategy that could however be defeated, at least in some cases. Immediately after the participants' made their choice, the computer move was displayed together with the turn outcome. Wins allowed the participants to gain one point (+1), losses were punished by the same amount (-1), while ties left the score unaltered (0).

A second group of participants was engaged in the same task arranged, however, as a game against nature. In this "Nature" condition the computer was presented not as a competitor but as a neutral game device allowing participants gain or lose points. In fact the programs the computer used were exactly the same of the previous condition. Instead of the classic RPS symbols, however, participants saw on the screen three geometric figures representing a sphere, a cube and a pyramid. At the beginning of each round the computer randomly matched each figure with an RPS move and behaved accordingly. Participants were told that they could obtain in each turn a score of +1, 0 or -1. It was also said that the criterion according to which the computer assigned scores to figures could not be easily guessed and that, in any case, it would change in each round. By relying on their "intuition" participants had to try to obtain in each round as many points as possible.

Another difference between these conditions, in addition to the setting and the use of different move images, was given by the fact that the computer, instead of the move it made in each turn, displayed the complete payoff matrix allowing participants to see both the score gained by their move and the scores they could have gained by making the alternative choices. Suppose a participant chose the sphere (matched for that round with Scissors) and the computer the cube (matched with Paper, while the pyramid was matched with Rock). The outcome was shown by displaying +1 in correspondence to the sphere (because Scissors beat Paper), 0 near the cube (each move ties with itself) and -1 near the pyramid (because Rock beats Scissors).

In the experiment a third condition (named "Implicit") was utilized that was presented against as a competitive situation in which participants had to choose one of the new symbols (the sphere, the cube and the pyramid) displayed on the screen. Participants were told that each figure could beat another figure, tie with itself, and lose against the third one but the payoff matrix was not revealed and had to be discovered by playing the game.

In summary, participants played against the computer, controlled by the same algorithms, in a classic RPS game, in a situation disguised as a game against nature, and in a competitive framework with unknown payoffs. We wanted to establish how participants would perform in the three conditions and, in particular, whether their behavior should be explained by using models of a different kind.

Method

Participants and design. Sixty students (37 males) enrolled at the University of Trieste, Italy, were recruited as participants. Their age varied between 18 and 32 years (*M*=21.4, *SD*=3.7). Participants were randomly assigned to one of the three experimental conditions (Classic, Implicit, and Nature) in which they were engaged in three RPS rounds, each one against a different algorithm whose order was counterbalanced between rounds. The experiment followed therefore a 3x3 mixed design with Condition as between subjects and Algorithm as within subjects factors.

Materials. Three algorithms were used in the experiment. The first one, Lag2, replicated the program described in the previous section. In this case, however, we implemented a Passive Lag2 algorithm which updated the net weights by assigning +1 to wins, 0 to ties and -1 to losses. The second algorithm, Random, played according to the optimal strategy by choosing its moves randomly from a uniform distribution. The third one, Biased, made also random moves but it sampled from a distribution in which one of the moves had a 50% probability of being selected, a second one a 35% probability and the third one 15%. At the beginning of each task the computer assigned randomly one move to each probability class.

Procedure. The experimental sessions were held individually. Participants were instructed about the game rules and it was stressed that each round would be played against a different opponent (in the Classic and Implicit condition) or a different program (in the Nature condition) which followed its own criteria in choosing the moves. After reading the instructions, participants were involved in three 100-turn RPS rounds, each round played against a different algorithm. Participants made their choices by clicking on an image displayed at the vertex of an imaginary equilateral triangle. The images were randomly placed at the vertices for each participant. After participants made their choice, the move played by the computer and the outcome score were shown in the Classic and Implicit conditions while in the Nature condition the scores that could have been obtained by choosing the alternative moves were also displayed. During the task participants were kept informed through a colored bar of their running total that was however reset after each round.

Results

We first ascertained whether the rounds, per se, could influence the participants' performance, i.e., whether the mere fact of having played 100, 200 or 300 RPS turns, independently of the condition and the algorithm, could represent a significant factor in determining their behavior. The scores obtained in the rounds were as follows: *M*=-0.17, *SD*=9.78 for Round1, *M*=1.64, *SD*=11.18 for Round2, and *M*=0.27, *SD*=10.40 for Round3, respectively. A repeated measures one-way ANOVA on the scores of each round did not revealed (*p*=.65) any significant effect. No signs of learning (or fatigue) were thus evidenced that could hinder the interpretation of further results.

We then analyzed the factors manipulated in the experiment, i.e., the Condition to which participants were assigned and the Algorithm against which they played. Table 1 reports the means and the standard deviations of participants' scores.

Table 1: Means (and standard deviations) of the scores.

	Lag2	Random	Biased
Classic	**-3.80** (9.61)	**-1.40** (9.84)	**6.80** (7.14)
Implicit	**-6.45** (10.17)	**-1.20** (5.74)	**5.95** (11.73)
Nature	**-6.40** (9.29)	**4.65** (8.76)	**6.55** (9.60)

A mixed design ANOVA revealed as significant the effect of the Algorithm only (*F*(2)=25.13 *p*<.000001), while Condition (*p*=.44) and the interaction (*p*=.29) did not seem to play any role. In other words, participants behave in the same way when they played the classic RPS game, knowing the relationship that existed between the moves, in the implicit RPS, when the payoff matrix was unknown, and in the non-competitive Nature condition. Two further ANOVAs restricted on the scores of the first 20 and 40 turns, respectively, provided similar results—i.e. the only significant effect was that of the Algorithm—suggesting that no payoff learning was needed to perform in the Implicit condition as in the Classic one, with the results of both conditions were similar to those obtained in the Nature one.

Figure 1 reports the results obtained by collapsing the three conditions. As evidenced, participants lost against Passive Lag2, tied against Random and won against Biased. Two *t* tests confirmed that both in the case of Lag2 (t(*t*(59)=-4.47, p < .0001) and of the Biased algorithms (*t*(59) = 5.24, *p*<.00001) the participants performance differed significantly from 0.

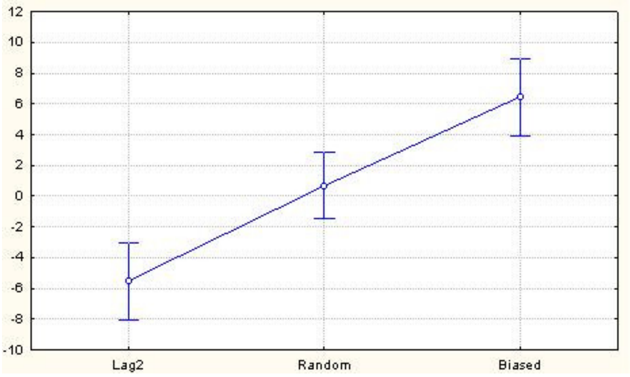


Figure 1. Mean scores obtained against the different algorithms (collapsed conditions) with bars representing 95% confidence intervals.

Modeling the results

Our results contrast with the commonly held assumption that players try to succeed at RPS by anticipating the opponent's move and by making a move capable of beating it. This approach could possibly work in the Classic and Implicit conditions but cannot be applied in the Nature one where the idea of “opponent's move” does not simply make sense. To explain the experimental results we would therefore be obliged to assume that different strategies were applied in the competitive and non-competitive settings. Moreover, following the same assumption, we would expect to find a difference, at least in the first phases, between the Classic and the Implicit conditions. While participants in the former know immediately what to play to defeat an anticipated move, those in the latter have to learn what beats what; since the very first turns, however, the results obtained in these conditions are not discriminable and, again, are similar to those obtained in the third one. Finally, because human participants, as reported by West & Lebiere (2001), had a tough time playing against the Aggressive Lag2 algorithm, we tried to facilitate their task by having them compete against a more manageable version. If Aggressive Lag2, which systematically beats the passive version, represents however the best incarnation of the above mentioned assumption, it is difficult to explain why participants systematically, and independently of any fatigue sign, lost against the Passive Lag2 algorithm.

A principle of economy suggests the possibility that, at least under the conditions examined in our study, participants do not follow different strategies and *do not* try to anticipate the opponent's move but they simply make those moves that are more likely to succeed, independently of the condition to which they have been assigned and the opponent they competed with.

To test this hypothesis, we pitted against our algorithms three different models, representing possible participants' strategies in RPS. The models were the Passive Lag2, Passive Lag1 and a procedural model which exploited the ACT-R's new utility learning mechanism (Anderson, 2007).

The idea on which Procedural ACT-R is based is that an organism, facing the problem of choosing among different moves or actions, will select that which worked best in the past, i.e., the action that was most useful in the previous history of the organism. The model associates therefore to each option a utility measure that is updated after each move application according to the reward it receives from the environment, Starting at an initial value of 0 the utility U_i of each move i at time n is updated according to the formula (Anderson, 2007, p. 160):

$$U_i(n) = U_i(n-1) + \propto [R_i(n) + U_i(n-1)]$$

where α is a learning parameter and R is the outcome received in each turn (the usual +1 for a win, 0 for a tie, and -1 for a loss). The choice of the move is however not deterministic but subjected to noise. The probability P_i that a given move i will be selected among a set of j options (including i too) is given by:

$$P_i = \frac{e^{U_i/s}}{\sum_j e^{U_j/s}}$$

where *s* it the noise parameter.

The choices made by the model are thus regulated by α and s; we set α to 0.2 and varied s to fit the experimental data. To allow a fair comparison, we implemented NoisyLag2 and NoisyLag1, the nondeterministic versions of the respective models in which the choices were made according to the same formula used in Procedural ACT-R.

A final problem had to be solved before running the simulations. While Procedural ACT-R could be employed in all the experimental conditions, it was not immediately clear how NoisyLag2 and NoisyLag1 could be used to simulate the participant's behavior in the game against Nature, in which the opponent's moves were not available to them. The only data the models had available were represented by the scores obtained by making the different moves. Discarding obviously the idea of storing the score associated with the move chosen by the player, we tried the other options obtaining a surprising result (at least for us): the behavior of the models was exactly the same independent of the fact they stored the best moves (i.e. the moves leading to a score of +1), the worst ones (-1) or those in-between (with a 0 score). In fact, after a moment's thought, we realized that the opponent's moves stored by the West & Lebiere’s (2001) lag models were exactly the moves a player should have made to tie in each turn!

That said, we ran a series of simulations with each model starting with s=0.1 and progressively augmenting the parameter through increments of 0.1 up to a final value of 14.0. We simulated 1000 runs of the model for each parameter value against each algorithm, and considered that the model was fitting the data when the 95% confidence intervals of the models' results were completely included within the 95% confidence intervals of the participants' data. The Procedural ACT-R model (with noise values ranging from 0.39 to 0.44) was the only model capable of replicating the participants' performance against all the different algorithms both in term of general performance (total means) and in terms of a temporal series of five successive 20-turns blocks. Both NoisyLag1 and NoisyLag2 did not to fit the participants' data against Lag2 because they were not able, for any s setting, to generate scores that were less than those obtained by the opponent. These models were in a sense too powerful to be considered as a good representation of the people's performance.

Conclusions

In the paper we presented the first results of a research project aimed at investigating the possibility of applying the principle of selection by consequence, traditionally adopted to explain human behavior in games against nature, to model the players' performance in competitive games. We focused on RPS which was previously explained by adopting some form of belief models, i.e. models that "starts with the premise that players keep track of the history of previous play by other players and form some belief about what others will do in the future based on past observation. Then they tend to choose a best-response, a strategy that maximizes their expected payoffs given the beliefs they formed." (Camerer & Ho, 1999, p.2) We found that two models of this kind (NoisyLag2 and NoisyLag1) were isomorphic with models that work by taking into account only the environmental rewards and we found that they were too powerful to be able to explain the human behavior. A purely procedural model based on the ACT-R new utility mechanism was able to fit the experimental data providing thus a simpler and more general explanation for the players' behavior.

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Camerer, C., & Ho, T. (1999). Experience-weighted attraction learning in normal form games. *Econometrica, 67,* 827-874.

Fum, D., Napoli, A., & Stocco, A. (2008). Somatic markers and frequency effects: Does emotion really play a role on decision making in the gambling task? *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, 1203-1208.

Fum, D., & Stocco, A. (2003). Outcome evaluation and procedural knowledge in implicit learning. In R. Alterman & D. Kirsh (Eds.), *Proceedings of the 25th Annual Conference of the Cognitive Science Society.* Boston, MA: Cognitive Science Society, 426-431.

Lebiere, C., & West, R. L. (1999). A dynamic ACT-R model of simple games. *Proceedings of the Twenty First Annual Conference of the Cognitive Science Society,* 296-301. Mahwah, NJ: Erlbaum.

Rapoport, A. & Budescu, D. V. (1997). Randomization in individual choice behavior. *Psychological Review, 104,* 603-617.

Rutledge-Taylor, M. F., & West, R. L. (2004). Cognitive modeling versus game theory: Why cognition matters. *Proceedings of the Sixth International Conference on Cognitive Modeling* 255-260. Pittsburgh, PA: Carnegie Mellong

Rutledge-Taylor, M. F. & West, R. L. (2005). ACT-R versus neural networks in Rock = 2 Paper Rock Scissors. *Proceeedings of the Twelfth Annual ACT-R Workshop, Trieste,* Italy: Università degli Studi di Trieste, 19-23.

Skinner, B. F. (1938). *The behavior of organisms.* New York: Appleton-Century-Crofts.

Stocco, A., Fum, D., & Napoli, A. (2009). Dissociable processes underlying decisions in the Iowa Gambling Task: A new integrative framework. *Behavioral and Brain Functions, 5*, 1

Sutton, R. S., & Bartho, A. G. (1998). *Reinforcement learning: An introduction.* Cambridge, MA: MIT Press.

Thorndike, E. L. (1898). Animal Intelligence; an experimental study of the associative processes in animals. *Psychological Review Monograph Supplement 2*(8):1-109.

Von Neumann, J., & Morgenster, O. (1953). *Theory of games and economic behavior.* Princeton, NJ: Princeton University Press.

Wagenaar, W. A. (1972). Generation of random sequences by human subjects: A critical survey of the literature. *Psychological Bulletin, 77,* 65-72.

Walker, D. & Walker, G. (2004). *The official rock paper scissors strategy guide.* New York: Simon & Shuster.

West, R. L. (1999). Simple games as dynamic distributed systems. *Proceedings of the Complex Games Workshop.* Tsukuba, Japan: Electrotechnical Laboratory Machine Inference Group.

West, R. L. & Lebiere, C. (2001). Simple games as dynamic, coupled systems: Randomness and other emergent properties. *Journal of Cognitive Systems Research, 1,* 221-239.

West, R. L., Stewart, T. C., Lebiere, C., Chandrasekharan, S. (2005). Stochastic Resonance in Human Cognition: ACT-R Versus Game Theory, Associative Neural Networks, Recursive Neural Networks, Q-Learning, and Humans. *Proceedings of the 27th Annual Conference of the Cognitive Science Society,* Mahwah, NJ: Erlbaum, 2353-2358.

Zagare, F. C. (1984). *Game Theory: Concepts and applications.* Newbury Park, CA: Sage Publications, Inc.

Reading Aloud Multisyllabic Words: A Single-Route Connectionist Model for Greek

Konstantinos D. Outos (outosk@hol.gr)
Graduate Program in Cognitive Science, Athens University Campus
GR-15771 Athens, Greece

Athanassios Protopapas (protopap@ilsp.gr)
Institute for Language & Speech Processing, Artemidos 6 & Epidavrou
GR-15125 Maroussi, Greece

Abstract

Multisyllabic word reading has received little attention in existing computational models, which are designed for English. The Greek language uses mainly multisyllabic words, while its orthography is feedforward consistent and includes a stress diacritic. We present here a computational model of reading aloud Greek words and nonwords, based on the connectionist "triangle" model, adapted to the Greek orthography, using a novel input representation. The network displays several effects from the word reading literature and successfully assigns stress. Further investigations are underway.

Keywords: Multisyllabic words; stress assignment; triangle model; reading aloud; nonword reading; connectionist; Greek.

Introduction

Computational Models of Reading Aloud

It has been 20 year since Seidenberg & McClleland (1989) presented the first "triangle" connectionist model of reading aloud. Since then, a variety of computational models have been presented, single-route (Plaut, McClelland, Seidenberg & Patterson, 1996; Harm & Seidenberg, 2001, 2004) or dual-route (DRC: Coltheart, Rastle, Perry, Langdon & Ziegler, 2001; CDP+: Perry, Ziegler, & Zorzi, 2007; Zorzi, Houghton, & Butterworth, 1998). These models differ in many respects, such as the existence of pre-defined graphophonemic decoding rules versus learning procedures, and the localist or distributed nature of lexical representations.

The triangle family of models has the advantage of discovering, during a training phase, regularities in the relations between orthographic and phonological representations in a set of words. The regularities are then generalized to novel stimuli, such as nonwords. Three implementations of this approach have been reported (Seidenberg & McClleland, 1989, Plaut et al., 1996; Harm & Seidenberg, 2001, 2004), using different representations, leading to differences in reading performance (Seidenberg & Plaut, 2006).

Most existing models have focused on reading English monosyllabic words. Problems associated with multisyllabic words, such as syllabification and stress assignment, have led to their exclusion. Likewise, most available data on human reading concern monosyllable English words, facilitating comparative model assessment. To present monosyllabic words on a connectionist input layer, Harm & Seidenberg (2001, 2004) used slots corresponding to single letters separated into subsyllabic units. Onset and coda consonants were placed around the central vowel nucleus. This representation aimed to minimize the dispersion problem (Plaut et al., 1996), while allowing the model to "capture the fact that phonemes in different positions sometimes differ phonetically" (Harm & Seidenberg, 1999, p. 493).

Exclusive attention to monosyllables, and corresponding syllable-based representations, pose severe constraints to extension towards multisyllabic words. Specifically, to retain the existing structure, words must be pre-syllabified before being presented to the model. However, this limits the applicability of this approach to situations in which orthographic syllabification is possible.

The Greek Orthography

The Greek orthography is feedforward consistent and predictable to a large extent (about 95%; Protopapas & Vlahou, in press). Most graphemes can be mapped unambiguously to single phonemes when context is taken into account. The only substantial source of inconsistency concerns words containing the CiV pattern, that is, an unstressed grapheme normally mapping to /i/ when it follows a consonant and precedes a vowel (Protopapas & Vlahou, in press). In such cases there are two possible pronunciations, one of which contains an /i/ and another which contains a palatal consonant. The correct pronunciation is lexically determined. In rare cases, this situation leads to homographs. For example, the Greek words for "permission" and "empty" are both written as άδεια. However, "permission" is the three-syllable word /'a.ði.a/ with an [i] forming the nucleus of the second syllable, whereas "empty" is the two-syllable word /'a.ðja/ with the palatal consonant [j]. Therefore, the CiV phenomenon affects not only graphophonemic consistency but also orthographic syllabification as well.

In Greek, lexical stress always falls on one of the last three syllables and is affected by morphology (Revithiadou, 1999). Stress is orthographically marked with a special diacritic on every word with two or more syllables (Petrounias, 2002). This diacritic also disambiguates certain vowel digraphs, therefore it is necessary to include in orthographic representations, along with diaeresis.

These characteristics make the vowel-centered syllabic slot representation unsuitable for Greek multisyllable words, at least for the orthographic input layer. The CiV phenomenon creates a major challenge because it precludes presyllabification. Therefore, it is not possible to simply add more syllabic templates to existing input representations.

In this paper we present a model of reading aloud Greek multisyllabic words and nonwords based on the triangle model, taking into account the special properties of Greek orthography. We report preliminary results showing that the model is successful in accounting for the Greek situation.

The Model

Our model is a modified version of the Harm and Seidenberg (1999, 2004) network (see also Zevin & Seidenberg, 2006). Modifications (besides the number of nodes at each layer) concern mainly the input and output representations. The network was designed to read words with 2–5 syllables written with 4–10 letters. These limits were imposed to reduce demands on computational resources and training time.

Design and Implementation

The model has 449 orthographic input units, 500 hidden units, 630 phonological output units, and 400 cleanup units (Figure 1). Units in the phonological layer were connected to every phonological and cleanup unit, including themselves. Each cleanup unit was also connected to all phonological units, turning the phonological-cleanup layers into a recurrent network, capable of creating attractors based on regularities discovered at the phonological output. These attractors aim to improve reading performance (especially for nonwords) by allowing the phonological output to settle into globally coherent states (Harm & Seidenberg, 1999).

Model implementation was based on “MikeNet” (version 8), as modified by Zevin and Seidenberg (2006), with additional changes to handle stress marking.

Representations

We considered a number of alternative approaches to the problem of presenting orthographic input without first syllabifying, keeping with the spirit of the preceding implementations and restrictions imposed by the available code. Simply presenting a (left-aligned) letter string to the network, without any positional constraints, led to poor performance. Therefore, we imposed grouping of successive consonant or vowel letters, without regard to syllabification. Analysis of consonant and vowel alternation in the entire training corpus indicated that this 40-slot template can hold every word with the sole restriction that adjacent consonants and adjacent vowels remain grouped:

```
CCCVVVVVCCCCCVVVVVVCCCCVVVVVCCCVVVVVCCVVC
```

To present a word to the network, each group of consecutive consonant or vowel slots is filled with letters, in a left-to-right direction. Shorter words leave the rightmost groups empty. All words fill at least one slot of the first two groups. To reduce the total number of connection weights, each slot was constrained to contain letters that appear in this position

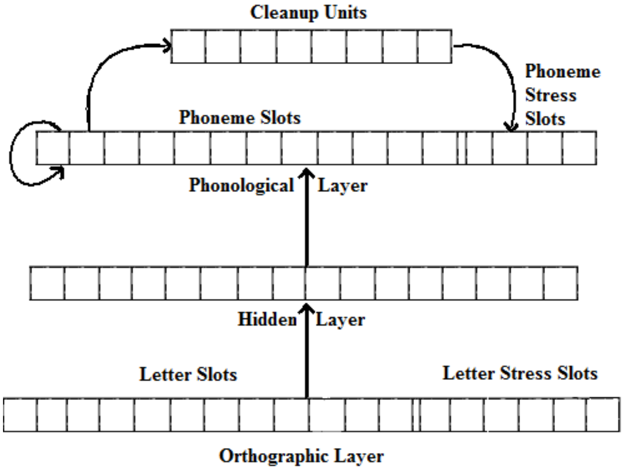


Figure 1. Model architecture.

in the training corpus. Thus, the input layer consists in one binary vector for every slot with length depending on the number of letters that may appear in it. Presence of a letter is indicated by setting its corresponding unit to 1. In addition to the 40-slot letter template, a set of 22 slots made up the stress marking template. These correspond to 5 vowel groups of the letter representation, encoding position and identity of the vowel letter marked with the stress diacritic.

For example, the orthographic representation of άδεια is:

Letters:

```
___α____δ____εια________________________
CCCVVVVVCCCCCVVVVVVCCCCVVVVVCCCVVVVVCCVVC
```

Stress:

```
   α____        ______     _____     ____   __
```

The phonological output representation is not affected by the CiV pattern. Therefore, the English implementation was simply augmented with additional syllable slots of the appropriate structure. There were 5 groups of 6 slots each, representing 5 syllables with up to 3 onset consonants and up to 2 coda consonants, making up a CCCVCC syllabic template. Each slot encodes 18 phonetic features, in corresponding units, as an 18-bit binary vector. A phoneme is present in a slot when its corresponding feature units are given an activation value of 1. Five additional slots represented the vowel of the stressed syllable. Each contained the same 18-bit vector as the corresponding vowel phoneme slot. This somewhat redundant representation of stress was as simple as possible given the constraints of the code.

Here is the phonological representation of the word /'a.ði.a/ (vertical lines indicate syllable boundaries):

Phonological:

```
___a__|__ði__|___a__|______|______
CCCVCC|CCCVCC|CCCVCC|CCCVCC|CCCVCC
```

Stress:

```
   a      _      _      _      _
```

Whereas the homograph /ˈa. ðʝa / is represented as

Phonological:

_ _ _ a _ _| _ ð ʝ a _ _|_ _ _ _ _ _|_ _ _ _ _ _|_ _ _ _ _ _
CCCVCC|CCCVCC|CCCVCC|CCCVCC|CCCVCC

Stress:

a _ _ _ _

Training the Network

The training corpus was a list of 120,745 word types with 2–5 syllables, 4–10 letters long, from an online written text corpus of about 30 million words (Hellenic National Corpus; hnc.ilsp.gr). The corresponding pronunciations were derived using a grapheme-to-phoneme transcription model developed for text-to-speech applications (Chalamandaris, Raptis, & Tsiakoulis, 2005). Words were presented to the model proportionally to their frequency of appearance in that corpus after logarithmic compression and conversion to relative probability, following Harm & Seidenberg (1999, Equation 1, p. 495), to allow low frequency words to appear during training. The relative probability of all words with less than .3 occurrences per million was set to 0.2.

Beginning with random initial weights, 9,000,000 training trials were given. On each trial, a word was randomly chosen from the training list and was presented to the model if a random number from a flat [0,1] distribution did not exceed its transformed frequency. The orthographic representation of the word was fixed on the input layer for 10 "ticks" (i.e., activation update cycles through the network). The phonological output after 12 ticks was compared to the target output. Connection weights were adjusted using continuous recurrent backpropagation, with a learning rate equal to 0.1.

A subset (about 10%) of the training corpus (12,017 words) was retained and used to track performance during training every 50,000 trials (plotted in Figure 2). Each word was presented to the network for 18 ticks. A response was considered correct if produced within that time. More ticks were allowed in testing than in training, because longer words might require longer times for the network to settle to the correct pronunciation. Pilot trials showed significant improvement using 18 ticks for testing, compared to 12 ticks. Further increases (up to 45 ticks) did not improve performance significantly. Only 12 ticks were used during training, because this led to faster training, presumably due to increased pressure for the network to learn.

Post-training Tests

To study the effects of the most important variables known to affect word reading times in human participants (Balota, Yap, & Cortese, 2006), a set of 150 words were selected to span a range of length, frequency, neighbourhood size, and bigram probability. The words were chosen so that the intercorrelations between these basic variables were minimal (Spearman's ρ<0.2) and not statistically significant, in order to isolate individual effects. A corresponding set of 150 nonwords were constructed, based on these words, with similar properties and variable ranges, taking care to avoid resemblance of nonwords to specific words. Human reading performance data for these stimuli are currently being collected, for future comparison with model performance.

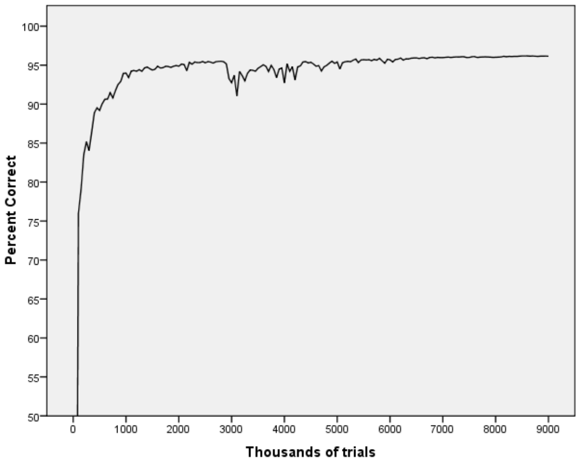


Figure 2. Word reading accuracy during training (%).

After training, these words and nonwords were presented to the model with and without a stress diacritic, to assess its performance on segmental pronunciation and on stress assignment. Response time was measured, for correct responses only, by the number of ticks it took the model to achieve a representation of the correct response. The orthographic input was presented to the input layer for 18 ticks.

The variables onto which response time were regressed included: probability of appearance in training, as a measure of frequency (words only); length (in letters); number of syllables, number of orthographic neighbours (words only), cumulative bigram (letter) probability, mean orthographic syllable frequency, and orthographic transparency measured as minimum non-directional grapheme-phoneme type probability (least transparent sonograph; Spencer, 2007). Of these, only syllabic frequency differed significantly between words and nonwords ($t(298) = 2.007$, $p = .01$).

Results

Overall, 96.14% of words were read correctly after 9m trials. Performance may increase with further training, as no asymptote is apparently reached (Figure 2). 2.37% of the errors were stress assignment errors, that is, stress produced on a vowel other than the one marked with the diacritic (e.g., /ði.i.ˈli.ze/ instead of /ði.ˈi.li.ze/). 4.53% were both segmental and stress assignment errors in that stress was erroneously produced on an incorrect vowel. The rest were segmental errors, that is, incorrect phonemes. When a stressed vowel was produced incorrectly, it was nevertheless stressed, showing that stress diacritic position was interpreted correctly (e.g., /ðʝa.vo.ˈlis/ instead of /ðʝa.vo.ˈlɛs/). 18.97% of the errors concerned stress assignment combined with a segment displacement, usually resulting from incorrect parsing of a CiV. In such cases, the stress diacritic was placed on the correct vowel at its new position, showing that the model can properly combine letter and diacritic information (/fi.'ɛ.stes/ instead of /'fçɛ.stes/).

To examine the effects of the aforementioned predictor variables, simple regressions were employed. This allows reliable detection of the most important effects, because the main variables were specifically uncorrelated in the test sets. Multiple regressions will be used once the critical variables are identified in the analysis of human data.

Of the 150 test words, presented with a stress diacritic, 98% were read correctly, with an average response time (RT) of 4.44 ticks. Linear regression, separately for each predictor, showed that number of letters accounted for 4.4% of RT variance (R^2 = .044, standardized β = .084, p = .011), and bigram probability for 4.5% (β = −.028, p = .010).

Of the 150 nonwords presented with a stress diacritic, 92% were read correctly, at a mean RT of 4.54 ticks. Number of letters accounted for 5.6% of RT variance (β = .116, p = .005), number of syllables for 3.2% (β = .179, p = .035), and bigram probability for 7.5% (β = -.046, p = .001).

These results are summarized in the following table.

Table 1: Summary of simple regression results for items presented with a stress diacritic. Significant predictors are presented in order of variance proportion accounted for.

Predictors	Words	Nonwords
Significant	Bigram prob. N Letters	Bigram prob. N Letters N Syllables
Not significant	N Syllables Frequency Syllable frequency N Neighbors Transparency	Syllable frequency Transparency

The RT difference between correct readings of words and nonwords was not significant (*t(283)* = -1.457, p = .148).

Of the 150 words presented without a stress diacritic, 53.33% were read correctly with a mean RT of 5.11 ticks, significantly slower than when presented with a stress diacritic (t(80) = -3.267, p = .002). Sixty errors were segmentally correct but incorrectly stressed words, including 6 responses with no stressed letter and 54 stressed at a different position. For the words produced correctly, the model assigned stress 27.5% on the final syllable, 42.5% on the penult, and 30% on the antepenult, a uniform distribution of stress (χ^2(2) = 3.100, p = .212). Taking into account all words, stress assignment to the final, penult, and antepenult was 30.7%, 44.5%, and 24.8%, respectively (significantly nonuniform, χ^2(2) = 8.423, p = .015). The correct stress positions for all words in the testing set were 30.7% on the final, 40% on the penult, and 29.3% on the antepenult; and for words stressed incorrectly by the model, 35.1%, 47.4%, and 17.5%, respectively. The corresponding proportions for the entire corpus, considering multisyllabic words only, are 30%, 44.9%, and 25%, respectively (Protopapas, 2006). Two words, which formed a CiV pattern when the stress diacritic was removed, were read by the model with the alternate pronunciation of the CiV and were stressed appropriately considering the segment pattern produced.

Of the 150 nonwords presented without a stress diacritic, 42.67% were read correctly, with a mean RT of 5.3 ticks, significantly slower than when presented with a stress diacritic (t(63) = -2.019, p = .048). The model stressed nonwords 28.7% on the final syllable, 44.1% on the penult and 27.2% on the antepenult, a nonuniform distribution (χ^2(2) = 7.162, p = .028).

Discussion

This is a first attempt toward a computational model of reading aloud Greek words and nonwords. The number of monosyllables in Greek is very small: fewer than 500 types were reported by Protopapas & Vlahou (in press), most of which were unrepresentative of the language in being either function words or recent loans. Therefore extension of existing approaches to multisyllabic representations was necessary in order to capture the major characteristics of this language. The Greek orthography is highly consistent for reading, with the exception of the CiV pattern. On the one hand, the high consistency makes the task of mapping letter sequences to phoneme sequences easier. On the other hand, the presence of the CiV phenomenon dictated a substantial change to the design of the model's representations, because pre-syllabification is not possible. The new design has the additional benefit that the number of syllables is not limited, as long as computational resources can handle the training. The concomitant drawback is susceptibility to the dispersion problem, because the same graphophonemic mappings must be learned repeatedly in different input-output slot positions.

Our novel orthographic representation bears some interesting features. The lack of pre-syllabification forces the model to learn mappings that might otherwise be distinguished by their subsyllabic position. The model must learn to map the letter sequences at the input layer to the syllabified output at the phonological output layer. This is not trivial, because letter positions are not fixed at the input, as they depend on word length and morphology. It is especially complex for successive vowel letters (up to 6 slots for single graphemes or digraphs) mapping to multiple syllables. However, even though the model is forced to learn the same mappings at several different slots, this does not seem to pose a serious problem for word reading performance or generalization to nonwords. This may be due to the relatively simple grapheme-phoneme mappings of the Greek orthography. It remains to be investigated whether the model learns to read in the same way as Greek readers do.

Figure 2 shows that early in training (100,000 trials) the model can already read correctly a considerable proportion of words (about 76%). The number of word types presented to the model (120,745), relative to the number of training trials (9 million), is huge, compared to the 3,123 words and 1 million trials in Harm & Seidenberg (1999), 6,103 words and 1.5 million trials in Harm & Seidenberg (2004), 5,870 words and 1 million trials in Zevin & Seidenberg (2006), and 9,911 words and 1.2 million trials in Pagliuca & Monaghan (in press). Despite the low ratio of tokens to types, the lack of syllabification, and the dispersion over slots, the model can read correctly more than 96% of the training corpus. This may be due to high orthographic transparency.

Due to the frequency-modulated random selection procedure, many words that are read correctly were never presented to the model during training, so they must by read by grapheme-to-phoneme conversion, as nonwords. It is instructive to examine the model's response to various letter strings containing the CiV pattern, because there is no rule for the CiV, either in terms of a statistical preponderance (Protopapas & Vlahou, in press) or in human participants' reading behavior (Protopapas & Nomikou, 2009). The results of the training test show instances of words read with the incorrect alternative pronunciation (but not with unrelated phonological outputs). These cases concerned low-frequency words that were never or little presented to the model during the training, so they are functional nonwords. Further tests, with controlled sets of words, are underway.

Another novel feature of this model is stress marking in the orthographic representation, corresponding to the diacritic of Greek orthography. Although this is not the first attempt to model stress assignment in reading (e.g., Monaghan, Arciuli & Seva, 2008; Rastle & Coltheart, 2000) or to include an orthographic representation of stress (Pagliuca & Monaghan, 2009, in press), it is probably the first attempt to consider a distinct orthographic representation for the stress diacritic itself while retaining vowel letter identity. The model seems to have learned the constraints on stressed syllables. No stress assignment error was observed on a syllable earlier than the antepenultimate even though syllable positions were not fixed or right-aligned. Analysis of incorrectly stressed words showed that the model has a stress "preference" for the penultimate syllable, like humans, in both words and nonwords. There was no significant distortion of stress assignment toward any syllable, indicating that the model does not assign stress randomly but follows the distribution of stress positions seen on Greek words.

The network seems to have learned to make a connection between stressed vowels and the stress diacritic, even when the stressed vowel was incorrectly produced or placed in the wrong syllable. In such cases, stress followed the vowel, either by changing vowel or by changing position. Only 0.1% of the total training corpus was read with the correct segmental pronunciation and incorrect stress. This means that the model has learned to use the stress diacritic. On the other hand, the model's stress assignment performance deteriorated very substantially when the diacritic was not presented, indicating an excessive reliance on the diacritic. Although this outcome is justifiable on the basis of the reliability and validity of the stress diacritic, it stands in contrast to behavioral data showing that Greek readers are not affected by the lack of a stress diacritic (Protopapas, Gerakaki, & Alexandri, 2007; Protopapas & Gerakaki, in press). Pretraining the phonological layer might produce in improved fit to human performance by reinforcing stress vowel connections in word representations. A connection from the orthographic directly to the phonological layer (Zorzi et al., 1998) might also improve performance on unstressed words.

In this preliminary investigation, two sublexical properties were found to affect reading times: word length (measured in letters or, for nonwords, in syllables) and bigram probability. In a review of factors affecting visual word recognition, Balota et al. (2006) reported significant effects of word length for low frequency words and nonwords. The effect on both words and nonwords in our results may be due to the relatively few repetitions of each word during training. This renders words effectively low-frequency, because they did not have many opportunities to affect the connection weights. An alternative or complementary explanation may relate to a fine grain of graphophonemic representation, which is expected for a language with high feedforward consistency. Reliance on a fine grain can lead to stronger length effects as more graphemic units must be individually mapped. As a reviewer pointed out, this might also explain the lack of frequency and lexicality effects. It remains to be investigated whether evidence for larger units of graphophonemic mapping may accumulate with higher ratios of trials to word types in the training procedure.

Nevertheless, the significant word length effect seems to run counter to common expectations regarding connectionist models. According to Rastle & Coltheart (2006), word length effects should not be exhibited by single-route connectionist models, because entire words are read in parallel and not serially, grapheme-by-grapheme, as in some dual-route models. Our results, although preliminary, are inconsistent with this prediction, showing that word length effects are possible in parallel distributed processing models, even for the highly consistent mappings of the Greek orthography. This finding may depend on a large range of word lengths, as imposed by the multisyllabic input and by the stimulation of more phonological attractors when more letters appear at the input, in part due to dispersion. Thus, our model sheds light on a long-standing issue in modeling reading aloud, which was not possible to address with previous models dealing only with monosyllabic words.

Word frequency is one of the most important variables affecting word reading performance in English (Balota et al., 2006). Our model was not affected by word or syllable frequency in this preliminary investigation, which may be attributed to the low token-to-type ratio that renders trained words effectively low frequency. Balota et al. noted that low frequency words exhibit larger effects of sublexical regularity, such as bigram frequency, compared to high frequency words. This was borne out in our model and may be related to the dispersion necessitated by our input representation. Specifically, as the same letters appear at different positions, the model is exposed to input bigrams more consistently than to words with larger common parts.

On the other hand, the absence of expected transparency (mapping consistency) effects warrants further investigation. Orthographic neighbors were also expected to affect reading performance, however the situation with neighbors may differ substantially from English because most Greek words have few or no neighbors, (mean neighborhood size was 1.69), perhaps due to their overall greater length.

In conclusion, this paper presents a computational model of reading aloud that can read Greek multisyllabic words and nonwords, using a novel orthographic input representation that includes stress marking. Critically, orthographic input was not pre-syllabified, whereas phonological output was. Preliminary tests indicate that the model reads words and nonwords with reasonably high accuracy, assigns stress correctly based on diacritic information, and produces effects of word length, previously thought incompatible with parallel processing, but no effects of frequency, which are large and robust in human data and other models. Further tests and elaboration will take place as comparable human data for Greek become available.

Acknowledgments

We thank Jason D. Zevin for the source code used to implement the model and Efthymia C. Kapnoula for the word and nonword testing stimuli.

References

Balota, D. A., Yap, M. J., & Cortese, M. J. (2006). Visual word recognition: The journey from features to meaning (A travel update). In M. Traxler & M. A. Gernsbacher (Eds.) *Handbook of psycholinguistics (2nd ed.).*

Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. (2001). DRC: A dual route cascaded model of visual word recognition and reading aloud. *Psychological Review, 108*, 204–256.

Chalamandaris, A. Raptis, S., & Tsiakoulis, P. (2005). Rule-based grapheme-to-phoneme method for the Greek. In *INTERSPEECH-2005*

Harm, M. W., & Seidenberg, M. S. (1999). Phonology, reading acquisition and dyslexia: Insights from connectionist models. *Psychological Review, 106*, 491–528.

Harm, M. W., & Seidenberg, M. S. (2004). Computing the meanings of words in reading: Cooperative division of labor between visual and phonological processes. *Psychological Review, 111*, 662–720.

Monaghan, P., Arciuli, J., & Seva, N. (2008). Constraints for computational models of reading: Evidence from learning lexical stress. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum.

Pagliuca, G., & Monaghan, P. (in press). Discovering large grain-sizes in a transparent orthography: insights from a connectionist model of Italian word naming. *European Journal of Cognitive Psychology.*

Pagliuca, G. & Monaghan, P. (2009). A connectionist model of reading for Italian. In J. Mayor, N. Ruh, & K. Plunkett (Eds.), *Connectionist models of behaviour and cognition II* (pp.301–312). Singapore: World Scientific.

Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: The CDP+ model of reading aloud. *Psychological Review, 114*, 273–315.

Petrounias, E. V. (2002). *Neoelliniki grammatiki kai sigkritiki analisi, tomos A: Fonitiki kai eisagogi sti fonologia [Modern Greek grammar and comparative analysis, Vol A: Phonetics and introduction to phonology]*. Thessaloniki: Ziti.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review, 103*, 56–115.

Protopapas, A. (2006). On the use and usefulness of stress diacritics in reading Greek. *Reading & Writing: An Interdisciplinary Journal, 19*, 171–198.

Protopapas, A., Gerakaki, A., & Alexandri, S. (2007). Sources of information for stress assignment in reading Greek. *Applied Psycholinguistics, 28*, 695–720.

Protopapas, A., & Gerakaki, S. (in press). Development of processing stress diacritics in reading Greek. *Scientific Studies of Reading*.

Protopapas, A., & Nomikou E. (2009). Rules vs. lexical statistics in Greek nonword reading. To appear in the *Proceedings of the 31st Annual Conference of the Cognitive Science Society*.

Protopapas, A., & Vlahou, E. L. (in press). A comparative quantitative analysis of Greek orthographic transparency. *Behavior Research Methods*.

Rastle K, Coltheart M (2006), "Is there serial processing in the reading system; and are there local representations?", In S. Andrews (Ed.), *From inkmarks to ideas: Current issues in lexical processing.* Hove: Psychology Press.

Rastle, K., & Coltheart, M. (2000). Lexical and nonlexical print-to-sound translation of disyllabic words and nonwords. *Journal of Memory & Language, 42*, 342–364.

Revithiadou, A. (1999). *Headmost accent wins: Head dominance and ideal prosodic form in lexical accent systems* [LOT Dissertation Series 15 (HIL/Leiden University)]. The Hague: Holland Academic Graphics.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review, 96*, 523–568.

Seidenberg, M., & Plaut, D. (2006). Progress in understanding word reading: Data fitting versus theory building. In S. Andrews (Ed.), *From inkmarks to ideas: Current issues in lexical processing*. Hove: Psychology Press.

Spencer, K. A. (2007) Predicting children's word-spelling difficulty for common English words from measures of orthographic transparency, phonemic and graphemic length and word frequency. *British Journal of Psychology, 98*, 305–338.

Zevin, J. D., & Seidenberg, M. S. (2006). Simulating consistency effects and individual differences in nonword naming: A comparison of current models. *Journal of Memory and Language, 54*, 145–160.

Zorzi, M., Houghton, G., & Butterworth, B. (1998). Two routes or one in reading aloud? A connectionist dual-process model. *Journal of Experimental Psychology: Human Perception and Performance, 24*, 1131–1161.

Testing a Quantitative Model of Time Estimation in a Load-Switch Scenario

Nele Pape (nele.pape@zmms.tu-berlin.de)
GRK Prometei, ZMMS, Technische Universität Berlin
Franklinstr. 28-29, FR 2-7/2, 10587 Berlin Germany

Leon Urbas (leon.urbas@tu-dresden.de)
Dept. of Electrical Engineering and Information Technology,
Technische Universität Dresden
01069 Dresden Germany

Abstract

A model of prospective time estimation was tested in two experimental variations which examine the influence of load switch in task demands on time estimation. The model predicts these influences on time estimates by means of memory processes such as spreading activation. The approach was integrated into a cognitive architecture and has previously been tested successfully. In two experiments participants had to work on a counting task with different levels of working memory demands (High/Low). The participants had to stop each trial after a perceived duration of a previously presented sample of 100 seconds (altered reproduction method) and received feedback. In the Low group most trials were performed in low load and one or two trials in high load (load switch), and vice versa for the High group. For the Low group the model predicts overestimations at load switches, but underestimations for the High group. We found that the model predictions in the first experiment only match the experimental results for the Low group, most probably due to the experimental design. In the second experiment, the design was therefore slightly changed and the timing task was embedded into a manual control task within a microworld environment. In this setting the model predictions match the time estimates for both groups. The series of experiments reported give strong evidence that the model is able to capture and to predict influences of task demands on time-estimates. The timing model may be used as a base for modeling subjective temporal reasoning and the timing of interaction with a dynamic system.

Keywords: Time estimation, cognitive modeling, coordinative working memory, memory processes, spreading activation, feedback.

Introduction

People can be good at estimating time and they sometimes rely on their estimates even when they are part of a safety-critical system. However, in stressful situations or in the course of demanding tasks, time estimates might be distorted to a large degree.

Time perception is crucial for everyday purposes and especially in the area of human-machine-interaction. In the context of operator performance, supervision of processes is a time critical task that might be prone to human errors, if other task demands rise suddenly.

The influence of task demand on time estimation has been examined thoroughly. A number of factors that are said to have an influence on time estimation are discussed in the literature. The most frequently mentioned factors are: attention (Block & Zakay, 1996; Zakay, 1993; Byrne, 2006), memory load (Brown, 1997; Brown & West, 1990; Dutke, 2005), or simply forgetting to estimate time if the task gets more demanding (Taatgen et al., 2007).

The most prominent model is the Attentional Gate Model (Block & Zakay, 1996). This assumes that a mental pacemaker regularly generates pulses to measure time. If a person directs attention to the course of time, a gate opens and the pulses are accumulated in a cognitive counter. When attention is distracted by a secondary task, the gate remains closed, pulses are not accumulated and the time-estimation is distorted. This way estimations turn out to be shorter whenever attentional ressources are captured by demanding secondary tasks.

A serious shortcoming of the Attentional Gate (ATG) Model is that it does not differentiate between specific and overall task demands. The model proposes influence of general attention but does not capture differences of specific task properties. Dutke (2005) therefore designed a counting task experiment to investigate the influence of two different working memory demands (sequential and coordinative) on time estimation. According to the ATG Model both demands would equally influence time estimation because attention is needed in both cases. However, Dutke's results show that both factors influence task-performance, but only high coordinative working memory demands distort time-estimates.

For the domain of human-machine-interaction, the susceptibility to workload induced distortions of time estimation is of high importance because operators do experience strong changes in workload (see e.g. Decortis and Cacciabue 1999). This might eventually lead to mishandling of the system due to a wrong timing of action. Furthermore, one can observe that most often time estimates need to be given under the very same general conditions as the reference time representations have been acquired before. Therefore we chose to set up a model that is designed for reproduction of time estimates (e.g. instead of giving time estimates verbally).

In the following we first sketch our computational implementation of a variant of the ATG Model which is prone to different task demands. We then introduce shortly the counting task and its specific task demands that may distort time estimation. Finally we show a series of two experiments that have been designed to challenge the models predictions.

A Computational Model of Time Estimation Involving Memory Processes

The idea behind the proposed model (Pape & Urbas, 2008) resembles some broadly accepted components of the ATG Model (Block & Zakay, 1996) with a pacemaker that generates pulses, an accumulator and an estimator, but without an associated gate. The main difference to the ATG model is a specific working memory account, which is realized by a mechanism to provide short-estimates between meaningful events (or "contextual changes" in the words of Block & Zakay, 1996) and an updating or construction process that integrates these short-estimates into a time estimate of the whole episode.

Figure 1 sketches the basic idea of the model: The vertical dashed line represents the pulses generated by the pacemaker as time goes by. The accumulator collects these pulses until a meaningful event occurs (depicted by an 'X' on the dashed line). The count of collected pulses together with some contextual information is stored in a temporal chunk (the short-estimate) that may be understood as an element of episodic memory (Tulving, 2002). The updating process then constructs a new episode-estimate by retrieval of the latest episode-estimate from memory and adding of the short-estimate. For instance, at the second event in the example shown in Figure 1, the episode-estimate, which carries 5 pulses, is retrieved and the newly accumulated 6 pulses in the short-estimate are summed up. A new episode-estimate with 11 pulses is stored in memory while the former remains. With a perfect memory, this new episode-estimate will be retrieved when the next event occurs, because it is the most recently generated chunk (with the highest activation). Additional memory activities might influence the activation level of two consecutive episode-estimates in a way that the wrong episode is retrieved instead of the latest episode-estimate (see dash-dotted line in Figure 1). So in our example instead of a final time representation with 24 pulses, a representation with 20 pulses is stored in memory. Therefore, demanding tasks cause time representations with fewer pulses than less demanding tasks. This mechanism generates shorter time representations only. Overestimations occur when the generated time representation is longer than a former time-representation. Contrary to other timing-models, this model needs no additional elements for pacemaker and accumulator variance and no attentional gate. Distortions and distribution of time-representations emerge naturally by means of variance in memory processes.

This approach was integrated into the cognitive architecture ACT-R (atomic components of thought – rational analysis; Anderson et al., 2004) and is called TaSTE (Task Sensitive Time Estimation) Module.

We utilized the sub-symbolic declarative memory mechanisms proposed and implemented in ACT-R without changes. The activation level A_i of a chunk i is calculated by the base-level, a noise component ε (set to 0.1) and a context component which is not shown in equation 1. For base-level activation the number of presentations n for chunk i and the time since the *jth* presentation are taken into account. The decay of activation is calculated with d (set to 0.4)

$$A_i = \ln(\sum_{j=1}^{n} t_j^{-d}) + \varepsilon$$

Equation 1: Calculating activation of chunk i.

Activation spreading from the current goal towards the episode-estimates is enabled via the above-mentioned contextual information and helps to keep the episode-estimates retrievable. The parameter association strength was modified to s = 6.

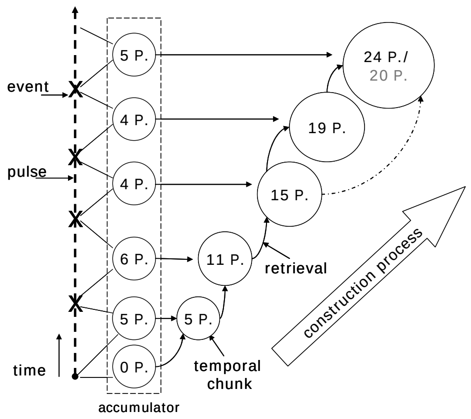


Figure 1: Construction process during an interval with several events. (Dotted arrow indicates the retrieval of an old instead of last time representation)

The Counting Task

The integrated timing-model was tested within a counting task (Dutke, 2005) with varying demands (sequential and coordinative) to compare human data to the predictions of the model. Sequential complexity refers to task variations that affect the number of simple and independent processing components and is demanding general intentional resources. Coordinative complexity refers to tasks in which the information flow between interrelated processing components needs to be coordinated (Mayr et al., 1996) and demands working memory resources.

In the counting task, the participants were asked to search lists of ten two digit numbers for either one or three targets (for low coordinative demand "16"; for high coordinative demand: "16", "38", and "67"). The sequential demand was varied with the overall number of targets contained in the lists (either 14 or 27 targets can be found within 40 lists).

The subjects have to count how often the different targets appear. On every third encounter of a target, the appropriate answer is given by pressing a specific key (e.g. labeled "18"), in all other cases the key marked with "No" should be pressed. After 400 sec., subjects were asked to reproduce the perceived duration by pressing a key to indicate the start and the end of the interval. Participants were randomly assigned to the four experimental conditions that result from the 2x2 between-subjects design (two levels of coordinative demands, two levels of sequential demands). Almost all participants underestimated the duration of the counting task. High coordinative demands produced larger reproduction errors and shorter estimates than low coordinative demands. For increased sequential demands the reproduction error was unaffected by the manipulation.

The model estimates showed the same effects of these demands as the human data (Pape & Urbas, 2008), because in the high coordinative condition more additional information has to be maintained. Both, simulation results in task performance and time estimations reveal comparable variability to each condition to empirical data, because the task model and the time module rely on retrieval processes where slight changes in activation lead to differences in results.

The load switch scenario

To adequately test the validity of the timing module we could either change the task or the scenario around the task as well as the estimation method. But, because with a new task it could be argued that the model data is dependent on the way the task was modeled and does not necessarily mirror the estimation processes that are assumed, we changed the task scenario and estimation method. This way we were able to reuse the model of the counting task that showed comparable performance to empirical data before (Pape & Urbas, 2008).

For the experiments reported here we also changed the interval duration to 100 seconds to check whether the model also holds for shorter intervals. Furthermore we modified the reproduction method. Instead of simply waiting, the participant had to work on the same task as in the encoding phase. We used repetitive timing to ensure that people were able to build up a good time representation before the load of the task switched (see Altman & Gray, 2008 for task switching scenarios) after several trials to either higher or lower coordinative demands.

Model runs

The model ran 22 times for each of two conditions representing the two groups used in the experiment for four different trials. To provide a reference the first trial was stopped after 100 seconds, the model thereafter used the built up representation as a reference to stop the next trial. In the case of the high condition group the first trial started with high coordinative demands which means the model had to cope in counting the occurrences of three targets and meanwhile building up a time representation. In case of the low condition group there was just one target to count. The time representation was used in the subsequent model run (a trial of equal load) for comparison to the new constantly updated representation. The task was stopped after an equivalent number of pulses had been collected. Because we assume that people build up a robust representation after a number of trials of equal load, we took the mean of the accumulated pulses for the interval and used it as time representation for the load-switch trial. In this trial the coordinative load changed compared to the previous which means low load in case of the high load group and vice versa.

This way we ended up with reproductions either derived in **inload** trials (trials according to the group condition) or **switch** trials for both groups (High/Low) (see model data figure 3 and 5).

No main effects in reproductions were found, but a significant interaction inload/switch*Group ($F(1,42)=7.5; p<.01$; $\eta^2=0.15$) show the different switch effects for the two groups. The model reproductions in the High group were much shorter in the switch trial and in the High group much longer than in the normal inload trials.

Experiment one

Our hypotheses generated by the model predictions were (1) reproductions performed in the same condition as experienced in the sample will be distributed around 100 seconds for both groups. (2) The load switch trial causes underestimations for the High group and overestimations for the Low group.

Participants

Forty-two participants (aged 21-48 years; Mean=26.05, SD=5.63) took part in the main experiment. The volunteers (25 male, 17 female) were paid 10 euros for participation.

Apparatus and setting

A standard keyboard was adapted as the entry device for the participants. Four keys on the number pad were covered with green tape that read 18, 34, 59, and also N and further apart another key marked Y. No sources of temporal information were available in the room.

Procedure

The participants were randomly assigned either to the High or Low group. Every experimental session began with the presentation of the sample duration. In every trial including the sample in the beginning, participants had to count the number of targets that appeared within the lists. Lists of 5 to 12 items (two digit numbers) appeared one after another in the middle of the screen for a time according to the number of items (3 to 10 seconds). Between lists the monitor was blank for 2 seconds. After the duration of 100 seconds, which was unknown to the participants, the task stopped and an instruction appeared on screen that the participant had to reproduce the experienced duration by starting and stopping the next trial by using the 'Y' button.

Session structure

The same instructions and training trials were given to all participants. After completing a demographic questionnaire the participants were informed about the counting task. After a training trial, the participants read another instruction about the experimental procedure and the reproduction procedure. Furthermore, they were informed that the length of lists, the number of lists, and the number of targets vary. Before a new trial started, the participants were informed look either for all three targets 18, 34, 39 or for just one target. No further targets were to appear than those mentioned.

After the 1st, 6th and 8th (last) reproduction the participants had to fill out a NASA-TLX questionnaire (Hart & Staveland, 1988) that measures workload.

Immediately following the experimental trials we conducted a structured interview to learn about the time estimation strategy, the difficulties of the tasks, and their strategy for the counting task the participants had applied.

Testing

After the sample-duration-trial (of 100 seconds), participants had to reproduce the duration 8 times with subsequent feedback about the quality of their reproduction (figure 2). A horizontal bar indicates the correspondence between sample duration and reproduced duration. If the horizontal bar is located below the middle area, the duration has been underestimated.

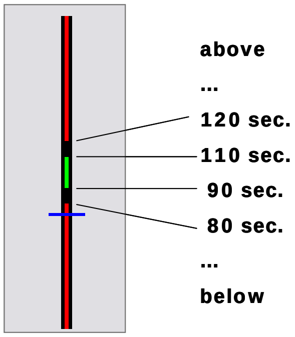


Figure 2: The feedback given after each reproduction (here the feedback indicates a strong underestimation).

No information about the assigned condition was given to the participants. Before every trial, participants were informed about the targets they had to count. When the trial was not stopped by the participant after 140 seconds, a message appeared on screen saying that no more lists are going to show up and the 'Y' button is to be pressed.

Results and comparison of experiment one

For the scores on the NASA TLX (1st and 2nd measures in load, the 3rd after the switch) a one-way repeated ANOVA revealed a significant interaction effect of NASA-TLX score and group (Low, High) (F(2,76)=13.8, p<.01; η^2=0.267). Planned contrasts showed that the first two measures in the NASA-TLX changed significantly to the third (group Low: F(1,38)=15.6, p<.01; η^2=0.291; group High: F(1,38)=24.1; p<.01; η^2=0.388). Therefore, the load-switch in the last trial seemed to have had the expected effect.

For the eight time-reproductions of the empirical data, the repeated ANOVA revealed a significant effect for reproductions (F(7,238)=8.86, p<.01, η=.46). There was a significant difference between inload reproductions to switch reproductions (F(2,43)=7.78;p<.05;η^2=0.35). But the predicted interaction between group and trial condition did not reach significance. Planed contrasts reveal that for the low group most reproductions in the inload condition were significantly shorter than the final one. Therefore just the low group showed the predicted switch effect, as shown in figure 3.

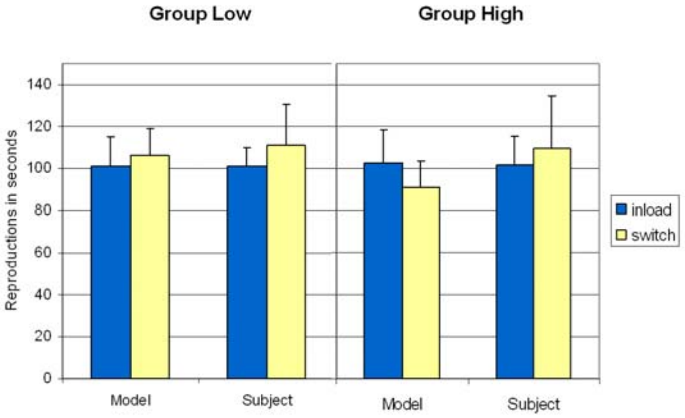


Figure 3: Model reproductions compared to empirical data in experiment one.

We had four possible explanations for the results. First, subjects in the High group reported in the interview that they were aware that their time perception would change after the switch to low load and therefore waited longer until they stopped the trial. Subjects in the Low group were too busy in the last high condition trial to reason about these things.

Second, some authors (Sturmer, 1966; Wearden et al., 1999) report that repetitive time estimations in a monotonic task with no background activity and no feedback reveal a lengthening effect, which means that estimates get longer the more estimates were made. We tried to avoid this by giving feedback but this might not have helped to totally prevent the effect. Third, the NASA TLX might have interfered with the estimates because after presenting the questionnaires participants showed a slightly longer reproduction.

Forth, the single switch in load after 7 inload trials might have been unexpected, causing participants to overestimate although participants were trained in both conditions. Therefore we conducted a second experiment that avoids the assumed factors.

Experiment two

For the second experiment subjects experienced four inload trials including the sample trial without reproduction. Then a first switch trial occurred. After that another four inload trials including the sample trial had to be completed before the second switch trial occurred.

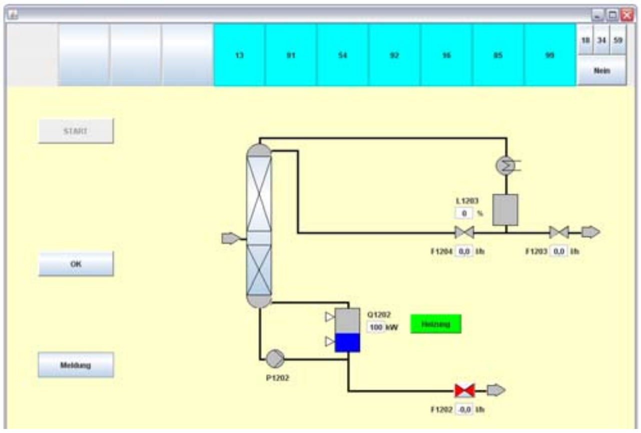


Figure 4: In the microworld scenario the dark blue liquid had to stay between the white triangles by opening the red valve below. At the same time the blue alarms had to be handled.

To add more background activity we used a microworld environment of an operator task in which the level of some liquid had to be maintained within a certain range and alarms had to be responded to (see figure 4). The operator has to handle certain important alarms which need to be counted, and ignore the remaining alarms. The alarm task resembled the counting task and for every new trial the participant in the role of the operator was informed about the important upcoming alarms just as in the previous experiment.

We assumed that the high workload of this multitasking set effectively hinders the participants to post-hoc reason about their way of time perception and compensate. Furthermore, we hoped to reduce the lengthening effect by inducing the first switch earlier and 'start anew' with a second sample trial afterward. Finally we eliminated the NASA TLX to avoid additional interference effects.

Participants second experiment

Fifty-three participants (aged 21-40 years; M=26.43, SD=4.84) took part in the second experiment. The volunteers (28 male, 25 female) were paid 10 euros for participation.

Procedure, structure and testing

The second experiment resembled the first experiment with the above mentioned differences. The participants received extra training for the operator task and had to interact with the mouse in the microworld environment instead of with the keyboard.

Results and comparison

A main effect for reproductions was found ($F(3,153)=4.382$; $p<.01$; $\eta^2=.079$). Furthermore for the second half of the experiment we found a significant interaction between reproductions and group ($F(3,153)=2.6$; $p=.053$; $\eta^2=0.049$) and a linear trend in increasing estimates by means of a planned contrast ($F(1,51)=10.7$; $p<.01$; $\eta^2=.173$).

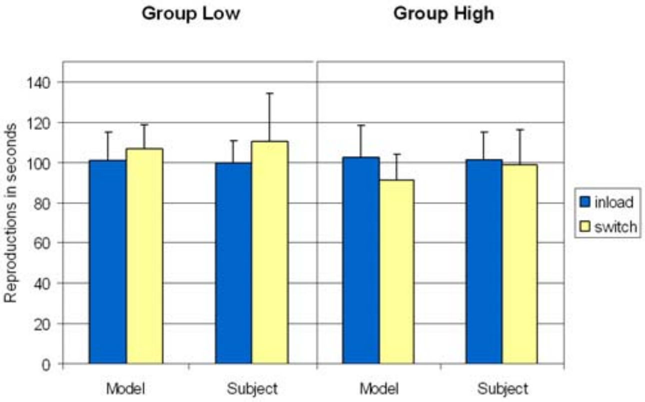


Figure 5: The comparison of the model predictions and the empirical reproductions of experiment two.

This time we find the predicted significant interaction for group*inload/switch as predicted by the model. Figure 5 shows the differences between inload and switch reproductions for the second half of the experiment with the significant interaction inload/switch*group ($F(1,51)=5.3$, $p<.05$; $\eta^2=.094$). Furthermore, not only the means of experiment and simulation resemble each other pretty much, but also the measures for the distributions are comparable (see table 1).

Table 1: The means and standard deviation in brackets of time reproductions for model and experimental data.

	Low		High	
	Model	Participants	Model	Participants
inload	100.9 (14.2)	99.8 (11.3)	102.4 (15.9)	101.6 (13.3)
switch	106.5 (12.6)	110.2 (23.9)	91.2 (12.6)	98.8 (17.6)

Discussion

The two experiments show that the TaSTE Module is able to predict human time estimates even under changing task demands not just in respect to the mean of the estimates but also in terms of distribution. Other current timing modules for ACT-R are not able to predict these task demand induced differences. The module presented by Taatgen et al. (2007) which has been designed for short term estimates assumes that distortions emerge from people "forgetting" to estimate time and restarting their timer. This would indeed result in shorter estimates. But the probability for restarting the timer has to be estimated for each task. Therefore it is only possible to replicate but not to predict distortions in time estimates. Byrne's (2006) timing module assumes that attention factors cause distortions. In the case of the counting task the same amount of time is available for attention to time under difficult and easy conditions. In the case of experiment two, hardly any time is given for attention to time, because of the supervision task for the level of the liquid and the alarms. Byrne's timing module therefore predicts no difference for the load switch but a high difference for experiment one and two.

Nevertheless our model still needs further work, because additional factors seem to influence time estimation. These are (1) the lengthening effect of repetitive estimates, (2) additional questionnaires that might also lengthen estimates such as the NASA TLX, and (3) people are aware of their time distortions and counteract if they have the resources to do so.

At least for the lengthening effect there might be some explanation in the implemented model: More temporal chunks will reduce the activation spread to the distinct chunks and more confusion will occur during the updating process of the time representation.

Conclusion

The results of the experiments show that variance and distortion of human time estimation may be modeled by basic memory mechanisms as implemented in ACT-R. In this sense the TaSTE module is an integrated model that builds upon principles that are found in other cognitive domains. This does not imply that time estimating processes have to work they way sketched here. But formalizing a quantitative model allows evaluating different mechanisms in different task setting.

Next steps are to analyze the sensitivity of the model against different kind of tasks. The limits of the model predictions concerning the durations between events and the influence of the structure of short-estimates should be investigated further.

Acknowledgments

We thank Liubov Cramer, Rebecca Wiczorek, Julia Niemann and Thomas Nicolai for their help in preparing and conducting the experiments, our cooperation partners at UCIT (Finland) for discussing this paper, and Richard Holmes, Berlin for comments on language. This work was sponsored by Volkswagen Stiftung (Research Group Modelling of User Behaviour in Dynamic Systems) and Deutsche Forschungsgemeinschaft (DFG Research Training Group Prospective Design of Human Technology Interaction, GRK 1013).

References

Altmann E.M., & Gray, W.D. (2008). An Integrated Model of Cognitive Control in Task Switching. *Psychological Review,* 115, 602-639.

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An Integrated Theory of the Mind. *Psychological Review,* 111, 1036–1060.

Block, R.A., & Zakay, D. (1996). Models of Psychological Time Revisited. In H. Helfrich (Eds.), *Time and mind. Proceedings of the International Symposium on Time and Mind.* Kirkland, WA: Hogrefe & Huber, 171-195.

Brown, S.W. Attentional Resources in Timing: Interference Effects in Concurrent Temporal and Nontemporal Working Memory Tasks. *Perception & Psychophysics,* 59 (1997), 1118-1140.

Byrne, M.D. (2006). An ACT-R Timing Module Based on the Attentional Gate Model. ACT-R Workshop. Retrieved from http://chil.rice.edu/projects/agm/index.html [2009-03-08].

Brown, S.W., & West, A.N. (1990). Multiple timing and the allocation of attention. *Acta Psychologica,* 75, 103-121.

Cowan, N. (2005). *Working memory capacity*. New York, NY: Psychology Press.

Decortis, F. & Cacciabue, P. C. (1988) Temporal dimension in cognitive models. In E. W. Hagen (Ed). *Proceedings of the 4th IEEE Conference on Human Factors and Power Plants.*

Dutke, S. (2005). Remembered duration: Working memory and the reproduction of intervals. *Perception & Psychophysics,* 67, 1404-1422.

Hart, S.G., & Staveland, L.E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. A. Hancock & N. Meshkati (Eds.), *Human Mental Workload* 139-183. Amsterdam: Elsevier Science Publishers.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1999). Modeling working memory in a unified architecture: An ACT-R perspective. In Miyake, A. & Shah, P. (Eds.) *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control.* New York: Cambridge University Press.

Mayr, U., Kliegl, R., & Krampe, R.Th. (1996). Sequential and coordinative processing dynamics in figural transformations across the life span. *Cognition,* 59, 61-90.

Pape, N., & Urbas, L. (2008). A Model of Time-Estimation Considering Working Memory Demands. *Proceedings of the 30th Annual Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Sturmer, G. (1966). Stimulus variation and sequential judgements of duration. *Quarterly Journal of Exp. Psychol.* 18, 4, 354-7.

Taatgen, N.A., van Rijn, H., & Anderson, J.R (2007). An Integrated Theory of Prospective Time Interval Estimation: The Role of Cognition, Attention and Learning. *Psychological Review,* 114, 3, 577-598.

Tulving E. (2002). Episodic memory: from mind to brain. *Annual review of psychology* 53, 1–25.

Wearden, J.H., Pilkington, R., & Carter E. (1999). "Subjective lengthening" during repeated testing of a simple temporal discrimination. *Behavioural Processes,* 46, 25-38.

Zakay, D. (1993). Relative and absolute duration judgments under prospective and retrospective paradigms. *Perception & Psychophysics* 54, 656–664.

Towards a neural network model of the visual short-term memory

Anders Petersen[1] (ap@imm.dtu.dk)
Søren Kyllingsbæk[1,2] (sk@imm.dtu.dk)
Lars Kai Hansen[1] (lkh@imm.dtu.dk)

[1]Department of Informatics, Technical University of Denmark,
Richard Petersens Plads, B. 321,
2800 Kgs. Lyngby, Denmark

[2]Department of Psychology, Copenhagen University
Linnésgade 22,
1361 Copenhagen, K, Denmark

Abstract

In this paper a neural network model of visual short-term memory (VSTM) is presented. The model aims at integrating a winners-take-all type of neural network (Usher & Cohen, 1999) with Bundesen's (1990) well-established mathematical theory of visual attention. We evaluate the model's ability to fit experimental data from a classical whole and partial report study. Previous statistic models have successfully assessed the spatial distribution of visual attention; our neural network meets this standard and offers a neural interpretation of how objects are consolidated in VSTM at the same time. We hope that in the future, the model will be developed to fit temporally dependent phenomena like the attentional blink effect, lag-1 sparing, and attentional dwell-time.

Keywords: visual attention, visual short-term memory, the magical number 4, winners-take-all network

Introduction

For everyday life, it is important for us to be able to perceive, comprehend, and react to events in our environment. Often, our rate of success is heavily dependent upon how efficient and how fast we can process, interpret and react to sensory stimuli, e.g. like when we are driving a car.

In the following we shall refer to *visual attention* as the process that enables us to focus our processing resources to certain important objects in the visual scene. Following the theory of visual attention (TVA, Bundesen, 1990) we assume that features have already been extracted and objects successfully segregated on the basis of their individual feature spaces. Our model deals with the important question of how only a limited sub span of all objects are actually selected and further encoded into VSTM.

Cattell already in the late 19th century demonstrated a surprising limit in how many objects that can be perceived at the same time – a limit only about 4 objects which may be held in the VSTM at the same time (Cattell, 1886; Cowan, 2000). This finding is independent of the number of objects visually presented at the same time (Sperling, 1960). Evidence further exist that the "magical number" of 3-to-4 objects is largely independent of how many features are encoded for each object, i.e. the complexity of the visual object, does not hold an influence on the memorial capacity of the VSTM; see (Luck & Vogel, 1997), but see also (Alvarez & Cavanagh, 2004).

Modelling the function of the VSTM, it is essential that the inherent capacity limitation is properly mimicked, since it seems a fundamental limit of the system. Most likely the VSTM would be heavily overloaded, should the system lack the ability to represent only the most salient of the visually appearing objects

The model

The model that we are presenting in this paper can actually be understood as three consecutive processes (See Figure 1).

The first process is simply extraction of visual features, we speak of this process as *'object matching'*, since we find it relevant to think that objects in the visual field are to some extent 'matched' against objects representations in Visual Long-Term Memory (VLTM). In this paper we do not consider the problem of which feature extraction techniques are biologically most plausible or perhaps technically most appropriate to use.

The second process that we shall consider in more detail is '*the attentional race*'. According to Shibuya & Bundesen (1988), all objects in the visual scene take a place in what one could think of as a race to become encoded. In Shibuya & Bundesen's race model, the 'odds' that a given object is selected as a winner in the race is directly related to the rate value with which the object participates. It is worth noting that the race is a stochastic, rather than a deterministic process, meaning that no one can beforehand predict readily which objects will win the race.

The third and last process that we shall consider is that of '*storage*' of object representation in VSTM. Inspired by (Usher & Cohen, 1999) we propose a competitive neural network model of VSTM, directly linking with several important assumptions expressed in Bundesen's Theory of Visual Attention (Bundesen, 1990).

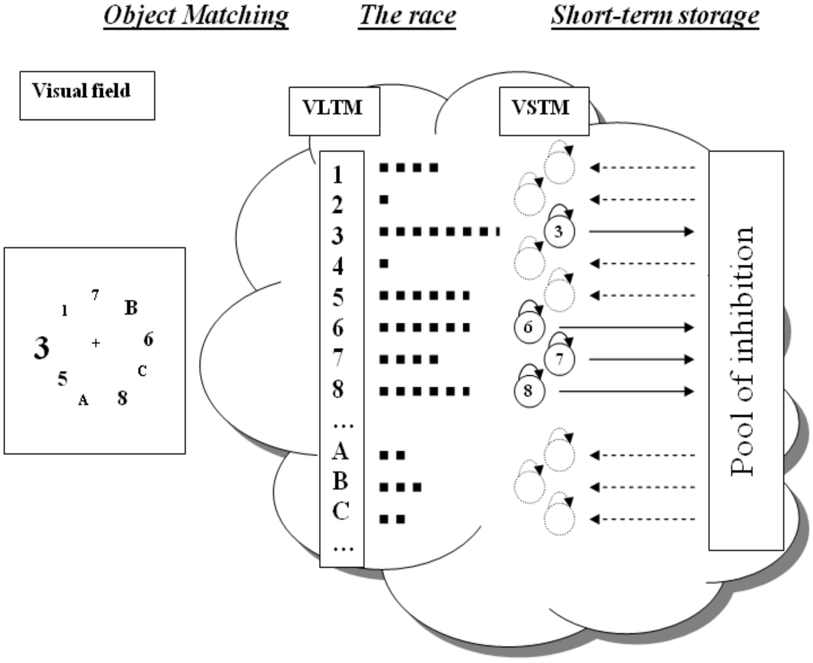


Figure 1: The Model Scheme – a partial report example. The task is to report the targets, i.e. digits and ignore the distractors, i.e. letters. The model predicts how visual elements participate in a race, where the winners become selected to be encoded in visual-short-term memory. Generally targets are processed faster than distractors, however we also see that in the example *homogeneity* is not assured, i.e. the targets (and distractors) are not of equal size (could also be contrast, letter type etc.) and therefore in the example they are illustrated as being processed with slightly different rates.

The neural theory of visual attention

The theory of visual attention (TVA) proposed by Bundesen (1990) is a unified theory of visual recognition and attentional selection. TVA provides a mathematical framework describing how the visual system is able to select individual objects in the visual field *S*, based on the visual evidence, η and the setting of two different types of visual preference parameters (pertinence, π and bias, β), representing the influence from higher cortical areas, including VLTM.

The output of the TVA-model is a set of rate parameters *v* that are directly related to the probability that a given characterization, *object x belongs to category i,* is encoded into the VSTM. The rate parameters are given by:

$$v(x,i) = \eta(x,i)\beta_i \frac{w_x}{\sum_{z\in S} w_z} \tag{1}$$

Where the attentional w_x weight of object *x* is:

$$w_x = \sum_{j\in R} \eta(x,j)\pi_j \tag{2}$$

Here $\eta(x,i)$ is defined as the strength of the sensory evidence that object *x* belongs to the visual category *i*. The pertinence of the visual category *j* is denoted by π_j and setting of these values effectively implements the so-called filtering mechanism. The perceptual decision bias of a visual category *i* is denoted by β_i and setting of these values conversely implements a complementary mechanism called pigeonholing.

The filtering mechanism increases the likelihood that elements belonging to a target category are perceived, without biasing perception in favor of perceiving the elements as belonging to any particular category.

Pigeonholing, conversely changes the probability that a particular category *i* is selected without affecting the conditional probability that element *x* is selected given that category *i* is selected.

A neural interpretation of TVA is given in (NTVA, Bundesen, Habekost, & Kyllingsbæk, 2005). Basically here pigeonholing (selection of features) is considered an increase in the rate of firing of neurons while filtering (selection of objects) is considered an increased mobilization of neurons.

Corresponding to the interpretation in NTVA the fraction $w_x/\sum w_z$ in equation (1), which is the relative attentional weight of object *x* compared to the weight of all objects *z* in the visual field *S*, can be directly interpreted as the relative fraction of neurons allocated to process a given object *x*, compared to the total number of neurons processing just any object z belonging to the visual field *S*.

Each and every encoding generally takes the form *object x belongs to category i*.

Denoting the set of all features as *R* the total processing capacity, can be considered a constant *C*, which equals the sum of all encoding rates *v*; see (Bundesen, 1990).

$$C = \sum_{x \in S} \sum_{i \in R} v(x,i) \tag{3}$$

Shibuya and Bundesen (1988) assume target as well as distractor homogeneity in their whole and partial report paradigm. This means that processing capacity is distributed equally among targets as well as among distracotors. When this is the case the rates of encoding for targets, v_T and for distractors, v_D can be calculated according to the formulas:

$$v_T = \frac{C}{T + \alpha D} \qquad v_D = \frac{\alpha C}{T + \alpha D} = \alpha\, v_T \tag{4}$$

Where *T* and *D* denote the number of targets and distractors presented, respectively. The ratio of discrimination between distractors and targets is denoted α.

The effective exposure duration τ is smaller than the actual exposure duration *t* by an amount t_0 corresponding to the temporal threshold before conscious processing begins. However the effective exposure duration can not be negative so computationally it is set to:

$$\tau = \max(0, t - t_0) \tag{5}$$

In the neural network model that we shall now describe we adopt the parameters *C*, α and t_0 and further, following Bundesen, we make use of equation (4) and equation (5).

The neural network model of VSTM

In TVA object features are encoded independently, and further there is the assumption that only one feature needs to be encoded for the object to be stored in VSTM. On the other hand; and in agreement with (Luck & Vogel, 1997), several features of the same object can be in the encoded state, and still it will only count as if one object is stored in VSTM. For this reason, and because here we are concerned about objects rather than features encoded, we simply sum over the entire number of object features, and in this way we obtain the total encoding rate v_x for object *x*:

$$v_x = \sum_{i \in R} v(x,i) \tag{6}$$

An object *x* can enter VSTM once it receives external excitation, *G* taking the shape of Poisson distributed spike trains, arriving with the rate parameter v_x. (See Figure 2).

A neural assembly that has obtained a positive level of activation will automatically seek to re-excite itself, so that it can stay in VSTM, at the same time trying to inhibit activation in other neuron assemblies representing other objects, i.e. working to suppress other object from co-temporally being stored in VSTM.

The initial condition for the simulations is that all neuron assemblies start with an activation of zero, i.e. no objects are initially stored in VSTM. As a consequence neither re-excitation nor lateral inhibition exists, before the assemblies are externally activated.

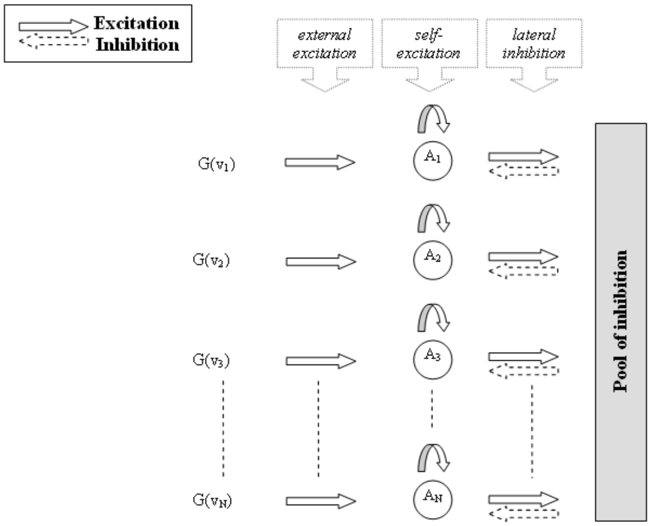


Figure 2: The neural network model of VSTM. The total number of neuron assemblies is *N* and each assembly is represented by a level of activation *A*

Implementation

The activation A_x of neuron assembly *x* (representing object *x*) is given by the first order differential equation:

$$\frac{dA_x}{dt} = -A_x + \alpha^* F(A_x) - \beta^* \sum_{z \neq x} F(A_z) + \gamma^* G(v_x) \tag{7}$$

The above equation characterizes a leaky accumulator model. There is passive decay of the activation towards the rest level, with a time constant chosen as 1, reflecting the time scale that physiologically is observed with synaptic currents (Usher & Cohen, 1999).

F is a squashing function that keeps the activation within bounds:

$$F(A) = 0, \qquad for\ A \le 0$$
$$F(A) = \frac{A}{1+A}, \qquad for\ A > 0 \tag{8}$$

As a consequence of the squashing function *F*, the parameter α^* is the limiting value of maximal self-excitation that assemblies can up-hold and the parameter β^* is the limiting maximal value of inhibition that can be sent from one assembly to another.

Also the model assumes we can not have negative self-excitation, i.e. self-inhibition and further the model does not implement any terms that could account for excitation laterally between the assemblies. The latter effect could for instance be included if one wanted to account for semantically related objects and their effect on the number of reported objects.

The attentional significance that object *i* is present in the visual field *R* is represented by the encoding rate $v_{i.}$ In our model we follow the approach from (Bundesen, 1990) and interpret this rate as the firing rate of a Poisson spike generator *G.* Hence γ^* characterizes the amplitude of the Poisson distributed input spikes arriving to the neuron assembly *x*.

The model was implemented in Matlab's Simulink toolbox. At least in the operated parameter domain we judge the stiffness of the system to be negligible so for simplicity we numerically apply Euler integration[1].

Model performance

The dataset

The data covers the performance of a single subject, participating in an extensive series of whole and partial report experiments. The subject was instructed to report targets, i.e. digits while ignoring distractors, i.e. letters displayed on an imaginary circle around a small fixation cross at the center of the screen. In practice experimental trials covered twelve whole and partial report conditions. In these the number of targets, *T* was between 2 and 6 and the number of distractors, *D* was between 0 and 6. Further, exposure durations *t* were varied systematically at 10, 20, 30, 40, 50, 70, 100, 150 and 200 ms. Each experimental condition was repeated 60 times but trials were mixed so that the subject had no a-priori knowledge of the experimental condition. Moreover trials were grouped into blocks to minimize the element of fatigue. Each presented character was immediately followed by a mask lasting for 500 ms. Further information can be found in (Shibuya & Bundesen, 1988)**.**

Performance of the neural network model

Figure 3 shows accumulated score distributions. The score is defined as the number of targets reported correctly. The upper most curve represents the accumulated score of j = 1, i.e. the probability of reporting 1 or more targets correctly. Other curves represent accumulated probabilities for reporting at least 2, 3, 4 or even 5 targets.

Shibuya and Bundesen (1988) proposed a mixture model, mixing probabilities obtained with using a statistical model that assumed memorial capacities of either $K = 3$ or $K = 4$ respectively.

There is a relatively close fit between the proposed mixture model and the empirical data. We see however that data points obtained with exposure duration around 50 ms are generally under-fitted and more noticeably the model does not account for cases where more than 4 targets are reported, as is actually the case in two out of three of the lower most plots.

What we observe with the previous model can be considered a trade-off between two conflicting demands. The first demand is to fit the initial part of the curves, i.e. the larger the processing capacity *C* the steeper the curves will rise, on the other hand the second demand, which is to keep the score distribution reasonably low for long exposure durations, require that the processing capacity *C* is not set too high. Hence the setting of *C* is set subject to a compromise.

Addressing the performance of our neural network model we think it clearly meets the standard of Shibuya and Bundesen's model. The neural model does however seem to have some trouble predicting 4 recognized items in the situations where no distractors were presented. Possibly this misfit can be diminished by running a more exhaustive optimization of model parameters. The parameters used for producing the figure were: $\alpha^* = 5$, $\beta^* = 0.1$, $\gamma^* = 2$, $C = 61.5$ Hz, $t_0 = 23$ ms and $\alpha = 0.367$. Moreover, and in contrast to Shibuya and Bundesen's model, our new model readily demonstrates its capability of predicting extreme cases, where more than 4 objects are reported.

[1] Assuming that only one spike should be allowed in each time step we must keep the integration step size sufficiently small. If the processing capacity *C* is 60 Hz, and the integration step size is kept at $dt = 0.001$, then the risk that two or more spikes will be present in a given time step is as low as 0.36 %.

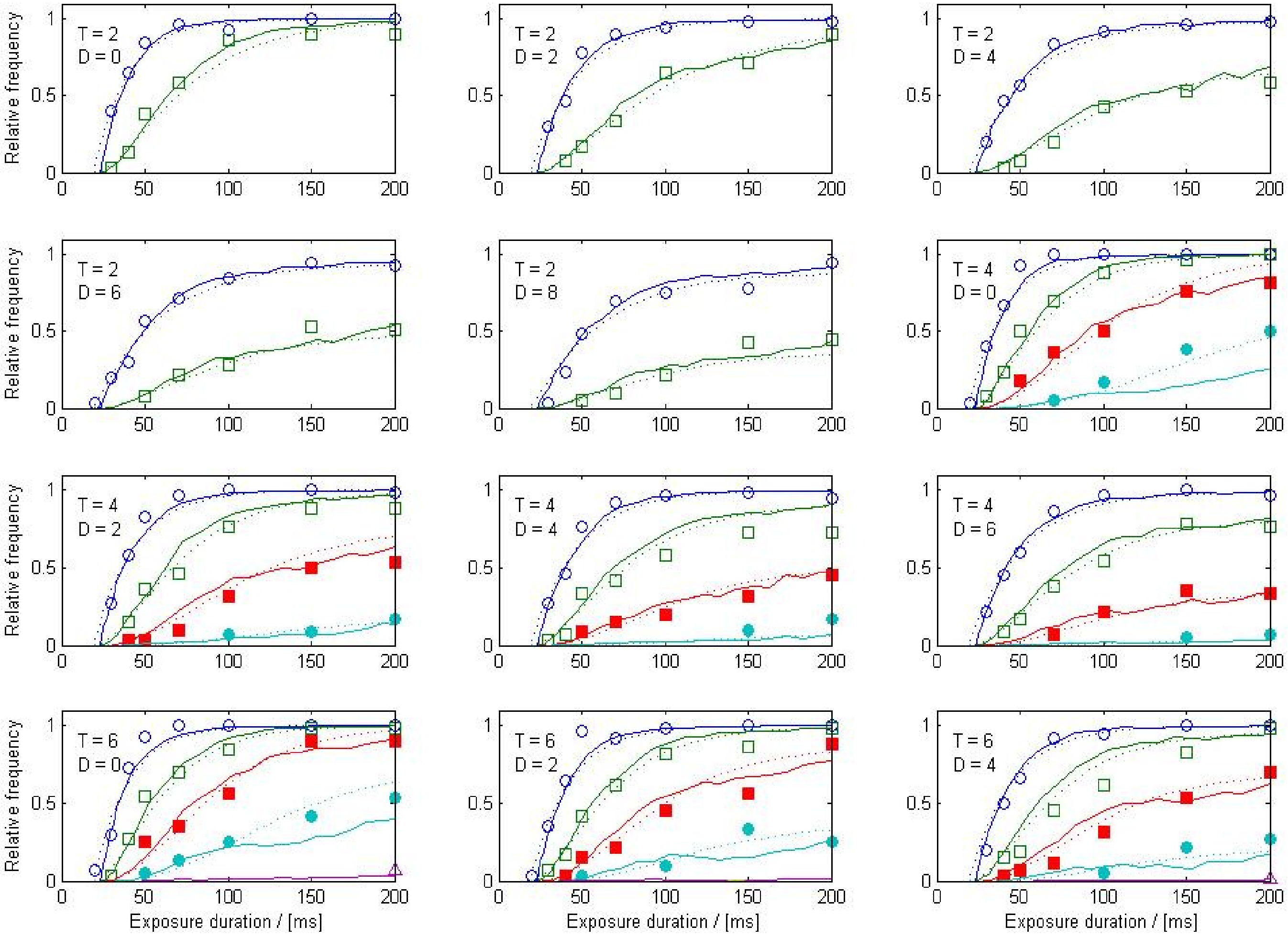


Figure 3: Accumulated score distribution for subject MP in (Shibuya & Bundesen, 1988). Probability of correctly reporting at least 1 target (blue, open circles), 2 targets (green, open squares), 3 targets (red, closed squares), 4 targets (cyan, closed circles) and 5 targets (magenta, open triangles). Empirically found values are plotted with symbols as markers. The dotted lines represent the fit by Shibuya & Bundesen (1988). Solid lines represent the performance of our neural network model. *T* and *D* denote the number of targets and distractors presented, respectively.

Discussion

This work represents an attempt to integrate the Theory of Visual Attention (Bundesen, 1990) with a simple type of winners-take-all type of network (Usher & Cohen, 1999), in the sense that the later implements a limited storage capacity of VSTM. Our new dynamic model of visual attention and VSTM is able to account for the complete set of data from whole and partial report experiments. Where the previous account by Shibuya and Bundesen (1988) treated extreme scores as outliers, the new model encompasses these as natural consequences of the internal dynamics. Further, the model explains VSTM capacity and consolidation as the result of a dynamic process rather than as a static store, which capacity is independent of processing capacity and the attentional set of the subject.

From daily life we know that humans are able to identify a very larger number of different objects. Therefore, we might think that we would have to include a neural assembly for each of these many objects candidates in our model of identification. However, what we shall argue is that our model's predictions are not affected if irrelevant neural assemblies (representing non-stimuli type of objects) are not included in the model, a useful feature which we of course make use of when we simulate with the model. The reason for this is that in the model only activated neural assemblies affect other assemblies, and so there is no lateral inhibition from inactive neural assemblies (which irrelevant assemblies tend to be) upon any other assembly. This means that adding more irrelevant assemblies generally does not affect our conclusions, except that computationally simulations become slower.

The model described gives no account of identification of individual features of an object; however it would be possible to approach this situation by having one neural assembly in the network per object feature, rather than just one neural assembly per object. In this case assemblies representing features that belonged to the same object might be modeled as having little or no lateral inhibition, ensuring that several features of the same object can be encoded without taking up additional VSTM storage space (Luck & Vogel, 1997).

Speaking of adding more neural assemblies, we ought to touch upon what it is that we think an assembly represents. Does the assembly manifest itself in one or more neurons, and how would this relate to efficient or distributed processing? The way we think about the model is that the assemblies conceptually represent different states of neural activation. As assumed, these states interact and as we have described we suppose that feedback mechanisms play an important role in keeping the activation of the assembly sustained, allowing for visual short-term memories.

A possible confound of the model is that it does not consider internal noise, which is likely to play a key role in many neural systems. A way to deal with this would be to transform the input stage (the Poisson distributed spike trains, arriving with the rate parameter v) to a stochastic diffusion process with wiener noise process included. For this to make sense the activation threshold for consciousness would have to take a higher value than the level of initial activation.

In future studies, we think it would be relevant to explore the implication of transforming the model into a stochastic differential equation as mentioned above. Because the model is temporally dependent it would also be interesting to know if it would be able to address the dynamic consolidation in VSTM found in temporally extended paradigms such as the attentional blink paradigm and studies of attentional dwell time; e.g. (Ward, Duncan, & Shapiro, 1996). Here, consolidation in VSTM is strongly dependent on competition between items already encoded into VSTM and visual items presented at a later point in time. Incorporation of such a competitive process follows naturally from the dynamic architecture of the present model.

Acknowledgments

We would like to thank Hitomi Shibuya & Claus Bundesen for access to the experimental data in (Shibuya & Bundesen, 1988). Also we would like to thank Marius Usher for kindly verifying parameters and settings in (Usher & Cohen, 1999). Further we would like to thank the anonymous reviewers for providing helpful suggestions as well as relevant comments.

References

Alvarez, G., & Cavanagh, P. (2004). The capacity of visual short-term memory is set both by visual information load and by number of objects. Psychological Science, 15(2), 106-111.

Bundesen, C. (1990). A Theory of Visual Attention. Psychological Review, 97(4), 523–547.

Bundesen, C., Habekost, T., & Kyllingsbæk, S. (2005). A Neural Theory of Visual Attention: Bridging Cognition and Neurophysiology. Psychological Review, 112(2), 291–328.

Cattell, J. M. (1886). The inertia of the eye and brain. Brain, (8), 295-312.

Cowan, N. (2000). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and Brain Sciences, 24(1), 87-185.

Luck, S., & Vogel, E. (1997). The capacity of visual working memory for features and conjunctions. Nature, 390(6657), 279-281.

Shibuya, H., & Bundesen, C. (1988). Visual Selection from Multielement Displays: Measuring and Modeling Effects of Exposure Duration. Journal of Experimental Psychology Human Perception Performance, 14(4), 591–600.

Sperling, G. (1960). The Information Available in Brief Visual Presentions. Psychological Monographs, 74(11), 1-29.

Usher, M., & Cohen, J. (1999). Short Term Memory and Selection Processes in a Frontal-Lobe Model. In Connectionist Models in Cognitive Neuroscience (pp. 78-91). Birmingham: Springer-Verlag.

Ward, R., Duncan, J., & Shapiro, K. (1996). The slow time-course of visual attention. Cognitive Psychology, 30(1), 79-109.

The Feature-Label-Order Effect In Symbolic Learning

Michael Ramscar, Daniel Yarlett, Melody Dye, & Nal Kalchbrenner
Department of Psychology, Stanford University,
Jordan Hall, Stanford, CA 94305.

Abstract

We present a formal analysis of symbolic learning that predicts significant differences in symbolic learning depending on the sequencing of semantic features and labels. A computational simulation confirms the Feature-Label-Ordering (FLO) effect in learning that our analysis predicts. Discrimination learning is facilitated when semantic features predict labels, but not when labels predict semantic features. A behavioral study confirms the predictions of the simulation. Our results and analysis suggest that the semantic categories people use to understand and communicate about the world might only be learnable when labels are predicted from objects.

Introduction

The ways in which symbolic knowledge is learned and represented in the mind are poorly understood. We present an analysis of symbolic learning—in particular, word learning—in terms of error-monitoring learning, and consider two possible ways in which symbols might be learned: learning to predict a label from the features of objects and events in the world; or learning to predict features from a label. This analysis predicts significant differences in symbolic learning depending on the sequencing of objects and labels, confirmed in computational simulations and an empirical study. Discrimination learning is facilitated when semantic features predict labels, but not when labels predict semantic features. We call this the Feature-Label-Ordering (FLO) effect. Our results and analysis suggest that the semantic categories people use to understand and communicate about the world can only be learned if labels are predicted from objects.

Learning

Learning is best conceived of as the process of acquiring probabilistic information about the relationships between important regularities in the environment (such as objects or events) and the cues that enable their prediction (Rescorla & Wagner, 1972). The learning process is driven by discrepancies between what is expected given a cue, and what is actually observed in experience (*error-driven learning*). The predictive value of a cues are strengthened when events are under-predicted, and weakened when they are over-predicted (Kamin, 1969; Rescorla & Wagner, 1972). As a result, cues compete for relevance, and the outcome of this competition is shaped both by positive evidence about co-occurrences between cues and predicted events, and negative evidence about non-occurrences of predicted events. This process produces patterns of learning that are very different from what would be expected if learning were shaped by positive evidence alone (a common portrayal of Pavlovian conditioning, Rescorla, 1988).

Symbolic learning

This view of learning can be applied to symbolic thought by thinking of symbols (i.e., words) as both potentially important cues (predictors) and outcomes (things to be predicted). For example, the word "chair" might be predicted by, or serve to predict, the features that are associated with the things we call chairs (both when chairs and "chair" are present as perceptual stimuli, or when they are being thought of in mind)

Word learning can thus take two forms, in which either:

(i) the cues are labels and the outcomes are features
(ii) the cues are features and the outcomes are labels.

In (i), which we term *LF-learning*, information allowing the prediction of a feature or set of features given a label is acquired, whereas in (ii), which we term *FL-learning*, information allowing the prediction of a label from a given feature or set of features is acquired. Since formal learning models are fundamentally relational (see e.g., Rescorla, 1988), LF- and FL-learning describe the two possible ways that the relations between labels and "meanings" can be structured in symbolic learning.

In FL learning, the set of cues being learned from is generally larger than the set of outcomes being learned about, whereas in FL learning, the set of outcomes is generally larger than the set of cues. As we will now show, these set-size differences in the number of cues and outcomes that are being learned about in each these two forms of word learning result in different levels of discrimination learning.

The structure of labels and the world

Symbolic labels are relatively discrete, and possess little cue-structure, whereas objects and events in the world are far less discrete, and possess much denser cue-structure. (By cue-structure we mean the number of salient and discriminable cues they simultaneously present.) Consider a situation in which say, a *pan* is encountered in the environment. A pan presents to a learner many discriminable features; shape, color, size, etc. In contrast, consider the label 'pan.' A native English speaker can parse this word into a sequence of phonemes [p^h *an*], but will otherwise be largely unable to discriminate many further features within these. While there are other discriminable aspects of speech (e.g., emphasis, volume, or pitch contour), ordinarily, the phonetic level dominates semantic categorization. Other features, such as pitch contour, do not *compete* with phonemes in the same way that color might vie for relevance with shape in an object. Further, because phonemes occur in a sequence rather than simultaneously, there can be little to no direct competition between them as cues. Labels thus provide learners with little competitive cue-structure.

The difference in cue-structure in turn affects the formal properties of the two forms of learning we described above. In LF-learning, because labels serve as cues and since individual labels have little cue-structure, learning involves predicting a set of features (the semantic features of objects and events) from a single cue (the label). Thus, essentially, LF-learning has a one-to-many form: one cue to many features.

In contrast, FL-learning involves predicting a single response (a label) from a larger set of cues (the features of an event or object). FL-learning has a many-to-one form: from many semantic features to a label.

Cue-competition in learning

Where many cues are presented simultaneously, they can compete for relevance in the prediction of a particular event. If a cue successfully predicts an event over time (positive evidence), the associative strength between the cue and the event will increase. Conversely, when a cue unsuccessfully predicts a given event—i.e., the event does not follow the cue (negative evidence), the associative strength between the cue and the response will decrease.

In one-to-many LF-learning, a single cue will be predictive of each of the many features encountered in an object or event. Because no other cues are available to compete for associative value, there can be no loss of potential associative value to other cues over the course of learning trials. By contrast, in many-to-one FL-learning, because many cues are available to compete for relevance, learning will separate the highly salient cues from the less salient cues, favoring cues with a high degree of positive evidence and disfavoring those with a high degree of negative evidence. FL-learning and LF-learning thus differ significantly in terms of cue-competition; the dense cue-structure of FL-learning fosters cue-competition, while the sparse cue-structure of LF-learning inhibits it.

Cue-structure and symbolic learning

To see how these factors affect symbolic learning, consider a simplified environment in which there are two kinds of objects: wugs and nizes. These objects have two salient features: their shape and their color. Wugs are wug-shaped and can be either blue or red. Likewise, nizes are niz-shaped and can be either blue or red. Suppose now that one is learning what wugs and nizes are under FL-learning conditions. Figure 1 represents FL-learning in this simplified environment:

At (i), a learner encounters an object with two salient features, shape-1 and red, and then hears the label 'wug'. The learner acquires information about two equally predictive relations, shape-⇒'wug' and red⇒'wug'. At (ii), the learner two new cues and a new label, and forms two new equally weighted predictive relations, shape-2⇒'niz' and blue⇒'niz'. Then at (iii), the learner encounters two previously seen cues, shape-1 and blue.

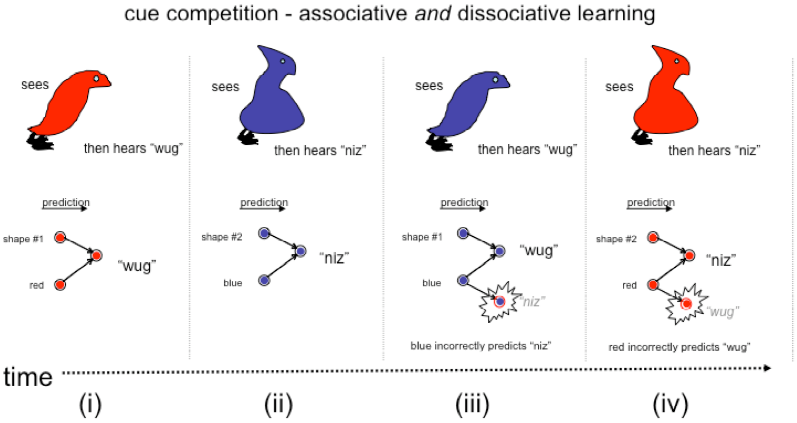


Figure 1. Cue competition in learning. The top panels depict the temporal sequence of events: an object is shown and then a word is heard over three trials. The lower panels depict the relationship between the various cues and labels in word learning.

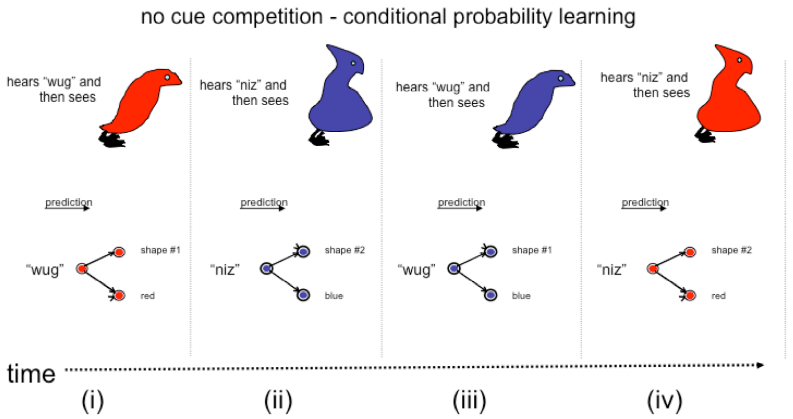

no cue competition - conditional probability learning
hears "wug" and then sees
hears "niz" and then sees
hears "wug" and then sees
hears "niz" and then sees
prediction
prediction
prediction
prediction
"wug"
shape #1
red
"niz"
shape #2
blue
"wug"
shape #1
blue
"niz"
shape #2
red
time
(i)
(ii)
(iii)
(iv)

Figure 2. When **labels predict features,** the absence of cue competition results a situation where the outcome of learning is simply be a representation of the probability of the features given the label.

Given what the learner already knows—i.e., shape-1⇒'wug' and blue⇒'niz'—she expects 'wug' and 'niz.' Only 'wug' occurs. As a result: (1) given positive evidence of 'wug', the associative value of the relation shape-1⇒'wug' increases; but importantly (2) negative evidence about the non-occurrence of 'niz' causes blue⇒'niz' to lose associative value. Crucially, as the value of blue⇒'niz' decreases, it's value *relative* to shape-2⇒ 'niz' changes. At (iv), a similar situation occurs. The learner encounters shape-2 and red and expects 'niz' and 'wug'. Only 'niz' is heard, so the associative value of shape-2⇒'niz' increases, while red⇒'wug' loses associative value.

FL-learning is *competitive*: as a cue loses associative value, its value *relative* to other cues may change. This can *shift* associative value from one cue to another.

Now consider LF-learning in a similar scenario (Figure 2). At (i), a learner encounters the label 'wug' and then an object with the two salient features, shape-1 and red. She thus learns about two equally valuable predictive relations 'wug' ⇒shape-1 and 'wug'⇒red. Similarly, at (ii), the learner acquires two further equally valued relations 'niz'⇒shape-2 and 'niz'⇒blue. Now, at (iii), the learner hears 'wug' and expects red and shape-1. However, shape-1 occurs and blue occurs. This has three consequences: (1) an increase in the associative value of 'wug'⇒shape-1; (2) 'wug'⇒blue becomes a new predictive relation; (3) negative evidence decreases the value of 'wug'⇒red. However, since 'wug' is the only cue, this loss of associative value is *not* relative to any other cues (likewise at iv). LF-learning is thus *non-competitive*, and simply results in the learning of the probabilities of events occurring given cues.

The Feature-Label-Order Hypothesis

Both FL and LF-learning capture probabilistic information predictive relationships in the environment. However, there are fundamental differences between the two. In FL-learning predictive power, not frequency or simple probability, determines cue values; LF-learning is probabilistic in far more simple terms. Given this, it seems that the sequencing of labels and features ought to have a marked affect on learning. **We call this the Feature-Label-Order hypothesis**.

We formally tested the FLO hypothesis in simulations using a prominent error-driven learning model (Rescorla &Wagner, 1972; see also; Allen and Siegel, 1996). We should note that the analysis of symbolic learning described here could be implemented in a number of other models (e.g., Pearce & Hall, 1980; Rumelhart, Hinton & McClelland, 1986; Barlow, 2001) and applied to learning other environmental regularities.

The Rescorla-Wagner model formally states how the associative values (*V*) of a set of cues *i* predicting an event *j* change as a result of learning in discrete training trials, where *n* indexes the current trial.

Equation (1) is a discrepancy function that describes the amount of learning that will occur on a given trial; i.e., the change in associative strength between a set of cues *i* and some event *j*:[1]

$$\Delta V_{ij}^{n} = \alpha_i \beta_j (\lambda_j - V_{TOTAL}) \qquad (1)$$

If there is a discrepancy between λ_j (the total possible associative value of an event) and V_{TOTAL} (the sum of current cue values), the saliency of the set of cues α and the learning rate of the event β will be multiplied against that discrepancy. The resulting amount will then be added or subtracted from the associative strength of any cues present on that trial.

The associative strength between a set of cues *i* and an event *j* will increase in a negatively accelerated fashion over time, as learning gradually reduces the discrepancy between what is predicted and what is

[1] V_{ij} is the change in associative strength on a learning trial *n*. α denotes the saliency of *i*, and β the learning rate for *j*.

observed. Given an appropriate learning-rate, learning asymptotes at a level that minimizes the sum-of-squares prediction error for a set of observed cues to an event.

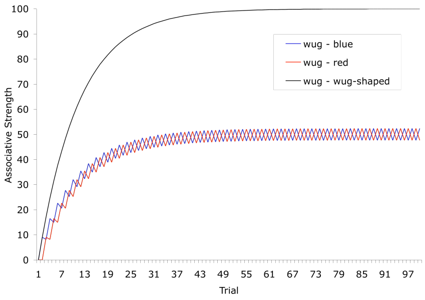


Figure 3. The development of cue values in a simulation of the LF-learning scenario depicted in **Figure 2**.

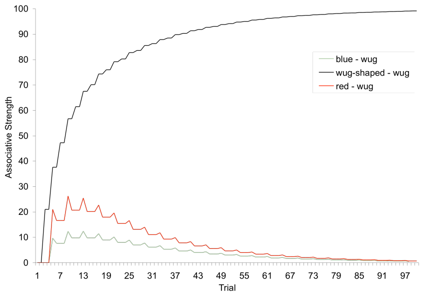


Figure 4. The development of cue values in a simulation of the FL-learning scenario depicted in **Figure 1.**

Discrimination and interference

Two computational simulations (in the Rescorla & Wagner, 1972 model, described below)[2] formally illustrate the differences in the representations of what gets learned in LF and FL-learning. As Figure 3 shows, LF-learning simply results in a representation of the probability of each feature given the label; e.g., the learned associative value of 'wug'⇒red is about half of the associative strength of 'wug'⇒wug-shaped, because 'wug' predicts red successfully only 50% of the times and wug-shaped successfully 100% of the time. In FL-learning (Figure 4), the representations learned reflect the *value* of cues: the associative relationship 'wug'⇒wug-shaped is very reliable, and is highly valued relative to cues that generate prediction error. In this case the association 'wug'⇒red is effectively unlearned.

It is important to note that in LF-learning, the lack of discrimination produced by learning can lead to problems of interference in predicting events (or responses to them). LF-learning tends to produce representations in which a number of competing predictions are all highly probable.

In our earlier wug / niz example there were equal numbers of wugs and nizzes: *red* cued "wug" 50% of the time and "niz" 50% of the time. Thus if a child trained LF on the animals saw a red wug and was asked what it was called, there is 100% probability that wug-shaped=wug and only 50% probability that red=niz. 'Wug,' is the obvious answer. Imagine, however, there were 20 times as many blue wugs as blue nizzes in the population, and 20 times as many red nizzes as red wugs. In this scenario, the color red will cue "wug" about 95% of the time and "niz" only about 5% of the time based on frequency of occurrence. For a child trying to name a red wug, there's again a near 100% probability that wug-shaped=wug, but now there's also a 95% probability that red=niz. There will be a large degree of uncertainty about the right answer. Tracking the frequencies of successful predictions will not pick out the cues that best discriminate one prediction from others, leading to *response interference*. While FL- and LF-learning can discriminate responses in an ideal world, LF-learning will fail to discriminate events (or responses) when frequencies vary (and in the actual world, frequencies will vary).

		Non discriminating features			Discriminating features					
		1	2	3	1	2	3	4	5	6
Category 1	75%	1	0	0	1	0	0	0	0	0
	25%	0	1	0	0	1	0	0	0	0
Category 2	75%	0	1	0	0	0	1	0	0	0
	25%	0	0	1	0	0	0	1	0	0
Category 3	75%	0	0	1	0	0	0	0	1	0
	25%	1	0	0	0	0	0	0	0	1

Figure 5: The abstract representations of the category structures used to train the Rescorla-Wagner models

Simulating interference

To illustrate the problem of response interference, we simulated category learning in the Rescorla-Wagner model using abstract representations of the category structures in Figure 5. The training set comprised 3 category labels and 9 exemplar features (3 non-discriminating features that were shared between exemplars belonging to different categories, and 6 discriminating features that were not shared with members of another category). The frequency of the sub-categories was manipulated so that each labeled category drew 75% of its exemplars from one sub-category and 25% of its exemplars from another subcategory. The two sub-categories that made up each labeled category did not share any features, such that learning to correctly classify one of the sub-categories paired with each label would provide no assistance with learning the other sub-category paired with that label. Finally, each low frequency sub-category shared its non-discriminating feature with the high frequency exemplars of a different labeled category. This

[2] The simulations assume either a *niz* or a *wug* is encountered in each trial, that each species and color is equally frequent in the environment, and that color and shape are equally salient.

manipulation was designed to create a bias towards the misclassification of the low-frequency exemplars. Learning to correctly classify low frequency exemplars necessarily required learning to weigh the discriminating feature more than the non-discriminating feature, despite its lower overall input frequency.

Two simulations were configured to created two networks of feature and label relationships. The first network learned associative weights from the 9 exemplar features (serving as cues) to the 3 labels (serving as events; "FL training"), while in the second case the network learned from the 3 labels (serving as cues) to the 9 features (serving as events; LF training). Each category had a high frequency exemplar, presented on 75% of the training trials for that category, and a low frequency exemplar (occurring 25% of the time). On each training trial a label and appropriate exemplar pattern were selected randomly to train each of the two networks. Training comprised 5000 trials, which allowed learning to reach asymptote. The model has several parameters that affect learning. For simplicity, the simulations assumed equally salient cues and events (α=0.01 for all i;●=0.01 for all j) and equal maximum associative strengths (= 1.0).

To test the FL-network, exemplar features were activated to determine the subsequent activation of the labels. Propagating these values across the weights learned by the network then determined the associative values that had been learned for each label given those features. Luce's Choice Axiom (Luce, 1959) was used to derive choice probabilities for the 3 labels given these activations, revealing that the FL-trained network categorized and discriminated well (the probability of correct classification for the low and the high frequency exemplars was p=1).

LF-network testing involved activating the labels in order to determine subsequent activation of the features. In turn, each label was given an input value of 1, and this then produced activation levels in the features, which were determined by the associative values learned in training. In order to assess the network's performance, the Euclidean distance between the predicted activations and the actual feature activations of the appropriate exemplar were calculated. For each label there were two sets of feature activations: those corresponding to the high and low frequency exemplars. To test learning of both exemplar types, a category and a frequency (either high or low) were selected, and the difference between the feature activations predicted by the network and the correct values for the category exemplars was computed. These differences were then converted to z-scores, and from these the probabilities of selecting the correct exemplar given the category label were calculated as follows:

$$P(x) = \exp(-z(dist(x,t))) \tag{2}$$

where P(x) is the likelihood of the network selecting exemplar x, z(·) returns the z-score of its argument relative to its population, dist(·,·) is the Euclidean distance function, and t is the exemplar pattern generated by the network. The P(x) likelihoods were normalized using Luce's Choice Axiom to yield normalized probability estimates. These revealed that the LF network performed poorly. At asymptote, it predicted the correct feature pattern with only p=.35 confidence for low frequency exemplars (chance), and p=.75 confidence for high frequency exemplars.

Testing the FLO Hypothesis

Consistent with our hypothesis, a notable Feature-Label-Order Effect was detectable in the simulations. The following experiment was designed to see whether human learning would show a similar effect.

Participants

32 Stanford Undergraduates participated for credit.

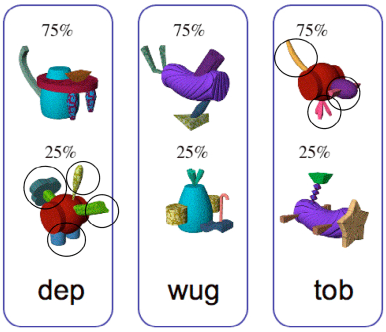


Figure 6. The category structures Experiment 1. (The stimuli are fribbles created by Michael Tarr's lab at Brown University.) The features that need to be weighted to successfully distinguish the sub-categories are circled on the low-frequency "dep" and high-frequency "tob" exemplars.

Method and Materials

Three experimental categories of "fribbles" were constructed, each comprising two sub-categories clustered around a non-discriminating feature and a set of discriminating features. The two sub-categories that made up each labeled category did not share features, and so learning to correctly classify one of the sub-categories paired with each label provided no assistance with learning the other sub-category paired with that label. The sub-categories were again manipulated so that 75% of the exemplars of a category belonged to one sub-category, and 25% to another, and each non-discriminating feature was shared by high frequency and low frequency exemplars that belonged to different categories. Thus learning to correctly classify low frequency exemplars necessarily required learning to weigh the discriminating feature more than the non-discriminating feature. A control category served to check that there were no differences in learning between the two groups other than those we hypothesized: all its exemplars shared just one, highly salient feature (all were blue). Because learning this category involved a binary pairing blue⇒bim, there was no "predictive structure" to discover. In the absence of competing exemplars, learning was predicted to be identical for FL and LF training.

To enforce LF or FL relationships as our participants studied "species of aliens" we minimized their ability to strategize (world learning is rarely a conscious process. All four categories were trained simultaneously, exemplars of each category were presented in a non-predictable sequence, and each exemplar was presented for only 175ms to inhibit participants' ability to search for features. FL training trials comprised 1000ms presentation of a label ("this is a wug"), followed by a blank screen for 150 ms, followed by 175ms exposure to the exemplar. LF training trials comprised 175 ms exemplar, 150 ms blank screen and 1000ms label ("that was a wug"). A 1000ms blank screen separated all trials (see Figure 10). A training block comprised 20 different exemplars of each experimental category – 15 high-frequency exemplars and 5 low-frequency exemplars – and 15 control category exemplars. Training comprised 2 identical blocks, with a short rest between the blocks.

Testing consisted of speeded 4 alternative forced-choice tasks. Half the participants matched an exemplar to the 4 category labels, and half matched a label to 4 previously exemplars drawn from each category. Participants were instructed to respond as quickly as they could (after 3500ms, a buzzer sounded and no response was recorded). Each sub-category (and the control) was tested 8 times, yielding 56 test trials.

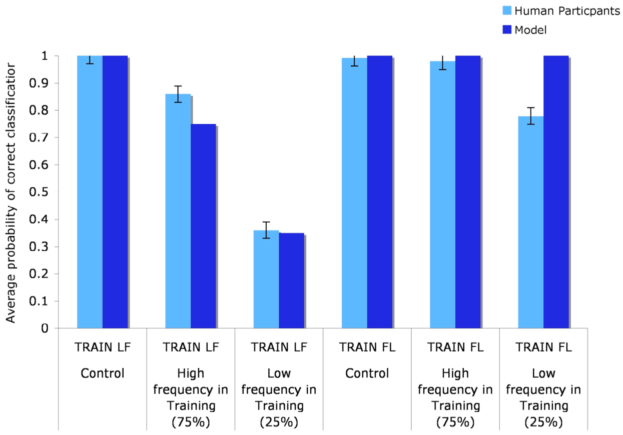


Figure 7: The predictions of the simulation plotted against the performance of participants in Experiment 1.

Results and discussion

The results of the experiment were remarkably consistent with our predictions; a 2 x 2 ANOVA revealed a significant interaction between exemplar-frequency and training (F(1,94)=20.187, p<0.001; Figure 6). The FL-trained participants classified high and low frequency items accurately (FL high p=.98; low p=.78), while the LF-trained participants only accurately classified high-frequency items (p=.86) and failed to classify the low frequency exemplars above chance levels (p=.36, t(47)=0.536, p>0.5). The control category was learned to ceiling in both conditions. Analyses of confusability (i.e., the rates at which exemplars were misclassified to the category with which they shared non-discriminating features) showed the same interaction between frequency and training (F(1,94)=8.335, p<0.005), with higher confusion rates after LF training (M=22.6%) than FL (M=6%; t(16)=5.23, p<0.0001). These differences were not due to a speed / accuracy trade-off; participants trained FL were faster as well as more accurate (LF M=2332ms, FL M=2181ms; t(190)=1.677, p<0.1).

To the degree that learning relational, and driven by prediction error (and there is considerable evidence that it is), LF- and FL-learning describe the two possible ways the relations between labels and "meanings" can be structured in learning. The Feature-Label-Ordering effect may thus be an inevitable aspect of symbolic learning. We believe this has many implications for our understanding of language and cognition.

Acknowledgments

This material is based upon work supported by NSF Grant Nos. 0547775 and 0624345 to Michael Ramscar

References

Barlow H. (2001). Redundancy reduction revisited, *Network*, **12**, 241-253

Kamin L.J. (1969). Predictability, surprise, attention, and conditioning. In: Campbell B, Church R (eds). *Punishment and Aversive Behaviour*. Appleton-Century-Crofts: New York.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical Analysis*. New York: Wiley

Pearce J.M. & Hall G. (1980) A model for Pavlovian learning. *Psychological Review*, 87:532-552

Rescorla R.A. and Wagner A.R. (1972). A Theory of Pavlovian Conditioning: Variations in the Effectiveness of Reinforcement and Nonreinforcement. In Black & Prokasy (Eds.), *Classical Conditioning II: Current Research and Theory*. New York: Appleton-Century-Crofts.

Rescorla R.A. (1988). Pavlovian Conditioning: It's Not What You Think It Is, *American Psychologist*, **43**(3), 151-160

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. A framework for Parallel Distributed Processing. in Rumelhart, McClelland, & the PDP research group. (1986). *Parallel distributed processing*. Volume I. Cambridge, MA: MIT Press

Siegel, S.G., & and Allan, L.G. (1996). The widespread influence of the Rescorla-Wagner model, *Psychonomic Bulletin and Review*, 3(3), 314-321

Learning on the fly
Computational analyses of an unsupervised online-learning effect in artificial grammar learning

Martin Rohrmeier (mrohrmeier@cantab.net)
Centre for Music & Science, Faculty of Music, University of Cambridge

Abstract

Humans rapidly learn complex structures in many domains. Some findings of above-chance performance of untrained control groups in artificial grammar learning studies raise the question to which extent learning can occur in an untrained, unsupervised testing situation with partially correct and incorrect structures. Computational modelling simulations explore whether an unsupervised online learning effect is theoretically plausible in artificial grammar learning. Symbolic n-gram models and simple recurrent network models were evaluated using a large free parameter space and applying a novel evaluation framework, which models the human experimental situation through alternating evaluation (in terms of forced binary grammaticality judgments) and subsequent learning of the same stimulus. Results indicate a strong online learning effect for n-gram models and a weaker effect for simple recurrent network models. Model performance improves slightly once the window of accessible past responses for the grammaticality decision process is limited. Results suggest that online learning is possible when ungrammatical structures share grammatical chunks to a large extent. Associative chunk strength for grammatical and ungrammatical sequences is found to predict both, chance and above-chance performance for human and computational data.

Keywords: Unsupervised learning; online learning; computational modelling; artificial grammar learning; n-gram model; neural network; artificial grammar learning

Introduction

Humans are very efficient learners. In many cases we learn without intention and without awareness, and it has been suggested that implicit learning constitutes one powerful and fundamental root mechanism of learning (Reber, 1993). Humans are even further able to learn and to adapt to the environment, *whilst being in the midst of things*: we pick up individual characteristics, or melodic features in a piece of music while we are listening or dancing to it, sportsmen are able to adapt to characteristics of their opponents or the environment while playing, or musicians adapt to characteristic musical patterns of other musicians while improvising together.

Humans acquire implicit knowledge about regular structures very quickly. Serial reaction time experiments have found humans to be able to acquire rule-based structures extremely rapidly (Reber, 1993). Similarly, under the artificial grammar learning paradigm (AGL) participants acquire rule-based structures rapidly after short familiarisation periods (Pothos, 2007). One question that arises in this context concerns how efficient humans may learn regular structures even during a test, or under more complex conditions involving a combination of both, regular and irregular structures. For instance, Dulany et al. (1984) found that untrained controls performed above chance, which might suggest that they have picked up some regularity in the structures during the testing. Redington & Chater (1996) discuss the possibility of such a learning process, whereas Reber & Perruchet (2003) argue that above chance performance of a control group would not stem from a learning effect but from confounding structural biases that may be easy to detect. However, two recent musical grammar learning experiments found a high performance of about 60% in untrained controls (Loui et al, 2008; Rohrmeier et al., submitted) which may reopen the question about a potential rapid online-learning effect.

This study addresses how online-learning on the fly could be theoretically possible based on computational modelling methods. It proposes a framework to model both the simultaneous learning of structures while being tested and the generation of binary grammaticality judgments, in a way that parallels the human situation. It aims to demonstrate that two standard computer models of learning reproduce an effect of unsupervised online learning under certain conditions regarding the stimulus structures. Further it explores why it turns out that grammatical structures, but not ungrammatical ones, are preferred as familiar even though the learning process happens under unsupervised conditions. These theoretical and computational observations raise several hypotheses regarding an efficient online-learning effect for future psychological research.

Experimental hints & evidence

In a musical AGL experiment, Rohrmeier et al. (submitted) found that untrained control participants were able to distinguish rule-consistent grammatical stimulus structures from ungrammatical structures throughout the course of a testing phase, even though they had no prior training. Once the performance of this group is plotted over time (throughout the course of the testing phase, in which the stimulus order was randomized), one finds a curve of the shape of a saturation curve (figure 1). The fact that the performance curve begins at a chance level of 0.5 (and not above) and steadily raises to a level of 0.62, suggests that participants gradually pick up some knowledge that enables them to distinguish the structures, with little prior bias. The study found the group performance to be significantly above chance after 11 steps into the testing phase.

This unusual result is surprising and rare in the context of other AGL studies. However there are not many cases of studies with untrained control groups. Dulany et al. (1984), Redington & Chater (1994), Dienes (reported in Redington

& Chater, 1996), and Loui et al. (2008) found above chance performance of untrained controls; whereas Altmann et al. (1995), Meulemans & Van der Linden (1997) and Reber & Perruchet (2003) did not. If well this set of experimental evidence is not decisive and further empirical work is required, computational modelling work may shed light on the question of whether an effect of rapid online-learning under complex conditions of partially grammatical and ungrammatical structures is theoretically plausible at all. In addition it may raise particular hypotheses regarding human learning performance based on theoretical considerations.

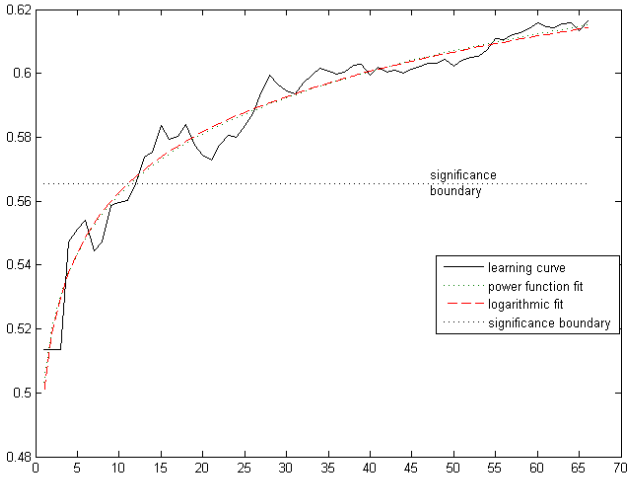


Figure 1: Performance of an untrained participant group during the testing phase.

Based on these considerations, this study aims to simulate a potential effect of online-learning from the angle of two different cognitively motivated models: a connectionist model with reference to connectionist theories of AGL (Pothos, 2007) and a symbolic n-gram model with reference to fragment or chunking based theories of human learning (Servan-Schreiber & Anderson, 1990; Perruchet & Pacteau, 1990).

Method

First, the modelling framework intends to model the simultaneity of learning and responding during testing. This departs from traditional machine learning or computational modelling methods (Mitchell, 1997; Bishop, 2006) as the typical separation between model training and model evaluation is suspended. In this framework the models are first evaluated for each given stimulus and then subsequently trained on the same stimulus. This method keeps the modular operations of training and evaluating the model with single strings (as learning during the processing of the stimulus would require significant changes in the mechanism of the model, in particular, the SRN).

Secondly, the modelling framework intends to capture the human testing situation, which involves having to decide about stimulus grammaticality immediately during the testing. Often computational models are simply evaluated by comparing the overall sequence familiarity for grammatical and ungrammatical sequences after the whole test evaluation (e.g. Kuhn & Dienes, 2008) but are not required like the human to give decisive binary grammaticality (G/UG) responses after each single stimulus without full information about the remaining test set. Consequently, the model responses would not be directly comparable to the human responses. Therefore, the present modelling framework applies a threshold decision technique to generate binary grammaticality judgments from the model's familiarity responses directly for each single stimulus (see below).

We use cross-entropy based on sequence predictability (Mitchell, 1997; Bishop, 2006; Pearce & Wiggins, 2004) as an estimate of the familiarity that a model assigns to a stimulus.

Models

N-gram model. Fragment based n-gram models are symbolic models which have been successfully used in computational linguistics and in music modelling (Manning & Schuetze, 1999; Pearce & Wiggins, 2004, 2006). This study employs a simple n-gram model after Pearce & Wiggins (2004) which stores fragments of the lengths 1 to n symbols from its input sequences, and creates predictions for the symbol sequence of a given test sequence by combining predictions from differently sized fragments using Moffat's (1990) method, which has been found to perform best in comparison to other smoothing and combining methods (Pearce & Wiggins, 2004). The model produces a familiarity response for a whole test sequence based on its information content, i.e. the mean cross-entropy of the prediction for each symbol of the sequence.

Simple Recurrent Network. The simple recurrent network model was implemented following Elman (1990). A familiarity response for a single test sequence is generated through the information content, i.e. cross-entropy based on the prediction of each symbol.

Deciding grammaticality judgements

Both models return familiarity values based on cross-entropy, which have to be classified on the fly into binary grammaticality responses. As the range and distribution of the familiarity values are unknown prior to the test and vary over time, the decision cannot be based on a static threshold value. The current familiarity value is instead classified as grammatical or ungrammatical when it is greater or smaller than the median of the available past familiarity values. The decision is made random for the first sequence as there is no reference value available.

Procedure

First the model is initialised and the sequence order is randomised. Then, for each stimulus of the testing set, the model computes, as outlined, a familiarity response based on cross entropy, which is compared to the median of the past responses and subsequently transformed into a grammaticality judgment. After each sequence evaluation, the model is trained with the stimulus.

Choice of free parameter space

Cleeremans & Dienes (2008) discuss the problem that regarding the choice of free model parameters there are few ways of determining cognitively meaningful parameter choices. The present simulations adopt the method by Kuhn & Dienes (2008) to define a grid over the range of possible meaningful parameters and to run a fixed number of simulations for each point in the parameter space. A parameter space of learning rate and momentum each of {0.1, 0.3, 0.5, 0.7, 0.9}, 2 learning epochs, and {10, 15, 25, 50, 80, 120} hidden units was used for the SRN models, resulting in a space of 150 parameter combinations. The n-gram models were evaluated using a parameter space of a maximal n-gram length of {2,3,4,5,6,∞}, where ∞ signifies that there was no upper limit for the fragment size and that fragments up to the whole string were stored.

Materials

Test sequences from the studies above which featured an untrained control group were used, if the stimuli were available. In addition, the stimuli by Brooks & Vokey (1991) as used by Tunney & Shanks (2003) were included in order to feature another well-known finite-state grammar.

Simulation 1

The purpose of simulation 1 was to investigate to which extent online learning could be simulated for the studies listed above. For each of the 7 grammars listed above, 80 instances of each the n-gram model and the SRN were run for each configuration in the parameter space above. In addition, the same number of control models were run, which featured no sequence training after stimulus presentations.

Table 1 displays the results. All n-gram models exhibit a significant and strong effect of online-learning for all parameters (all $p<0.0005$). In many cases mere bigram learning proves sufficient for a performance level which is barely topped by larger contexts, a finding that is consistent with evaluations by Pearce & Wiggins (2004). Further, many n-gram models outperform human results. SRN models also show significant above chance performance, typically for 50 or more hidden units and a learning rate of 0.5 or higher. All control models performed not different from chance (all df=79, $p>0.05$) for all stimulus sets, suggesting that there was no model induced bias. In general, the SRN models tend to have a less strong effect of online-learning and often perform slightly lower than humans. However, unlike many n-gram models, SRN models exhibit around chance performance for the stimulus set by Reber & Perruchet (2003), just like in the human results. The structures by Meulemans & Van der Linden, exp. 2a were not learned by either models or humans, whereas in their exp 2b, interestingly, models and humans preferred ungrammatical structures as familiar.

Simulation 2

The purpose of simulation 2 was to investigate to which extent the window of available past familiarity judgments influences the online-learning efficiency. Therefore, one small change was introduced to the process of the grammaticality judgement decision: whereas the grammaticality response compared the current familiarity value to all previous familiarity values, now it was only compared to the last 5, 10, 20, or 30 values, using a sliding window technique. The cognitive motivation for this change was to incorporate some of the effect of human memory limitations in the modelling.

The same models and the same parameter space as in simulation 1 have been evaluated for the different memory windows above. Results revealed that performance for both model types slightly improved overall when less (window size of 10 or 20) but not too little context (window size of 5) of familiarity judgments is taken into account. The mean model performance improved for .003, .010, .013, .007 (n-gram models), and 0.012, 0.016, 0.014, 0.009 (SRN models) percent points for memory windows of 5, 10, 20, 30 respectively, compared to an unlimited memory window[1]. This small improvement may be explained through the fact that familiarity values tend to increase and to converge throughout the test. When the familiarity window excluded older values in which the models were in a prior, less stable state, the performance improves, having an even greater effect for high-performing models[1].

Why do the right structures get picked?

The behavioural and computational findings beg the main question of how it is possible that grammatical structures may potentially be learned gradually and in an unsupervised manner, within an environment that contains 50% ungrammatical structures, i.e. a fair amount of misleading and wrong information. The model simulations give rise to a potential explanation and a hypothesis for human behaviour extending Redington & Chater's (1996) argument: stimulus structures, both grammatical and ungrammatical structures, share a large set of fragments or chunks, and those are acquired with every testing of grammatical and ungrammatical stimulus. If one assumes that the learning of chunks or fragments constitutes one major part in artificial grammar learning (Servan-Schreiber & Anderson, 1990; Perruchet & Pacteau, 1990; Pothos, 2007), the chunk distribution of stimuli would supposedly play a major role in the learning. Whereas grammatical chunks appear relatively frequently, ungrammatical chunks, however, arise from violations in the structure and are thus expected to appear less frequently. Once a learner detects differences between chunk frequency in stimuli, a distinction between grammatical and ungrammatical chunks might be possible on that base. Therefore, the reason why responses converge toward grammatical structures may rely on the fact that grammatical sequences tend to have higher chunk

[1] Detailed results had to be omitted out due to space limitations.

frequencies on average than ungrammatical sequences.

Accordingly, one might hypothesise that if grammatical and ungrammatical chunks were to appear comparably frequently in the whole test set, the learner could not distinguish between them. Secondly, it would be expected that the learner picks the structures with the larger share of frequent fragments as grammatical; and hence the selection would converge toward either grammatical or ungrammatical structures depending on which one encompasses the more frequent chunks. Using the associative chunk strength (ACS) measure (Meulemans & Van der Linden, 1997), we would predict that the set of stimuli with the greater mean ACS with respect to the whole set of testing structures will be preferred and that the performance would be around chance if both mean ACS values were very similar.

Accordingly, the proportion of mean grammatical ACS to ungrammatical ACS was calculated for the different stimulus sets used above. The ACS proportion values were roughly about 1 for Meulemans & van der Linden, exp 2A, Reber & Perruchet; greater than 1 for Dulany et al., Loui et al., Rohrmeier et al., and Tunney & Shanks, and smaller than 1 for Meulemans & van der Linden, exp 2B. Mean ACS values for grammatical and ungrammatical structures were significantly different for Dulany et al, Rohrmeier et al., Tunney & Shanks Meulemans & van der Linden, exp 2B (all p<0.02). Both human performance and model performance match the pattern of the ACS proportions in terms of both direction and extent of performance: Human performance for the first (balanced) studies is at chance, and models perform not as well or at chance. Human and machine performance for the second set of studies is above chance. In the third case, human performance is below chance (Meulemans & van der Linden (1997) do not report if it is significant) and this is matched by significant below chance performance of the computational models. The correlation between ACS proportions and human as well as model performance were high: 0.71 (human performance), 0.98 (2-gram & 3-gram models), greater than 0.90 (other n-gram models), greater than 0.84 (SRN models with 80 or 120 hidden layers and learning rates greater than 0.7), and 0.89 (best SRN model). Finally, it is interesting to note that n-gram models show that some above chance online learning was possible for the structures by Reber & Perruchet, and Loui et al., even though the mean ACS values for their grammatical und ungrammatical structures were not significantly different (both p>0.4).

The learning curve

Another related question concerns the shape of the learning curve. Assuming that the performance curve of the online learning effect mainly depends on the gradual acquisition of information (about the distribution of the stimulus features or chunks) throughout the testing phase, a very simple estimate of the learning and its growth can be formulated based on common considerations. Assuming that new information gained about the sequences decreases as more sequences are known, a decreasing function of information intake may be expressed:

$$f(t) = b \cdot x^{-a} \quad for \ \ \mathrm{a, b} \in \mathrm{R}^{+} \tag{1}$$

Accordingly, the total knowledge about the structures at a certain time step is the amount of the information acquired up to that time:

$$K(t) = \int f(t)\, dt \tag{2}$$

Further, assuming that the performance in term of the likelihood of a correct response is proportional to the total knowledge about the sequences at a time, simple performance curve estimates can be derived:

$$P(t) \propto K(t) \tag{3}$$

$$P(t) = c \cdot K(t) = c\int f(t)dt = \begin{cases} C + k_1 x^m & a \neq 1 \\ C + k_2 \log(t) & a = 1 \end{cases}$$

$$with \quad k_1 = \frac{cb}{1-a} \quad k_2 = cb \quad m = 1-a \tag{4}$$

This consideration yields a logarithm or power function prediction, based on two or three free parameters, for the performance curve of the online learning effect. These curves relate to well-known power laws of human learning (Newell & Rosenbloom, 1981; Anderson, 1995) and fit the human data well, which was available for the study by Rohrmeier et al. (Fig. 1). They also match the computational learning curves (Fig. 2) well (all R^2>0.94; further details were omitted due to space limitations).

Discussion and Conclusion

The findings above suggest that there are some theoretical and empirical grounds to assume an online learning effect. The results from the first and second simulation show that the online-learning effect can be reproduced by cognitively motivated symbolic and connectionist models and that a limited memory window improves the performance.

The learning effect is possible when ungrammatical structures contain grammatical fragments to a large extent. The considerations and simulations suggest that online learning occurs because responses tend to converge towards sequences with high ACS values, independently of them being grammatical or ungrammatical. This yields a hypothesis for future experimental work: behavioural experiments may reveal whether participants indeed would tend to choose structures with high ACS independently of whether they are rule based or not in an online learning situation. Future work may further assess to what extent ACS of grammatical and ungrammatical sequences predicts the direction and extent of human performance well.

Theories of AGL (Pothos, 2007) propose that there are several theoretically plausible forms of the acquired knowledge, such as chunk knowledge, anchor positions, rule knowledge, or, microrules. This research was based on chunk knowledge and showed that it could predict an online-learning to a certain extent. It remains open which effect the other features or factors may have with regards to the online learning effect.

Although the models in this study show an effect of online learning, the results do not fully account for human results: the fragment-based n-gram models tended to learn 'too efficient' and to outperform the human results whereas the SRN models tended to perform worse than human results. From this perspective, strongly n-gram based accounts of human learning (Perruchet & Pacteau, 1992) would require to incorporate explanations of lower human performance compared to the efficiency of models based on n-gram representations, whereas connectionist accounts would need to account for the better human performance.

One remaining question concerns why this effect has not been commonly found in other studies. The reason why Reber & Perruchet (2003) have found no online learning effect of untrained controls in their experiments, appears to stem from the fact that their grammatical and ungrammatical structures are highly balanced in terms of their ACS. Other studies, in which ACS was unbalanced towards grammatical structures or ungrammatical structures found performance in favour of potential online learning. Yet more experimental evidence is needed.

Another potential explanation for the little present evidence of the effect may be that unambiguously clear control group instructions are difficult to generate and that stimulus appearance might influence learnability in the context of online learning where very quick memorisation is required. Most AGL studies use abstract letter sequences such as VNRX which have little overlap with everyday structures, language, or sounds. In this respect it is striking that two studies which used melodies of simple sequential structure (Rohrmeier et al., submitted; Loui et al., 2008) found very high performance of untrained controls about 60%. Similarly, Reber & Perruchet's (2003) study found higher performance when using consonants common in French language. Whether there is an effect of stimulus domain and appearance for online learning remains to be further explored. These findings have an impact for the AGL research paradigm in as much as some learning effect during testing has to be assumed, even though its additional impact after a learning phase might be small.

Acknowledgments

This work was supported in part by Microsoft Research through the European PhD Scholarship Programme and the Arts & Humanities Research Council.

References

Altmann, G.T.M., Dienes, Z., and Goode, A. (1995). On the modality independence of implicitly learned grammatical knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*, 899–912.

Anderson, J. R. (1995). *Learning and Memory: An Integrated Approach*. John Wiley & Sons.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Brooks, L.R., and Vokey, J.R. (1991). Abstract analogies and abstracted grammars: Comments on Reber (1989) and Matthews et al. (1989). *Journal of Experimental Psychology,* 55: 316-323.

Cleeremans, A., & Dienes, Z. (2008). Computational models of implicit learning. In R. Sun (Ed.), *Cambridge Handbook of Computational Psychology*. Cambridge University Press, 396-421.

Dulany, D. E., Carlson, R. A., and Dewey, G. I. (1984). A case of syntactical learning and judgment: How conscious and how abstract? *Journal of Experimental Psychology: General,* 113:541-555.

Dienes, Z., Broadbent, D. E., and Berry, D. C. (1991). Implicit and explicit knowledge bases in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 17, 875-882.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 16*, 41–79.

Kuhn, G., & Dienes, Z. (2008). Learning non-local dependencies. *Cognition*, 106, 184-206.

Loui, P., and Wessel, D. (2008). Learning and linking an artificial musical system: Effects of set size and repeated exposure. *Musicae Scientiae*, 12(2), 207-230.

Manning, C.D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Meulemans, T., and Van der Linden, M. (1997). Associative chunk strength in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23:1007–1028.

Mitchell, T. (1997). *Machine Learning*, McGraw Hill.

Moffat, A. (1990). Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, *38*(11), 1917–1921.

Newell, A. and Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In R, A. J., editor, *Cognitive Skills and their Acquisition*, pages 1–51. Erlbaum, Hillsdale, NJ.

Pearce, M. T., and Wiggins, G. A. (2004). Improved methods for statistical modelling of monophonic music. *Journal of New Music Research,* 33(4), 367-385.

Pearce, M. T. and Wiggins, G. A. (2006). Expectation in melody: The influence of context and learning. *Music Perception*, 23(5):377–405.

Perruchet, P. and Pacteau, C. (1990). Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *Journal of Experimental Psychology: General*, 119:264–275.

Pothos, E. M. (2007). Theories of artificial grammar learning. *Psychological Bulletin*, 133(2):227–244.

Reber, A. S. (1993). *Implicit learning and tacit knowledge: An essay on the cognitive unconscious.* New York: Oxford University Press.

Reber, R., and Perruchet, P. (2003). The use of control groups in artificial grammar learning. *The Quarterly Journal of Experimental Psychology*, 56A: 97–115.

Redington, M., and Chater, N. (1994). The guessing game: A paradigm for artificial grammar learning. In A. Ram & K. Eiselt (Eds.), *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society* (pp. 745-749). Hillsdale, NJ: Erlbaum.

Redington, M., and Chater, N. (1996). Transfer in artificial grammar learning: A reevaluation. *Journal of Experimental Psychology: General,* 125:123-138.

Rohrmeier, M., Rebuschat, P., and Cross, I. (submitted). Incidental learning of melodic structure.

Servan-Schreiber, D. and Anderson, J. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16:592–608.

Tunney, R. T. and Shanks, D. R. (2003). Subjective measures of awareness and implicit cognition. *Memory & Cognition*, 31(7):1060–1071.

Evaluation	**Parameters**		**Dulany et al, 1984**	**Reber & Perruchet, 2003**	**Loui et al, 2008**	**Rohrmeier et al, submitted**	**Meulemans & Van der Linden, 1997**		**Tunney & Shanks, 2003**
							Exp 2a	**Exp 2b**	
ACS proportion	bi- and trigrams		1.372	1.009	1.026	1.223	0.975	0.833	1.101
Human results (untrained controls)			0.560	0.445 0.513 0.490	0.60	0.616	0.490	0.450	–
		max n							
n-gram model		2	0.764*	0.538*	0.587*	0.688*	0.486	0.406*	0.579*
mw = ∞		3	0.769*	0.540*	0.592*	0.686*	0.488	0.397*	0.585*
		4	0.758*	0.566*	0.583*	0.726*	0.485	0.411*	0.569*
		5	0.757*	0.573*	0.579*	0.773*	0.502	0.421*	0.575*
		6	0.760*	0.552*	0.597*	0.798*	0.503	0.431*	0.582*
		∞	0.758*	0.574*	0.587*	0.819*	0.491	0.432*	0.576*
n-gram control			0.500	0.500	0.500	0.500	0.500	0.500	0.500
	hid	**lr**							
SRN models	10	0.1	0.491	0.503	0.505	0.506	0.498	0.491	0.503
m = {0.1,0.3,		0.3. 0.5	0.511	0.503	0.507	0.517	0.490	0.490	0.511
0.5,0.7,0.9}		0.7. 0.9	0.518	0.500	0.512	0.530*	0.493	0.476	0.515
mw = ∞	15	0.1	0.499	0.502	0.505	0.511	0.490	0.494	0.505
for all models		0.3. 0.5	0.517	0.499	0.508	0.528*	0.488	0.482	0.519
		0.7. 0.9	0.531*	0.502	0.520	0.529*	0.487	0.471	0.516
	25	0.1	0.500	0.497	0.498	0.515	0.489	0.488	0.511
		0.3. 0.5	0.520	0.500	0.516	0.531*	0.488	0.470	0.519
		0.7. 0.9	0.538*	0.498	0.519	0.536*	0.486	0.463*	0.527*
	50	0.1	0.512	0.494	0.511	0.516	0.493	0.487	0.514
		0.3. 0.5	0.533*	0.501	0.517	0.534*	0.483	0.465*	0.529*
		0.7. 0.9	0.550*	0.500	0.526*	0.542*	0.474	0.449*	0.536*
	80	0.1	0.513	0.502	0.511	0.514	0.488	0.479	0.520
		0.3. 0.5	0.536*	0.497	0.520	0.534*	0.483	0.462*	0.538*
		0.7. 0.9	0.558*	0.497	0.537*	0.544*	0.473	0.446*	0.539*
	120	0.1	0.520	0.501	0.514	0.530	0.478	0.476	0.524
		0.3. 0.5	0.542*	0.495	0.527*	0.538*	0.470	0.452*	0.538*
		0.7. 0.9	0.566*	0.498	0.543*	0.549*	0.474	0.439*	0.538*
SRN control			0.492	0.499	0.501	0.501	0.490	0.496	0.497
Best scoring SRN mw = 20. m=0.1	120	0.7	0.579*	0.493	0.552*	0.577*	0.481	0.444*	0.538*

Table 1. Associative chunk strength proportions for bi- and trigrams and mean performance (SD was omitted due to space limitations) for n-gram models and SRN models with no restrictions on the memory window. SRN results were collapsed over all momentum values. All marked (*) mean values are significantly different from chance (all df=79, p<.0001). Displayed parameters are maximal fragment length for n-gram models (max n), number of hidden layer units (hid), learning rate (lr), momentum (m) for SRN models, and memory window size (mw, in number of past stimuli).

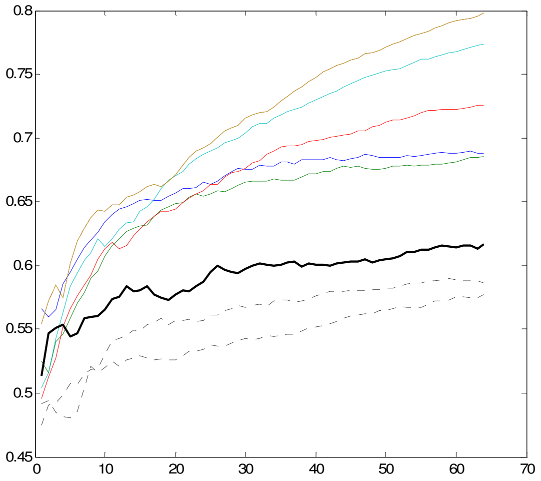


Figure 2. Comparing online learning curves for the sequences by Rohrmeier et al. (submitted) for (from top to bottom) n-gram models (coloured) for n=6,5,4,3,2, human performance (thick line) and two high scoring SRN models (dashed, hid=120/80, lr=0.7, mw=20/10, m=0.1/0.7 respectively). Power functions fit all learning curves well (all R^2>0.94), yet plots or details were omitted here due to space limitations.

A Topic Model For Movie Choices and Ratings

Timothy N. Rubin (trubin@uci.edu)
Mark Steyvers (mark.steyvers@uci.edu)
Department of Cognitive Sciences, University of California, Irvine
Irvine, CA, 92697-5100

Abstract

User input to recommendation systems such as Netflix provide an excellent opportunity to study human choice and preferences. We present a probabilistic model that captures two processes that underlie human input to recommendation systems; the process by which individuals choose items to rate, and the process by which they select a rating for those items. Using movie rating data collected by Netflix, we demonstrate that this model can generate accurate predictions about missing movie ratings. Furthermore, we show that the implicit information that users reveal through their choice processes can be used to improve prediction accuracy even in the total absence of explicit ratings.

Keywords: Choice, Decision Making, Recommendation Systems, Topic Models, LDA, Machine learning

Introduction

Recommendation systems are becoming increasingly important in industry and academia. While the field of recommender systems is heavily researched in the area of machine learning and data-mining (see Adomavicius & Tuzhilin, 2005 for an overview), it has been largely ignored by the cognitive science community. This is somewhat surprising, because an accurate model of human preferences requires understanding the basic psychological processes underlying choice and judgment. In addition, the goal of any recommendation engine is ultimately to provide a good prediction of what a particular individual will like. This requires an understanding of individual differences as they relate to preference judgments and choice behavior.

Consider the process by which you produce a movie rating. Typically, you first choose a movie to watch, then watch the movie and form an opinion of it, and finally translate this opinion into a discrete rating. This full sequence of events is important in determining what ratings are actually observed by a commercial recommendation system such as Netflix. And at each point in this process, choice plays a key role. We choose movies to watch based both on our preferences and on the situation—what mood we are in, what type of movie we feel like that night, and who we are with. And our opinion of a movie can be significantly influenced by the conditions in which we saw it (for example, you might love horror movies, but have a bad opinion of *The Shining* because it gave your child nightmares for a month). Even the process of picking a discrete rating based on an internal representation of preference involves choice.

In addition to determining which ratings are observed, choices reveal information about peoples' preferences; without knowing someone's actual movie ratings, we can get a sense of their movie tastes from which movies they see. Hofmann (2004) described the two complementary sources of information about user preference as *implicit* data (which movies users watch or otherwise show interest in), and *explicit* data (the ratings users assign to movies). The notion of implicit vs. explicit data presents an interesting question—how much, exactly, can we learn about an individual's preferences through their choices alone? Suppose that all we know about a user is that they have watched *Full Metal Jacket*, *The Godfather*, and *Goodfellas*. How accurately can we predict ratings that this user will give to other movies based solely on this information? And more to the point, how well can we make recommendations for them? Now suppose that we are told that they gave ratings of 3, 5 and 4 to these movies respectively (on a scale of 1-5). How much additional knowledge do we now have about this user? How much better can we make predictions (and recommendations) for this user?

In this paper, we attempt to answer these questions by developing a model of human ratings that describes the process by which individuals choose movies and then produce a rating for them. We develop a probabilistic framework for understanding individual differences in preference, and specify a generative model that describes how users choose movies to watch and choose ratings for these movies. After demonstrating that this model can produce interpretable dimensions of movie preferences, we compare how well this model can make predictions given different amounts of both implicit and explicit user data. We apply this model to a subset of the Netflix dataset that was released as part of a competition for researchers to develop the next generation of recommender systems (Bennett and Lanning, 2007).

The Current State of Recommendation Systems

The majority of recommendation systems currently use collaborative-filtering based techniques such as a k-Nearest-Neighbors algorithm (kNN) (Schafer et. al, 2007). Collaborative-filtering approaches typically generate recommendations for a user by finding items that have been given high ratings by similar users (where "similarity" is measured using a metric such as the Pearson correlation coefficient between the users observed ratings). While this often produces accurate predictions, the psychological underpinnings of this model are unclear; these approaches do not model latent psychological features, nor do they account for individual differences in choices. Furthermore, while collaborative filtering produces clusters which can illuminate groups of similar items, they do not produce dimensions that are readily interpretable; although knowledge that two movies have positive covariance can be useful for predictions, it does not tell us why these two movies are similar.

Another common technique for analyzing user-ratings is singular-value decomposition (SVD), in which a matrix of ratings for a set of users is decomposed into spaces where users as well as movies are modeled as points in a high-dimensional space (Sarwar et al., 2000). This technique captures the notion that individuals can be characterized by a set of latent features. However, it is difficult to extend the SVD representation to allow for variations in the ways users and items are represented; because there are no separable dimensions, users and items cannot be similar in some respects but dissimilar in others. Furthermore, this technique does not capture the processes by which items are chosen or ratings are generated.

Modeling User Choice When considering rating data that are volunteered by a user, there are two separate processes that have significant impact on which items are rated. The first process involves movie choice—why does a particular user choose to watch a particular set of movies but not others? The second process guides rating choice—given that a user has watched some movie, what determines whether they will actually provide a rating for it, and if they do provide a rating, how do they choose a rating that reflects their opinion of the movie?

Marlin et al. (2007) showed that users are more likely to rate items for which they have a strong opinion (particularly when the opinion is favorable). These authors go on to demonstrate the significance of missing-data models for producing unbiased predictions for user-ratings. This is an important result, but for the purposes of this paper we do not account for this missing-data mechanism. Rather, we focus on the largely ignored questions of how users choose movies to watch, and choose ratings to represent their opinions of the movie.

The Ratings Topic Model

This paper presents the Ratings Topic Model, a probabilistic model of movie ratings (Figure 1). The model attempts to capture two related processes: the process of choosing a movie to watch, and the process of choosing a rating for the movie. Our model combines features of Latent Dirichlet Allocation (LDA) and the ordered-logit model to explain both processes. LDA is an established probabilistic framework for extracting latent dimensions from data, particularly in the field of corpus analysis (Blei et al., 2003). The ordered-logit model is an econometric model for Likert rating scales (Train, 2003), and is related to the polytomous Rasch model studied in psychometrics (Andrich, 1978). Our model is related to a model proposed by Hofmann (2004). However, Hofmann (2004) focuses on a formulation of this model in which user choice processes are not explicitly considered and do not influence users' ratings. Furthermore, his model lacks a generative process by which users convert their preferences into discrete ratings.

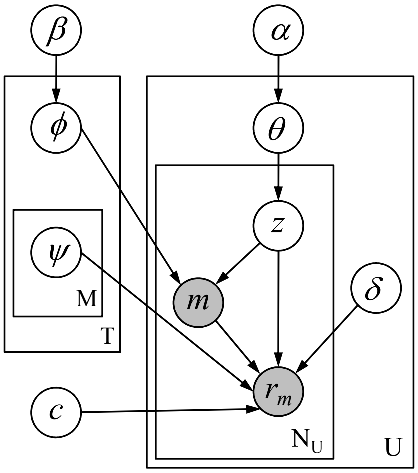


Figure 1: Graphical Model for the Ratings Topic Model

The Ratings Topic Model addresses some of the weaknesses inherent to both collaborative filtering and SVD-based approaches to modeling ratings. In addition to describing the role that choice processes play in determining what data is observed, LDA produces a set of separable latent dimensions of human preference. Without modeling separable dimensions, it is difficult to explain the underlying reasons why sets of items are rated similarly. This is particularly true with something as complex as human preferences, since items can be liked or disliked for different reasons by different users. Additionally, items or people can be highly similar with respect to one feature (e.g. a particular genre), while being dissimilar with respect to a different feature. For example, which of these would you consider more similar to the television series *The Sopranos*: *Casino*, or *Sex and The City*? It is likely that people would disagree on this answer, because although the genre of *The Sopranos* is closer to that of *Casino*, *Sex and the City* is similar to *The Sopranos* in that it they are both critically acclaimed television series produced by H.B.O.

Our probabilistic approach employs LDA to model user movie choices and preference, and an ordered-logit model to capture the process by which preferences are converted into an observed rating. We assume that users can be modeled as mixtures of topics, and that each topic represents a probability distribution over movies and preferences. In this process, once a user has selected a topic, some movies are more likely than others to be watched, and some movies are more likely than others to be enjoyed. Intuitively, we can think of a topic as any feature that might guide what people choose to watch or how they rate it (e.g. genre, release date). Once a movie has been selected, the user's rating for the movie is a function of the topic used to choose it.

The Ratings Topic Model is a generative model in that it defines a process to generate the distribution of preferences and choice probabilities for each topic, and the process by which users produce a set of ratings on the basis of these topics. For all topics $z = 1 \ldots T$, we pick a multinomial probability distribution over movies ϕ, which determines the probability $p(m_i|z_i = j)$ of choosing each movie, $m = 1 \dots M$, given a topic, z. For each topic, movies are independently assigned a preference parameter $\psi_{m,t}$ which determines how much a user will enjoy the movie given the topic used to choose it.

For each user, we first sample a multinomial mixture of topics (θ) from a Dirichlet prior α. This mixture determines the probability $p(z_i|u)$ that the user's choice and rating will come from topic z. Each time we produce a rating for a user, we first select a topic according $p(z_i|u)$, and then select a movie from that topic according to $p(m_i|z_i = j)$. The probability that the user will choose movie i is given by:

$$p(m_i) = \sum_{j=1}^{T} p(m_i|z_i = j)P(z_i = j)$$

Once a movie has been selected, a numerical rating for that movie is generated according to the probabilities specified by the ordered-logit component of the model.

The ordered-logit model treats ratings as a function of utility (U), which we define as the sum of the preference parameter and a bias parameter: $U_{u,m} = \psi_{t,m} + \delta_u + \varepsilon$. The bias parameter δ_u is specific to each user and determines the general tendency of a user to give favorable ratings. The probability of observing rating r_i is defined as probability that U falls between the rating thresholds $c_i$and c_{i+1}. Noise is modeled using a logistic function, such that:

$$P(r = r_{u,m}|\, \psi_{t,m}, \delta_u, c) = P(c_i < U_{u,m} < c_{i+1}) = \frac{1}{1 + e^{\,\psi_{t,m}+\delta_u - c_{i+1}}} - \frac{1}{1 + e^{\,\psi_{t,m}+\delta_u - c_i}}$$

The rating-thresholds c determine which values of U correspond to each of the possible observed ratings (1...5) and are set globally – all users are assumed to have to same set of rating thresholds (but different biases). Figure 2 Illustrates how relative rating probabilities change as a function of *U*.

Model parameters were learned through Markov-Chain Monte Carlo methods, using a hybrid of Gibbs sampling and Metropolis-Hastings steps. Details of inference procedure are provided in supplementary material.[1]

Dataset The Ratings Topic Model was evaluated on a subset of the Netflix dataset. This dataset is comprised of over 100 million anonymized user ratings on movies and television shows collected between 1998 and 2005. For model evaluation we selected a relatively dense subset of 500 movies and 10,000 users, containing approximately 950,000 ratings (about 20% of elements were thus filled, in contrast to 1% for the full Netflix dataset). The model was run using *T* = 1, 10, 20, 25 and 50 Topics.

Topic Examples

For every topic, a number of informative features can be visualized: (1) a ranking based on $p(m_i|z)$ that shows the movies most likely to be chosen given that a user has selected the topic, (2) a ranking based on $\psi_{m,t}$, showing the movies which have the highest and lowest expected ratings given the topic, and (3) a ranking based on $p(r, m_i|z)$ illustrating the movies with the highest joint probability of being chosen and being assigned rating of either a 1 or 5. Figure 3 illustrates these features using three topics taken taken from a single Gibbs sample using *T*=25.

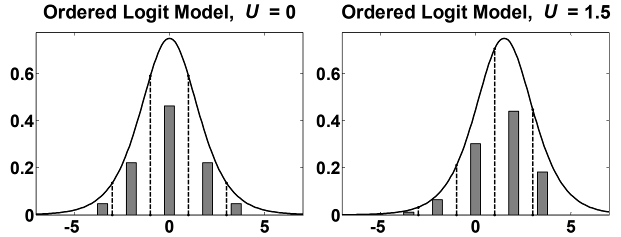


Figure 2: Left panel illustrates the logistic distribution for *U*=0, with rating thresholds depicted by dashed vertical lines. The shaded bars show probabilities of each rating. The right panel illustrates how rating probabilities change when *U* shifts from 0 to 1.5. When *U*=0 the most likely rating is a "3"; when *U*=1.5, the most likely rating is a "4".

Probability of Movies Given a Topic A Topic's probability distributions over movies models the processes guiding movie choice. Since movie choice is an overt process, it is not surprising that this feature typically discriminates topics in an intuitive manner. The movies that are most likely to be chosen given some topic usually have obvious thematic similarities. Looking the examples given in Figure 3 we can see that the movies most likely to be chosen under each topic are from similar genres. For example, the movies most likely to be chosen under Topic 4 are all horror films, with an emphasis on "classic horror" films. The movies most likely to be chosen under Topic 20 are fairly recent romantic comedies, while those in Topic 23 are mostly recent crime dramas.

Expected Movie Ratings Given a Topic While the choice dimension of a topic is highly interpretable, it does not always reflect user preferences; just because people are likely to watch a movie doesn't mean that they are likely to enjoy it. To interpret the topic along the dimenion of preference, we can look at which movies have the highest and lowest expected ratings given some topic (this is a function of parameter $\psi_{m,t}$).

For example, consider a person that often chooses movies according the distribution in Topic 20 (i.e., he is very likely to watch romantic comedies), and suppose that he is browsing for this type of movie one night. The model predicts that he is likely to enjoy movies with high values of $\psi_{m,t=20}$. Thus, even though he is more likely to choose *10 Things I Hate About You* than season 5 of *Sex and the City*, the model predicts that he will be more likely to enjoy *Sex and the City*. On the other hand, a person that is in the mood for a crime drama and therefore chooses a movie from Topic 23 is expected to strongly dislike *Sex and the City*.

[1] http://www.socsci.uci.edu/~trubin/

This does not mean that everyone who watches a lot of *The Sopranos* is expected to dislike *Sex and the City*; it just means that if they choose *Sex and the City* from this topic they they are unlikely to enjoy it. In fact, a user choosing from Topic 5 (not shown) has a high probability of giving both *The Sopranos* and *Sex and the City* high ratings.

Joint Probabilities of Ratings and Choices While the predicted rating for a movie under a topic can be highly informative, it often does not tell the whole story. Since these predictions are conditional on users actually choosing the movie under the topic, the probability of observing high or low ratings from movies that have the highest and lowest expected values may be relatively small. On the other hand, if we consider the movies that are most likely to be both chosen and given a high or low rating, we are often able to find the most liked or disliked movies that are topic-relevant. For example, *The English Patient* has a very low expected rating under Topic 4, but if also has a low probability of being chosen. However, looking now at the movies that are most likely to be given a rating of 1 under the topic, we see mostly topically relevant movies that tend to be disliked, such as *The Ring Two* and *The Grudge*. Since these movies are often chosen and subsequently disliked, we label these movies as those that are "most likely to dissapoint" users. Conversely, movies that have a probability of being chosen and then liked we label as "most likely to please".

Sometimes the fact that there is a very high preference for a movie can overcome the fact that it isn't among the most likely movies to be chosen, as with *Sex and The City* in Topic 23; although none of the seasons of this show are among the top 15 most likely to be chosen, they compose all seven of the top "most likely to please" spots, because they have such a high expected rating.

In some cases, the connection between the preference and the choice dimension is not totally intuitive. For example, both *Labyrinth* and *The Neverending Story* are expected to be well-liked by users picking from Topic 4, even though they don't have a high probability of being chosen. These sorts of "unlikely favorites" are particularly interesting when we consider the domain of recommendation systems. While it might generally be a smart approach to recommend movies with high probabilities of being chosen and also being liked, these recommendations may not always be particularly useful since the user is likely to choose them anyway. The most interesting and useful recommendations might be those movies that are unlikely choices but that nevertheless are likely to be enjoyed.

Predicting User Ratings and Choices

A standard approach for model assessment is to see how well a model can predict unobserved data. For this purpose, we removed five ratings from each user in our Netflix subset. These items were used as a test set, while all remaining ratings were used to train the model using $T = 1$, 10, 20, 25 and 50 topics. Several performance measures

	Choice Dimension $p(m \mid t)$		**Preference Dimension** $E(r \mid m, t)$	**Joint Probability** $p(r, m \mid t)$
			Topic 4	
p	***Most Likely Choices***	**E(*r*)**	***Highest Rated***	***Most Likely To Please***
.031	Poltergeist	4.4	Labyrinth	The Exorcist
.030	Carrie	4.2	The Exorcist	Poltergeist
.029	A Nightmare on Elm Street	4.2	The NeverEnding Story	Misery
.027	Halloween	4.2	Aliens	Halloween
.025	Misery	4.1	Alien	A Nightmare on Elm Street
.024	Scream	4.0	Primal Fear	Carrie
.023	Saw	4.0	Superman: The Movie	The Lost Boys
.022	The Exorcist	4.0	Misery	Scream
.022	The Grudge	4.0	Poltergeist	Saw
.021	The Lost Boys	4.0	South Park: Bigger, Long…	Alien
.021	Friday the 13th	4.0	Lean on Me	Bram Stoker's Dracula
.020	Final Destination 2	4.0	The Life of David Gale	Aliens
.020	Stir of Echoes	3.9	Bram Stoker's Dracula	Stir of Echoes
.020	Sleepy Hollow	3.9	Thelma & Louise	Frailty
.019	Frailty	3.9	Halloween	Dawn of the Dead
.017	From Hell	3.9	The Lost Boys	Labyrinth
.017	I Know What You Did La…	3.9	Sleepers	Fatal Attraction
.016	The Haunting	3.9	Hostage	The NeverEnding Story
.016	Rosemary's Baby			
.016	Hide and Seek	**E(*r*)**	***Lowest Rated***	***Most Likely To Dissapoint***
.016	Bram Stoker's Dracula	2.3	Where the Heart Is	Dreamcatcher
.016	Dreamcatcher	2.3	Dr. Dolittle 2	The Ring Two
.015	Stigmata	2.2	Sneakers	White Noise
.015	Resident Evil	2.2	Team America: World Pol…	The Haunting
.014	The Ring Two	2.1	The English Patient	Catwoman
.014	The Gift	2.1	Black Sheep	The Grudge
.014	Fatal Attraction	2.0	Catwoman	Hide and Seek
.013	Alien	1.9	8 Mile	Scary Movie 2
			Topic 20	
p	***Most Likely Choices***	**E(*r*)**	***Highest Rated***	***Most Likely To Please***
.019	Ever After: A Cinderella St…	4.9	Sex & the City: Season 6-1	Sex & the City: Season 3
.018	10 Things I Hate About You	4.9	Sex & the City: Season 4	Sex & the City: Season 2
.015	Kate & Leopold	4.9	Sex & the City: Season 3	Sex & the City: Season 6-1
.015	Save the Last Dance	4.8	Sex & the City: Season 6-2	Sex & the City: Season 1
.015	Pretty in Pink	4.8	Sex & the City: Season 1	Sex & the City: Season 4
.014	Clueless	4.8	Sex & the City: Season 2	Sex & the City: Season 5
.013	She's All That	4.8	Sex & the City: Season 5	Sex & the City: Season 6-2
.013	The Prince and Me	4.7	Friends: Season 1	Friends: Season 2
.013	Say Anything	4.7	Friends: Season 2	Friends: Season 1
.013	Practical Magic	4.4	Sleeping Beauty	Say Anything
.012	America's Sweethearts	4.4	The Parent Trap	10 Things I Hate About You
.012	Bridget Jones: The Edge…	4.4	Singin' in the Rain	Clueless
.012	Win a Date with Tad Ham…	4.2	Sense and Sensibility	Pretty in Pink
.012	Cruel Intentions	4.2	Life as a House	Ever After: A Cinderella St…
.011	What a Girl Wants	4.2	Primal Fear	Sliding Doors
.011	Chasing Amy	4.2	The Phantom of the Opera	Breakfast at Tiffany's
.011	My Girl	4.2	Beauty and the Beast	The Parent Trap
.011	Sex & the City: Season 2	4.1	Say Anything	Little Women
.011	Down With Love			
.011	40 Days and 40 Nights	**E(*r*)**	***Lowest Rated***	***Most Likely To Dissapoint***
.011	Bring It On	2.3	Waiting for Guffman	Little Black Book
.011	Sliding Doors	2.3	Saving Silverman	Kate & Leopold
.011	Return to Me	2.2	Team America: World Police	Alfie
.011	Where the Heart Is	2.2	The Naked Gun	Intolerable Cruelty
.011	Sex & the City: Season 3	2.1	Eyes Wide Shut	Eyes Wide Shut
.010	Uptown Girls	2.1	Half Baked	I Heart Huckabees
.010	Sex & the City: Season 1	2.1	The Cell	America's Sweethearts
.010	Hope Floats	1.9	Little Nicky	Win a Date with Tad Ham…
			Topic 23	
p	***Most Likely Choices***	**E(*r*)**	***Highest Rated***	***Most Likely To Please***
.032	The Sopranos: Season 1	4.8	24: Season 1	The Sopranos: Season 1
.032	The Sopranos: Season 2	4.8	Band of Brothers	The Sopranos: Season 2
.031	The Sopranos: Season 3	4.8	The Sopranos: Season 1	The Sopranos: Season 3
.030	The Sopranos: Season 4	4.7	The Sopranos: Season 2	The Sopranos: Season 4
.024	Heat	4.7	The Sopranos: Season 3	Casino
.023	Casino	4.7	The Sopranos: Season 4	Heat
.020	Donnie Brasco	4.5	Casino	Band of Brothers
.017	Rounders	4.3	Glory	24: Season 1
.014	Swingers	4.3	Swingers	Swingers
.014	The Untouchables	4.3	Hoosiers	Rounders
.014	Sleepers	4.3	The Last of the Mohicans	Donnie Brasco
.014	The Score	4.3	Friday	Glory
.013	Primal Fear	4.3	Heat	Lock, Stock and Two Smo…
.012	Lock, Stock and Two Smo…	4.2	Apocalypse Now Redux	The Untouchables
.012	The Godfather, Part III	4.2	City of God	Primal Fear
.012	True Romance	4.2	Lock, Stock and Two Smo…	Apocalypse Now Redux
.012	The Professional	4.1	The Good, the Bad and the .	True Romance
.012	The Insider	4.1	Primal Fear	Sleepers
.011	Boyz N the Hood			
.011	Glory	**E(*r*)**	***Lowest Rated***	***Most Likely To Dissapoint***
.011	The Game	2.0	I Heart Huckabees	White Chicks
.011	Spy Game	2.0	The Transporter	The Transporter
.011	Apocalypse Now Redux	2.0	Beauty and the Beast	Sex & the City: Season 3
.010	Band of Brothers	2.0	Sex & the City: Season 4	Sex & the City: Season 6-1
.010	25th Hour	1.9	Sex & the City: Season 6-2	Alexander: Director's Cut
.010	The Hurricane	1.9	Sex & the City: Season 6-1	The Cell
.010	24: Season 1	1.9	Sex & the City: Season 3	Eyes Wide Shut
.010	Raging Bull	1.4	White Chicks	Sex & the City: Season 4

Figure 3: Topic features from a single Gibbs Sample, $T = 25$

were computed to evaluate how accurately the model could predict test data using different numbers of topics. Performance was compared across different values for *T*, and against several baseline predictors.

To evaluate the accuracy of rating predictions we computed both the percent of correct predictions (using a single *maximum a posteriori* prediction for each rating), and the perplexity of the posterior predictive distribution. Perplexity is a standard measure of performance in the field of information-retrieval, and is computed as $e^{\left[-\log\frac{1}{n}\sum p(r_i)\right]}$. Perfect performance (i.e. assigning all probability to the true rating) yields a perplexity of one, while a completely uninformative prediction (assigning uniform probability to all ratings) yields a maximum perplexity of 5. The three baseline predictions for ratings we used were (1) the full marginal distribution of ratings across all users and movies in the training set, (2) the marginal distribution of ratings for the movie being rated, and (3) the optimal blend of the movie's marginal distribution with the user's marginal distribution of ratings. As shown in Figure 4, the Ratings Topic Model outperformed the baseline predictions when the number of topics was greater than one. The model made the most accurate predictions when 25 topics were used.

In addition to making predictions about ratings, the Ratings Topic Model makes predictions about user choices; for each user and movie, it assigns a $p(m_i|u_j)$, where $\sum_i p(m_i|u_j) = 1$. Predictions are made after training items have been removed, such that the prediction goal is to assign as much probability to the five test items as possible. The accuracy of these predictions was measured using perplexity. In this case, an uninformative prediction (which assigns uniform probability to all movies) yields a perplexity equal to the number of movies remaining after training items are removed. For the purposes of comparison, perplexity was also computed for the following two baseline predictions: (1) assigning uniform probability to all movies being chosen, and (2) assigning each movie its marginal probability of being chosen across all training data. Results are shown in Figure 4. The Ratings Topic Model outperformed the baseline predictions when $T > 1$, and achieved best performance with *T*=25.

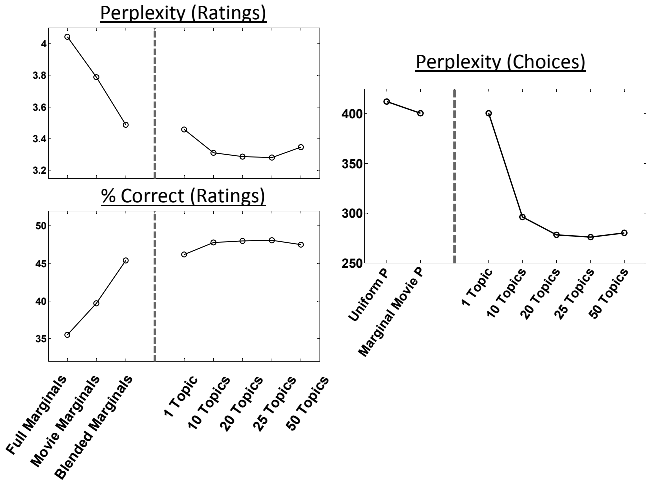


Figure 4: Accuracy of model and baseline measures for rating predictions (left) and choice predictions (right)

Implicit vs. Explicit Data

The results described in the previous section demonstrate that the Ratings Topic Model makes reasonably accurate predictions about both user choices and user ratings. For these purposes, the model uses both implicit and explicit preference data (user choices and ratings, respectively). However, it is still unclear whether the choice data itself can be used to improve rating predictions (and accordingly, whether it can improve user recommendations). In other words, is implicit data useful only for the purpose of understanding user choices, or does it capture information about user preferences, which are only explicitly observed through the ratings themselves?

To address this question, we systematically varied the amount of explicit information (i.e., the number of movie ratings) and implicit information (i.e., the number of movie choices) that was observed for each user and measured how this affects prediction accuracy for missing ratings. For this simulation, we removed a subset of 1,000 test-users from our 10,000 user subset. Complete data for the 9,000 remaining users was used to train the model on 25 topics. Topic parameters $\psi_{t,m}$ and ϕ were then fixed, so that it was only necessary to fit parameters θ and δ for each test-user.

For model evaluation, all but 50 ratings for each test-user were removed, such that we had a 1,000 user x 500 movie matrix, with 50 ratings observed in each row. This matrix was then randomly split into a training set and validation set containing 40 and 10 ratings per user respectively. The model was trained under 45 different conditions in which the number of observed ratings and choices was manipulated. (Note that since it is impossible to observe a rating without a choice, the number of choices observed here refers to the number of choices that were observed *in addition* to the observed ratings). For each condition, posterior estimates of parameters were averaged over *N* chains to generate predictions for validation data. Measures of performance under each condition were obtained using five-fold cross validation, such that all ratings in the test-set were used once in the validation set.

Measuring Performance User bias accounts for a large amount of variance in Netflix user ratings. Since bias can only be observed from users' explicit ratings, prediction accuracy does not provide a good measure to determine how much we can learn about preferences from implicit vs. explicit data. Furthermore, while it is important to account for bias when trying to accurately predict missing ratings, it is unimportant when we are interested in understanding user preferences or when making recommendations. More relevant for these purposes is the ability to predict the *relative* enjoyment of different movies. Therefore, we evaluated model performance by measuring how well it could predict which movies were rated higher than others.

For each user, all pairs of unequal ratings in the validation set provide a single comparison about relative movie preference; for each of these comparisons, we computed the posterior predicted probability that user *u* will give movie *j* a higher rating than movie *k*:

$$p(r_u > r_k) = \sum_{v=1}^{5} p(r_j > r_k \,|\, r_j = v)\, p(r_j = v)$$

We computed two measures of the accuracy of this prediction across all paired-comparisons for all users. First, we computed the perplexity of the estimate (where the baseline value of perplexity for this prediction is 2, which is obtained by assigning a .5 probability that movie *j* will be rated higher than movie *k*). In addition, we generated a binary prediction using the *maximum a posteriori* estimate of which rating would be higher, and computed the percent of these predictions that were correct. Baseline for this binary measure is 50%, since it is the expected result if we were to make random guesses. The condition with zero ratings and choices presented in the table below provides a second baseline for these measures; without any ratings or choices, predictions for all users are generated using the prior values for parameters ϕ and δ.

Figure 5 shows the perplexity and percent correct for all paired-comparisons, averaged across the five validation sets using five-fold cross validation. Looking within each row from right to left, we can see that given a fixed number of training ratings, the model is able to improve its predictions using additional knowledge about user choices. For example, for a user with 5 ratings, knowledge about 20 additional choices improves performance about as much as 10 additional ratings. Even without any ratings, knowledge about choice can significantly improve performance; the model achieves similar performance when trained with 40 choices as it does when trained with 15 ratings.

Perplexity

	Choices								
Ratings	**0**	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
0	1.823	1.806	1.793	1.787	1.782	1.777	1.775	1.774	1.771
5	1.799	1.786	1.780	1.775	1.771	1.770	1.769	1.768	
10	1.781	1.777	1.772	1.771	1.767	1.765	1.764		
15	1.772	1.766	1.768	1.762	1.762	1.760			
20	1.764	1.764	1.762	1.761	1.758				
25	1.759	1.759	1.756	1.756					
30	1.754	1.754	1.755						
35	1.750	1.753							
40	1.749								

Pct. Correct

	Choices								
Ratings	**0**	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
0	68.0	68.7	69.2	69.5	69.7	70.0	70.0	70.0	70.2
5	68.8	69.5	69.8	69.8	70.3	70.2	70.3	70.3	
10	69.7	69.8	70.1	70.1	70.4	70.4	70.4		
15	70.1	70.4	70.2	70.5	70.4	70.5			
20	70.5	70.3	70.5	70.6	70.7				
25	70.6	70.5	70.7	70.7					
30	70.9	70.9	70.6						
35	71.0	70.8							
40	71.0								

Figure 5: Prediction perplexity and percent correct for paired-comparisons, when model is trained with different amounts of choice and ratings data

Conclusion

The Ratings Topic Model provides a general framework for understanding the processes that underlie individual's rating behaviors in recommendation systems. The model can make accurate predictions about both unobserved ratings and choices, while generating interpretable dimensions that guide these processes. Furthermore, we have shown that the model can use implicit choice data in to improve predictions about a user's explicit ratings, even in the complete absence of ratings data. In addition to this being of psychological interest, it is a useful feature for real-world recommendation systems since such systems have access to a large amount of implicit preference data.

References

Adomavicius, G., Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734-749.

Andrich, D. (1978). A rating formulation for ordered response categories. *Psychometrika*, *43*, 561-73.

Bennet, J., Lanning, S., *The Netflix Prize*. *KDD Cup*, 2007.

Blei, D. M., Ng, A.Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research,* 3, 993–1022.

Griffiths, T. L., Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences, 101,* 5228-5235

Hofmann, T. (2004). Latent Semantic Models for Collaborative Filtering. *ACM Transactions on Information Systems*. 22(1): 89-115.

Marlin, B.M., Zemel, R. S., Roweis, S., Slaney, M. (2007). Collaborative filtering and the missing at random assumption, *UAI 2007: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*.

Sarwar, B. M., Karypis, G., Konstan, J. A., Reidl, J. (2000). Application of dimensionality reduction in recommender system–A case study. *Proceedings of the ACM WebKDD 2000 Web Mining for E-Commerce Workshop.*

Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. (2007) Collaborative Filtering Recommender Systems. *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer.

Train, K. (2003). *Discrete Choice Methods with Simulation*. Cambridge University Press.

Computational and Explanatory Power of Cognitive Architectures: The Case of ACT-R

Holger Schultheis (schulth@sfbtr8.uni-bremen.de)
SFB/TR 8 Spatial Cognition, Universität Bremen, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany

Abstract

Cognitive architectures constitute a generally preferable approach to create computational accounts of human cognition. Yet, cognitive architectures are also hard to assess. Following up on and extending the work of Cooper (2007), we further assess the popular cognitive architecture ACT-R in this paper. It turns out that ACT-R fares worse than one may expect both regarding the scope of empirical effects it has been shown to account for and regarding its explanatory power.

Keywords: cognitive architectures; Lakatos; Turing machines; counter machines; ACT-R.

Cognitive Architectures and Their Assessment

One approach to building computational models is to develop the model as part of a *cognitive architecture*. Cognitive architectures can be characterized as implemented theories of the fixed mechanisms and structures that underlie human cognition. As such cognitive architectures strive to offer a framework in which all of human cognition can be modeled. Building on the common mechanisms provided by the architecture, computational models for particular domains or tasks can be created by adding task and domain specific information to the architecture (cf. Lehman, Laird, & Rosenbloom, 1998). Thus, a cognitive model developed in the scope of a cognitive architecture can be viewed as consisting both of the architectural mechanisms and the task / domain specific information (i.e., content) added to the architecture.

Employing cognitive architectures for modeling human cognition has the advantage that otherwise isolated and fragmentary accounts of human cognition can be integrated to ultimately (hopefully) yield an account of human cognition as a whole (Newell, 1990). In this sense, building models in cognitive architectures is preferable to building isolated models. Obviously, this advantage of cognitive architecture will only hold, if the employed architecture is a good approximation of the general mechanisms and structures that underlie human cognition. To not jeopardize the aim of arriving at a veridical account of all human cognition, the quality of the cognitive architecture needs to be assessed and possibly improved by changing the architecture.

As Cooper (2007) points out, assessing cognitive architectures is less straightforward than assessing isolated models. Whereas isolated models lend themselves naturally to Popperian falsification, cognitive architectures do not. Against this background, already Newell (1990) argued that the development of cognitive architectures should be guided not by Popperian falsification but by criteria as arising from the theory put forth by Lakatos (1970). Following this suggestion and further supporting it, Cooper (2007) employs Lakatosian criteria to assess the two architectures Soar (Newell, 1990) and ACT-R (Anderson, 2007).

In this paper we bring to the foreground further criteria for assessing the merit of cognitive architectures. Moreover, we combine these additional criteria with the Lakatosian criteria described in Cooper (2007) to continue the assessment of the cognitive architecture ACT-R. To do this, we first briefly describe the notions and criteria relevant for the assessment. Subsequently, we assess ACT-R regarding these criteria. This comprises (a) describing those aspects of ACT-R which are most relevant for the presented assessment and (b) conducting formal and literature analyses to assess ACT-R's standing with respect to the considered criteria. Finally, we close with some implications the assessment's results have for (the future development of) ACT-R.

Assessment Criteria

Lakatosian Criteria

According to Lakatos (1970) scientific development occurs in the scope of so called research programs. Roughly speaking, each such research program comprises both a *hard core* and a *protective belt*. The hard core consists of all those assumptions which are central to the program, that is, giving them up would mean to give up the research program. In contrast, the protective belt is made up of assumptions and hypotheses of a more peripheral nature, that is, assumptions which may help to further specify aspects of the research program, but to which the research program is not irrevocably committed.

Research programs generally develop by (empirically) testing predictions derived from the hard core and the protective belt. If the predictions are confirmed, this supports the research program. If the predictions are refuted, this may lead to a change of the protective belt (i.e., some peripheral assumptions) of the research program. Depending on the consequences of the change of the protective belt, Lakatos (1970) calls a research program *theoretically progressive* or not. A research program is theoretically progressive if and only if the change of the assumptions increases the empirical content of the research program, that is, allows the research program to account for more empirical phenomena than before the change. Importantly, research programs which are not theoretically progressive are not scientific but only pseudoscientific. Lakatos (1970) further categorizes research programs as to whether they are *empirically progressive* or not. If and only if a research program's predictions are empirically confirmed, it is empirically progressive.

Sticking to Lakatos' terminology, cognitive architectures are research programs. Accordingly, one can use the notions of theoretical and empirical progressiveness to assess cognitive architectures. Consequently, following up on and adding to the work of Cooper (2007), we more closely consider ACT-R's theoretical progressiveness in this contribution.

To Can and Cannot

The above outlined Lakatosian criteria stress the ability of a cognitive architecture to account for empirical findings[1]. What a cognitive architecture can account for is, however, only one part of an architecture's quality. As Roberts and Pashler (2000) remark, it is equally important to decide on the quality of a given architecture to know what the architecture cannot account for. Neglecting the "cannot" aspect is a serious problem, because a cognitive architecture is intended to provide the basis to explain human behavior and not arbitrary behavior.

To illustrate the problem, consider a certain architecture S which allows to model a certain empirically found effect f. Let us assume that f is a reaction time difference between two experimental conditions A and B such that reaction times are longer in A. Assume further that $\bar{f}$ is the hypothetical (i.e., not observed) effect that reaction times are longer in condition B. An interesting question now is whether S also allows to model $\bar{f}$. If S allows modeling $\bar{f}$, S accounts for both the empirically found effect and its opposite.

Given such a situation, the explanatory value of S is called into question. S is a cognitive architecture and should, thus, realize the mechanisms and structure underlying human cognition. S's ability to account for both f and $\bar{f}$ undermines its assumed cognitive plausibility, because humans do only behave according to f but not according to $\bar{f}$. If the structure and mechanisms of the human mind constrains human cognition and behavior to f, a cognitive architecture which allows modeling $\bar{f}$ is erring with respect to at least some part of the structure and mechanisms underlying human cognition.

Thus, to fully judge the quality of a cognitive architecture, it is equally important to know what the architecture cannot account for as it is to know what the architecture can account for. Ideally, the architectural mechanisms and structure constitute a framework which constrains the content that can be added to it such that the set of all models possible in the architecture accounts for and only for all phenomena empirically observable in human cognition and behavior (cf. also Taatgen, 2003).

In line with its importance and in addition to theoretical progressiveness, the question what can and cannot be modeled in ACT-R is one major point of inquiry in the subsequent assessment of ACT-R.

ACT-R

ACT-R (see Anderson, 2007; Anderson et al., 2004) consists of several components which are called modules. One of these modules, called the production module, stores and executes a set of productions. Each production specifies under which conditions it is applicable. If the current state of the ACT-R system satisfies a production's conditions, the production can be executed which will lead to a change of the state of the system. Additional modules of ACT-R include the declarative module (storing declarative knowledge in the form of proposition-like pieces of knowledge called *chunks*), the goal module (managing the current goal), and several perceptual motor modules (realizing ACT-R's interaction with the environment). Each of these modules is interfaced to the overall system by a buffer. The working of the procedural module draws heavily on these buffers. Production conditions and effects are specified nearly exclusively in terms of buffer content. The productions conditions are checked against the buffers' content and production application will normally change the content of one or more buffers.

Regarding the Lakatosian criteria of architecture assessment mentioned above it is interesting to what extent one can distinguish the hard core and the protective belt realized by ACT-R. As Cooper (2007) remarks, although the developers of ACT-R have never explicitly used Lakatosian terminology to draw such a distinction, such a distinction suggests itself from the descriptions of the notions underlying ACT-R's development. For example, Anderson (1976, pp. 114) proposes several "preconceived notions" which constitute the skeleton of ACT-R. These preconceived notions, such as to distinguish between and to employ both procedural and declarative knowledge, constitute the hard core of ACT-R and have remained unchanged since their proposal in 1976. All aspects of ACT-R other than the preconceived notions can be viewed as constituting the protective belt. For instance, the formulas and mechanisms used to select one of several productions or one of several pieces of declarative knowledge are part of the peripheral assumptions.

This protective belt of ACT-R is largely parametrizable. Using the parameters the architecture provides one can determine both which of the peripheral assumptions to employ (e.g., whether to use certain formulas to determine which production to select) and how the selected peripheral assumptions behave. Since ACT-R has been first proposed by Anderson (1976), its protective belt has changed considerably. In its current version (6.0 [r723], see Bothell, 2009) which we consider here, ACT-R has about 50 parameters. Only for few of them general recommendations of how to set them exist (Anderson et al., 2004).

The Cannot in ACT-R

As a cognitive architecture, ACT-R constitutes a computational framework for building cognitive models. Due to this computational nature one manifest starting point to investigate what ACT-R cannot do is to ask which subset of the set

[1] Strictly speaking, cognitive architectures per se do not account directly for any empirical findings. Only the models which can be build in an architecture can account for empirical phenomena. However, to ease the subsequent exposition we will talk of architectures that account for findings instead of using the more cumbersome wording of architectures that allow building models which account for empirical findings.

of all computable functions cannot be realized in ACT-R. As it turns out, there are no functions which can be computed, in principle, but not in ACT-R. In the following, we prove this constructively by presenting two particular ACT-R models[2].

Universal Turing Machine A Turing machine is a computing machine which was introduced by Turing (1936). A Turing machine consists of an infinite tape partitioned into cells and a control unit moving over the tape. Each cell contains a single symbol and each machine can deal only with a finite, predefined set of symbols. The control unit can read and write on the tape one cell at a time and can move from one cell to one of the two neighboring cell. At every point in time the Turing machine is in one of a finite number of states. Depending on the machine's current state and what is read from the cell currently in focus, the machine will write to the cell in focus and / or move to an adjacent cell.

Although quite simple in their setup, Turing machines have been found to be able to compute a wide range of functions (see Minsky, 1967, for an in-depth treatment of Turing machines and several example machines). More precisely, it is generally assumed—though unproven—that the set of functions which can be computed by Turing machines is identical to the set of all computable functions. What is more, certain Turing machines, called universal Turing machines, are able to emulate the working of any other Turing machine, that is, universal Turing machines can compute anything that Turing machines in general can compute. Put differently, universal Turing machines are computing machines which can compute all computable functions. In the remainder of this section we describe an ACT-R model which emulates a universal Turing machine. The chosen machine is a machine with 4 states and 6 symbols which has been proposed by Rogozhin (1996).

To emulate the chosen machine, the tape of the machine is realized as the content of the declarative module. Each cell on the tape is represented by a chunk in declarative memory. Such a chunk *c* essentially stores (a) the symbol contained in the cell *c* represents, (b) the chunk which represents the cell which would be to the right of the cell represented by *c* on the tape, and (c) the chunk which represents the cell which would be to the left of the cell represented by *c* on the tape.

The goal buffer contains the chunk representing the cell that is currently in focus. In addition to the cell information the goal buffer also stores the current state of the machine.

The reading and writing of information onto the tape as well as the movement of the control unit is realized by productions. Basically, four types of productions are employed to realize the operations of the control unit:

- *update*: Depending on the current state of the machine and the symbol in the current cell (i.e., the corresponding symbol stored in the goal buffer), this type of production writes a symbol into the current cell (i.e., updates the corresponding slot in the goal buffer).
- *prepare transition*: As described above, the combination of a state and symbol also affords a move of the control unit. This type of production prepares such a move. By drawing on the information about the neighboring cells given in the currently focused-on cell, the production requests the retrieval of the appropriate chunk (i.e., the chunk representing the cell to move to).
- *get next*: The "get next" type of production is applicable whenever a chunk representing a cell on the tape is available in the retrieval buffer. The main purpose of this production type is to modify the cell representation in the retrieval buffer such that it can serve as the representation of the current cell in the goal buffer. This preparation comprises basically two things. First, the machine's state as resulting from the previously encountered state-symbol combination is stored in the appropriate slot of the chunk in the retrieval buffer. Second, the chunk currently in the goal buffer is stored as either the right or left neighbor of the cell represented by the chunk in the retrieval buffer. If the control unit has "moved" to the left, the chunk in the goal buffer is stored as the right neighbor and vice versa.
- *do transition*: This type of production replaces the chunk currently in the goal buffer with the chunk currently in the retrieval buffer.

These four types of productions when being executed in the sequence in which they were described constitute one elementary operation of a Turing machine: Read a symbol and then, based on the combination of current state and the read symbol, write a symbol, update the state and move to the next cell. Since the movement direction, the state to change to, and the symbol to be written depend on the previous state and the read symbol, for each possible state-symbol combination these four productions have to be slightly different. Consequently, to emulate the universal Turing machine in question, our model employs a variation of this 4-tupel of productions for each of the 24 possible state-symbol combinations.

Representing the tape by declarative memory and the working of the control unit by productions as described, allows to emulate the universal Turing machine by running the model in ACT-R. The only thing one has to do to emulate the machine computing a certain function is to provide the initial tape configuration as chunks in declarative memory and to set the initial focus to the appropriate cell of the initial tape configuration. We have successfully emulated several Turing machines using this approach. For these model runs we enabled sub-symbolic processing in ACT-R and set the latency factor parameter to 0.1. All other parameters of ACT-R were left at their default values as described in Bothell (2009).

Consequently, as the presented model runs completely in ACT-R, ACT-R allows to emulate a universal Turing machine. This shows that there is no computable function which cannot be computed in ACT-R. Moreover, the model we describe next demonstrates that this is not the only way to realize universal computation in ACT-R.

[2] Model code is available from http://www.cosy.informatik.uni-bremen.de/staff/schultheis/ICCM09-models/

Universal Counter Machine A second class of computing machines is called counter machines. A counter machine comprises a finite number of registers and can interpret a finite set of instructions. The registers store integer values and can be tested and manipulated by the instructions which are part of the instruction set of the machine. To compute some function f, a counter machine has to be equipped with an initial set of values in its registers and a program, that is, a sequence of instructions from the machine's instruction set. The machine will execute the program and once the end of the program is reached, the result of the computation will be available in one (or more) of the registers.

Minsky (1967, pp. 255) has proven that a counter machine employing only three instructions and two registers can compute any computable function. The required instructions are INC(r_i) (add 1 to register r_i and go to the next instruction), JZDEC(r_i, n) (if $r_i = 0$ go to instruction n, otherwise subtract 1 from r_i and go to the next instruction), and GO(n) (go to instruction n). Since this counter machine is universal, for any computable function f there exists a program (i.e., a sequence of instructions) and an initial value for both registers such that the counter machine computes f.

As in the case of Turing machines, it is possible to emulate computation using counter machines by devising appropriate ACT-R models. To show this, it suffices to explain how an ACT-R model can realize (a) the two registers, (b) the three instructions, and (c) the sequence of instructions. In our model, the two registers are realized as slots in a chunk, where this chunk remains in the goal buffer for the complete model run. The instructions are realized as productions. To control the sequence of instructions a third slot in the chunk in the goal buffer stores a label. This label is tested in the condition of the productions such that only the production corresponding to the current label is applicable. Against this background the three types of instructions outlined above can then be transcribed by productions as follows:

- INC(r_i): This instruction is realized by reading the current value of r_i from the goal chunk, adding 1 to that value by using the `!bind!` statement of ACT-R, and storing the resulting value again in the goal chunk.

- GO(n): To effect such a GO statement, a production needs only to change the label in the goal chunk such that it corresponds to instruction n.

- JZDEC(r_i, n): Two productions are necessary to transcribe this instruction. Both productions test the content of r_i using the `!eval!` statement of ACT-R. The first production is only applicable if $r_i = 0$ holds and essentially works as the production mimicking the GO instruction. The second production is only applicable if $r_i > 0$ holds and subtracts 1 from r_i analogous to the workings of the INC instruction.

Importantly, these methods for transcribing a program of the universal counter machine as an ACT-R model, are not program specific. Put differently, any program formulated for the universal counter machine can be transcribed as an ACT-R model. Consequently, the universal counter machine can be completely emulated in ACT-R. To illustrate the emulation of the counter machine, we implemented a model which computes the sum of two numbers. The parameter settings for this model are identical to those used in the Turing machine model. By appropriately initializing the first register, running the model computes the sum of the two numbers and encodes the result as a number in the first register.

The possibility to emulate a universal counter machine in ACT-R provides additional evidence that there is no computable function that cannot be realized in ACT-R. Although this second evidence may seem unessential, as explained in the next section, the fact that universal computation can be realized in ACT-R in more than one way is of relevance for assessing the architecture.

Summary and Discussion Both models presented above paint a clear picture of which functions cannot be realized in ACT-R: There is simply no computable function that cannot be computed using ACT-R. That is, ACT-R does not seem to fulfill the requirement to constrain the models that can be built in it too well. Consequently, at least regarding the "cannot" criterion ACT-R fares poorly.

One may be inclined to object to this conclusion or the way it was brought about. Therefore, we list and discuss several possible objections in the remainder of this section.

First, one may argue that the fact that ACT-R is Turing-complete is neither new nor problematic. Regarding originality, Anderson (1976, pp. 140) already presented the sketch of a proof of ACT-R's Turing completeness. However, the proof presented in Anderson (1976) refers to the initial version of the cognitive architecture. Over the past 30 years the overall setup of the architecture has changed considerably. In particular, certain changes (see e.g., Anderson & Lebiere, 1998, p. 440) were explicitly implemented to reduce the computational power of ACT-R. Thus, the Turing completeness of ACT-R in its current version could not be derived from the 1976 proof, but had to be newly established.

Yet, Turing completeness of ACT-R (or any cognitive architecture) may not be considered a problem. In proposing the physical symbol system hypothesis Newell (1980) argued that any system able to realize human-level intelligence necessarily needs to be Turing-complete. Against this background, it may not be immediately obvious why the above described models constitute problems for ACT-R. The problem is that it is unclear and dubitable that the presented models realize Turing completeness appropriately. As the two models indicate, Turing completeness can be realized in several ways. When using universal computing machines to achieve results in computation theory it may not be crucial which of all possible realizations of universal computation one employs. For a cognitive architecture such as ACT-R, however, the way universal computation is achieved is essential. Striving to constitute a theory of human cognition as a whole, ACT-R must realize Turing completeness in the same way as Turing completeness is achieved in the human cognitive system. Among other things, this requires that the timing behavior of the architecture and of human cognition match closely. Thus, computing any function f in ACT-R should take about the same time as human cognition requires to compute f. Put differently, to live up to its aim of being a satisfactory cognitive architecture, ACT-R should not allow to compute f considerably faster or slower than the human cognitive system computes f. This is not the case, since, as the two models show, ACT-R can be made to compute any f in a wide range of times. Different universal machines diverge considerably with respect to the time they need to compute any f (e.g., Woods & Neary, 2009). The two models prove that there is a wide range of universal machines which can be realized in ACT-R. Not only does ACT-R allow implementing different types of universal computing devices (i.e., Turing machines and counter machines), but also for each of these types numerous instances can be realized. For example, one could implement a universal Turing machine with different states, symbols, and transition rules (Rogozhin, 1996). Likewise, universal counter machines with different instruction sets and / or more registers (Minsky, 1967) can be built in ACT-R analogously to the second model described above. Thus, ACT-R allows to realize any function with a wide range of times. In the worst case, it may even be possible that any function can be realized in arbitrary time in ACT-R. Regardless whether this is the case, it seems clear that their is too few which ACT-R cannot do, to consider ACT-R as satisfactorily constraining what can be implemented in it.

A second objection that may be raised concerns the compliancy of the models, that is, the extent to which the models are formulated in keeping with the spirit of the architecture (Young, 2003). Perhaps one may want to argue that some of the model's components are violating one or more theoretical stances implicitly being part of the architecture. One difficulty with such an argument is that there is no clear and explicit definition of what type of model components do and which type of model components do not keep with the spirit of ACT-R. In addition, for constituting a satisfactory account of the fixed mechanisms and structures underlying human cognition, it should be the architecture itself and not some code of how to use the architecture that constrains what models one can build in the architecture. Thus, an objection in terms of compliancy fails to address the core issue brought up by the above presented models and considerations.

In summary, the presented analyses indicates that what cannot be done in ACT-R is considerably less than desirable. Multiple realizability of universal computation on several different time scales leaves too much room for implementing behavior which ACT-R should not allow to be implemented. As a result, ACT-R's ability to meet the "cannot" criterion is, to say the least, debatable.

The Can in ACT-R

After having considered what ACT-R cannot do, we now turn to the question what ACT-R can do. A first answer to this question directly derives from the models presented above: ACT-R allows to compute every computable function. However, this is, as also mentioned above, only a partially satisfactory answer. To fully judge ACT-R's "can" ability, it is important to more closely consider whether ACT-R allows to compute these functions as the human cognitive system computes them (e.g., regarding timing). Essentially this amounts to examine for which tasks and domains of cognition ACT-R models can be built that closely mimic human behavior and cognition. In Lakatosian terms, it is necessary to examine the empirical content of ACT-R.

Judging from the plethora of publications on ACT-R (models) listed on the ACT-R web site one would expect that ACT-R does well with respect to this empirical content criterion. To verify this impression of ACT-R's empirical content, we reviewed all papers presenting ACT-R models which were listed on the ACT-R site as being published either 2007 or 2008. These two years were chosen because they presumably represent the current state of the art in ACT-R modeling.

Overall 35 papers presenting models accounting for various aspects of human cognition are available. This is an impressive number which seems to indicate the large empirical content encompassed by ACT-R. On closer inspection, however, it turns out that the empirical content of the current ACT-R version is (a) unclear and (b) probably less than suggested by the number of presented models. The reason for this is the way the ACT-R community proceeds with the change of parts of the protective belt of ACT-R—both across and within different ACT-R versions.

Each change in version is accompanied by a change of at least some of the peripheral assumptions in ACT-R. For example, from ACT-R 4 to ACT-R 5 the goal stack was replaced by the goal buffer and from ACT-R 5 to ACT-R 6 the formula for computing production utility was considerably modified. Although there is, of course, nothing wrong with such changes per se, for each of these changes it is mostly unclear whether they are theoretically progressive, that is, whether they increase the empirical content of ACT-R. To show that the current ACT-R version's content is increased compared to its predecessors would require to prove (by reimplementation in the current version) that empirical phenomena accounted for by older ACT-R versions can still be accounted for by the current version. Such reimplementation is rarely done. On the contrary, even 2008 published modeling work is partly conducted in ACT-R 4 (e.g., Altmann & Gray, 2008) and ACT-R 5 (e.g., Gunzelmann & Gluck, 2008).

Furthermore, even for single ACT-R versions, the empirical content is unclear. There are mainly two reasons for that: First, by appropriately setting particular parameters several peripheral assumptions can be and are switched on or off at will. For instance, some models employ base-level learning or production learning while others do employ neither. Second, it is not unusual for ACT-R modeling work to modify or extend the protective belt. Of the 35 modeling paper reviewed, 17 considerably changed the protective belt, for example, by changing existing modules (e.g., Maanen & Rijn, 2007) or adding new modules (e.g., Juvina & Taatgen, 2007).

This frequent change of the protective belt across and within ACT-R versions renders it difficult to judge the empirical content of the current version of ACT-R. Whether all the modeling work employing differing protective belts is reconcilable is an open question. Due to this problem, it is not clear whether ACT-R is theoretically progressive. But even if it is, ACT-R's empirical content remains to be determined.

Conclusion

In this paper we picked up on and extended the methodology proposed by Cooper (2007) to assess the merit of cognitive architectures. We applied the methodology to one of the most commonly employed cognitive architectures, ACT-R. In this assessment, ACT-R fares worse than one may have expected. For one, ACT-R's ability to account for human cognition is less evident than suggested by the available host of modeling papers employing ACT-R. It remains to be investigated to what extent different modeling work can be integrated into ACT-R without varying its peripheral assumptions. Even if ACT-R's ability to account for empirical data turns out to be substantial, the explanatory value of this is called into question by ACT-R's computational power. Since ACT-R in its current version must be assumed to allow computing functions in a lot of ways different from human cognition, it is unclear to what extent ACT-R mirrors and, thus, explains the mechanisms and structure underlying human cognition.

Overall, ACT-R has served and still serves an important function in providing a platform for modeling human cognition. Interesting accounts of various aspects of human cognition have been formalized in ACT-R. Yet, to substantiate ACT-R's status as a cognitive architecture constituting a unified theory of cognition, it is necessary (a) to more closely determine its actual empirical content and (b) to more strongly constrain what ACT-R allows to be implemented.

Acknowledgments

In this paper work done in the project R1–[Image-Space] of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition is presented. Funding by the German Research Foundation (DFG) is gratefully acknowledged. We also thank Astrid Kurbjuweit for fruitful discussions and two anonymous reviewers for their constructive comments.

References

Altmann, E. M., & Gray, W. D. (2008). An integrated model of cognitive control in task switching. *Psychological Review*, *115*, 602-639.

Anderson, J. R. (1976). *Language, memory, and thought.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036 - 1060.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum.

Bothell, D. (2009). ACT-R 6.0 reference manual. Retrieved February 5, 2009, from http://act-r.psy.cmu.edu/actr6/reference-manual.pdf. [Computer software manual].

Cooper, R. P. (2007). The role of falsification in the development of cognitive architectures: Insights from a Lakatosian analysis. *Cognitive Science*, *31*, 509-533.

Gunzelmann, G., & Gluck, K. A. (2008). Approaches to modeling the effects of fatigue on cognitive performance. In *Proceedings of the 17th Conference on Behavior Representation in Modeling and Simulation.*

Juvina, I., & Taatgen, N. (2007). Modeling control strategies in the n-back task. In *Proceedings of the 8th International Conference on Cognitive Modeling.*

Lakatos, I. (1970). Falsification and the methodology of scientific research programs. In I. Lakatos & A. Musgrave (Eds.), *Criticism and the growth of knowledge.* Cambridge, UK: Cambridge University Press.

Lehman, J. F., Laird, J., & Rosenbloom, P. (1998). A gentle introduction to Soar, an architecture for human cognition. In S. Sternberg & D. Scarborough (Eds.), *Invitation to Cognitive Science, Volume 4.* Cambridge, MA: MIT Press.

Maanen, L. van, & Rijn, H. van. (2007). An accumulator model of semantic interference. *Cognitive Systems Research*, *8*, 174-181.

Minsky, M. (1967). *Computation – Finite and infinite machines*. Englewood Cliffs, NJ: Prentice Hall.

Newell, A. (1980). Physical symbol systems. *Cognitive Science*, *4*, 135 - 183.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, *107*, 358 - 367.

Rogozhin, Y. (1996). Small universal turing machines. *Theoretical Computer Science*, *168*, 215-240.

Taatgen, N. (2003). Poppering the Newell Test. *Behavioral and Brain Sciences*, *26*, 621 - 622.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *2-42*, 230-265.

Woods, D., & Neary, T. (2009). The complexity of small universal Turing machines: A survey. *Theoretical Computer Science*, *410*, 443-450.

Young, R. M. (2003). Cognitive architectures need compliancy not universality. *Behavioral and Brain Sciences*, *26*, 628.

A Computational Model for Behavioural Monitoring and Cognitive Analysis using Cognitive Models

Alexei Sharpanskykh and Jan Treur
Department of Artificial Intelligence, Vrije Universiteit Amsterdam,
De Boelelaan 1081, NL-1081 HV Amsterdam, The Netherlands
http://www.few.vu.nl/~{sharp, treur}
{sharp, treur}@cs.vu.nl

Abstract

This paper proposes a way in which cognitive models can be exploited in practical applications in the context of Ambient Intelligence. A computational model is introduced in which a cognitive model that addresses some aspects of human functioning is taken as a point of departure. From this cognitive model relationships between cognitive states and behavioural aspects affected by these states are determined. Moreover, representation relations for cognitive states are derived, relating them to external events such as stimuli that can be monitored. Furthermore, by automatic verification of the representation relations on monitoring information the occurrence of cognitive states affecting the human behaviour is determined. In this way the computational model is able to analyse causes of behaviour.

Introduction

One of the interesting areas in which cognitive models can be applied in a practically useful manner is the area of Ambient Intelligence, addressing technology to contribute to personal care for safety, health and wellbeing; e.g., (Aarts, Harwig, and Schuurmans, 2001). Such applications make use of sensor devices to acquire sensor information about humans and their functioning, and of intelligent devices exploiting knowledge for analysis of such information. Based on this, appropriate actions can be undertaken that improve the human's safety, health, and behaviour. Commonly, decisions about such actions are made by these intelligent devices only based on observed behavioural features of the human and her context (cf. Brdiczka, Langet, Maisonnasse, and Crowley, 2009). A risk of such an approach is that the human is guided only at the level of her behaviour and not at the level of the underlying cognitive states causing the behaviour. Such a situation might lead to suggesting the human to suppress behaviour that is entailed by her internal cognitive states, without taking into account these cognitive states (and their causes) themselves.

As an alternative route, the approach put forward in this paper incorporates a cognitive analysis of the internal cognitive states underlying certain behavioural aspects. To this end, a computational model is described, in which a given cognitive model of the human's functioning is exploited. A cognitive model is formalised using the Temporal Trace Language (TTL) (Bosse, Jonker, Meij, Sharpanskykh, and Treur, 2009). In contrast to many existing cognitive modelling approaches based on some form of production rule systems, TTL allows explicit representation of time and complex temporal relations. In particular, using TTL one can specify references to multiple time points, temporal intervals and histories of states, such as, for example, is needed when modelling delayed response behaviour from an external perspective.

By performing cognitive analysis the computational model is able to determine automatically which cognitive states relate to considered behavioural (or performance) aspects of the human, which external events (e.g., stimuli) are required to be monitored to identify these cognitive states (monitoring foci), and how to derive conclusions about the occurrence of cognitive states from such acquired monitoring information. More specifically, monitoring foci are determined by deriving representation relations for the human's cognitive states that play a role in the cognitive model considered. Within Philosophy of Mind a representation relation relates the occurrence of an internal cognitive state property of a human at some time point to the occurrence of other (e.g., external) state properties at the same or at different time points (Kim, 1996). For example, the desire to go outside may be related to an earlier good weather observation. As temporal relations play an important role here, in the computational model these representation relations are expressed as temporal predicate logical specifications. In general, other temporal languages may be used as well. From these temporal expressions externally observable events are derived that are to be monitored. From the monitoring information on these events the computational model verifies the representation expressions, and thus concludes whether or not the human is in such a state. Furthermore, in case an internal state has been identified that may affect the behaviour or performance of the human in a certain way, appropriate actions may be proposed.

The paper is organised as follows. First, the modelling approach is introduced. Then, an example used throughout the paper to illustrate the approach is described. After that an overview of the computational model is provided. More details on this model are described in the following sections: First, a procedure for identifying cognitive states relevant for considered behavioural aspects is described. Then, a procedure for generating representation relations for the relevant cognitive states is described. After that the process of monitoring is considered. Finally, the paper is concluded with a discussion and summary.

Modelling approach

To model the dynamics of cognitive processes with an indication of time, a suitable temporal language is required. In the current paper, to specify temporal relations the Temporal Trace Language (TTL) is used. This reified temporal predicate logical language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. Dynamics are represented in TTL as an evolution of states over time. A state is characterized by a set of state properties expressed over (state) ontology Ont that hold. In TTL state properties are used as terms (denoting objects). To this end the state language is imported in TTL. Sort STATPROP contains names for all state formulae. The set of function symbols of TTL includes $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$: STATPROP x STATPROP $\rightarrow$ STATPROP; not: STATPROP $\rightarrow$ STATPROP, and $\forall$, $\exists$: S^{VARS} x STATPROP $\rightarrow$ STATPROP, of which the counterparts in the state language are Boolean propositional connectives and quantifiers. To represent dynamics of a system sort TIME (a set of time points) and the ordering relation > : TIME x TIME are introduced in TTL. To indicate that some state property holds at some time point the relation at: STATPROP x TIME is introduced. The terms of TTL are constructed by induction in a standard way from variables, constants and function symbols typed with all before-mentioned sorts. The language TTL has the semantics of many-sorted predicate logic. A special software environment has been developed for TTL, featuring a Property Editor for building TTL properties and a Checking Tool that enables automated formal verification of such properties against a set of traces.

The modelling approach presented in this paper adopts a rather general specification format for cognitive models that comprises past-present relationships between cognitive states and between cognitive states and sensor and effector states, formalised by temporal statements expressible within TTL. In this format, for a cognitive state a temporal pattern of past states can be specified, which causes the generation of this state; see also (Jonker and Treur, 2003). A *past-present statement* (abbreviated as a *pp-statement*) is a statement φ of the form $B \Leftrightarrow H$, where the formula H, called the *head* and denoted by head(φ), is a statement of the form at(p, t) for some time point t and state property p, and B, called the *body* and denoted by body(φ), is a past statement for t. A *past statement* for a time point t over state ontology Ont is a temporal statement in TTL, such that each time variable s different from t is restricted to the time interval before t: for every time quantifier for a time variable s a restriction of the form $t > s$ is required within the statement. Sometimes B is called the *definition* of H.

Many types of cognitive models can be expressed in such a past-present format, such as causal models, dynamical system and connectionist models, rule-based models, and models in which memory of past events is used, such as case-based models. In the next section an example of a cognitive model specified in past-present format is given.

Case Study

To illustrate the proposed model a simplified example to support an elderly person in food and medicine intake is used. The following setting is considered. In normal circumstances the interval between two subsequent food intakes by the human during the day is known to be between 2 and 5 hours. When the human is hungry, she goes to the refrigerator and gets and consumes the food she prefers. Sometimes the human feels internal discomfort, which can be soothed by taking medicine X. The box with the medicine lies in a cupboard. There is no food consumption for 2 hours after taking medicine. To maintain a satisfactory health condition of the human, intelligent support is employed, which functionality is described by the computational model presented throughout the paper.

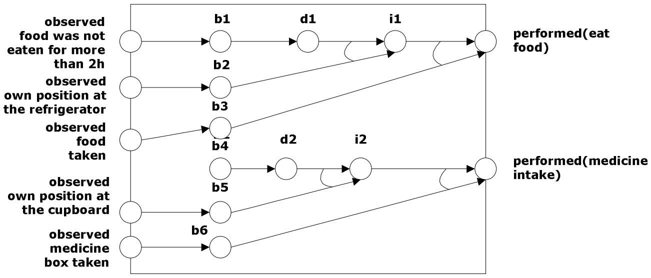

observed food was not eaten for more than 2h
observed own position at the refrigerator
observed food taken
observed own position at the cupboard
observed medicine box taken
b1
d1
i1
b2
b3
b4
d2
i2
b5
b6
performed(eat food)
performed(medicine intake)

Figure 1. Cognitive model for food and medicine intake

The behaviour of the human for this example is considered as goal-directed and is modelled using the BDI (Belief-Desire-Intention) architecture (Rao and Georgeff, 1991). The graphical representation of the cognitive model that produces the human behaviour is given in Figure 1. In this model the beliefs are based on the observations. For example based on the observation that food is taken, the belief b1 that food is taken is created. The desire and intention to have food are denoted by d1 and i1 correspondingly in the model. The desire and intention to take medicine are denoted by d2 and i2 correspondingly. The cognitive model from the example was formalised by the following properties in past-present format:

IP1(c): *General belief generation property*
At any point in time a (persistent) belief state b about c holds *iff* at some time point in the past the human observed c. Formally:

$\exists t2\ [\ t1 > t2\ \&\ at(observed(c), t2)\ ]\ \Leftrightarrow at(b, t1)$

IP2: *Desire d1 generation*
At any point in time the internal state property d1 holds *iff* at some time point in the past b1 held. Formally:

$\exists t4\ [\ t3 > t4\ \&\ at(b1, t4)\ ]\ \Leftrightarrow at(d1, t3)$

IP3: *Intention i1 generation*
At any point in time the internal state property i1 holds *iff* at some time point in the past b2 and d1 held. Formally:

$\exists t6\ [\ t5 > t6\ \&\ at(d1, t6)\ \&\ at(b2, t6)]\ \Leftrightarrow at(i1, t5)$

IP4: *Action eat food generation*
At any point in time the action eat food is performed *iff* at some time point in the past both b3 and i1 held. Formally:

$\exists t8\ [\ t7 > t8\ \&\ at(i1, t8)\ \&\ at(b3, t8)]$

$\Leftrightarrow \text{at(performed(eat food), t7)}$

IP5: *Desire d2 generation*
At any point in time the internal state property d2 holds *iff* at some time point in the past b4 held. Formally:

$\exists \text{t10 [t9 > t10 \& at(b4, t10)]} \Leftrightarrow \text{at(d2, t9)}$

IP6: *Intention i2 generation*
At any point in time the internal state property i2 holds *iff* at some time point in the past b5 and d2 held. Formally:

$\exists \text{t12 [t11 > t12 \& at(d2, t12) \& at(b5, t12)]} \Leftrightarrow \text{at(i2, t11)}$

IP7: *Action medicine intake generation*
At any point in time the action medicine intake is performed *iff* at some time point in the past both b6 and i2 held. Formally:

$\exists \text{t14 [t13 > t14 \& at(i2, t14) \& at(b6, t14)]} \Leftrightarrow \text{at(performed(medicine intake), t13)}$

Cognitive Analysis: Overview

First, a set of goals is defined on the human's states and behaviour. These goals may concern, for example, the human's well-being or the quality of performance in task execution. The goal for the case study is to maintain a satisfactory health condition of the human. Each goal is refined into more specific criteria that should hold for the human's functioning. In particular, for the case study the goal is refined into three criteria:

(1) *food is consumed every 5 hours (at latest) during the day*;
(2) *after the medicine is taken, no food consumption during the following 2 hours occurs*;
(3) *after 3 hours from the last food intake no medicine intake occurs.*

Based on the criteria expressions, a set of output states (called *an output focus*) and a set of internal (cognitive) states (called *an internal focus*) of the human are determined, which are used for establishing the satisfaction of the criteria. For the case study the output focus consists of the states performed(eat food) and performed(medicine intake).

A cognitive model of the human defines relations between an output state and internal states which cause the generation of the output state. The latter provide a more in depth understanding of why certain behaviours (may) occur. In general, using a cognitive model one can determine a minimal specification that comprises temporal relations to internal states, which provides necessary and sufficient conditions on internal states to ensure the generation of an output state. An automated procedure to generate such specifications is considered in the next section. Such a specification is a useful means for prediction of behaviour. That is, if an essential part of a specification becomes satisfied (e.g., when some important internal state(s) hold(s)), the possibility that the corresponding output state will be generated increases significantly. If such an output is not desired, actions can be proposed in a knowledgeable manner, based on an in depth understanding of the internal states causing the behaviour. Thus, the essential internal states (called *predictors for an output*) from specifications for the states in the output focus should be added to the internal focus.

Normally states in an internal focus cannot be observed directly. Therefore, representation relations are to be established between these states and externally observable states of the human (i.e., the representational content should be defined for each internal state in focus). Representation relations are derived from the cognitive model representation as shown in a section below and usually have the form of more complex temporal expressions over externally observable states. To detect occurrence of an internal state, the corresponding representational content should be monitored constantly, which is considered in a section later in this paper.

Generating Predictors for Output States

One of the tasks is the identification of (internal) predictors for outputs. A predictor(s) for a particular output can be identified based on a specification of human's internal dynamics that ensures the generation of the output. In general, more than one specification can be identified, which is minimal (in terms of numbers of internal states and relations between them), however sufficient for the generation of a particular output. Below an automated procedure for the identification of all possible minimal specifications for an output state based on a cognitive model is given. The rough idea underlying the procedure is the following. Suppose for a certain output state property p the pp-statement B ⇔ at(p, t) is available. Moreover, suppose that in B only two atoms of the form at(p1, t1) and at(p2, t2) with internal states p1 and p2 occur, whereas as part of the cognitive model also specifications B1 ⇔ at(p1, t1) and B2 ⇔ at(p2, t2) are available. Then, within B the atoms can be replaced (by substitution) by the formula B1 and B2. Thus, at(p, t) may be related by equivalence to four specifications:

$\text{B} \Leftrightarrow \text{at(p, t)}$ $\quad$ $\text{B[B2/at(p2, t2)]} \Leftrightarrow \text{at(p, t)}$

$\text{B[B1/at(p1, t1)]} \Leftrightarrow \text{at(p, t)}$ $\quad$ $\text{B[B1/at(p1, t1), B2/at(p2, t2)]} \Leftrightarrow \text{at(p, t)}$

Here for any formula C the expression C[x/y] denotes the formula C transformed by substituting x for y.

```
Algorithm: GENERATE-MINIMAL-SPECS-FOR-OUTPUT
Input: Cognitive model X; output state in focus specified by at(s, t)
Output: All possible minimal specifications for at(s, t) in list L
1  Let L be a list containing at(s, t), and let $\delta p$, $\delta$ be empty substitution lists.
2  For each formula $\varphi_l \in L$: $at(a_i, t) \leftrightarrow \psi i_p(at_1,\ldots, at_m)$ identify $\delta_i = \{at_k/body(\varphi_k)$ such that $\varphi_k \in X$ and $head(\varphi_k)=at_k\}$. Then $\delta$ is obtained as a union of $\delta_i$ for all formulae from L.
3  $\delta = \delta \setminus \delta p$
4  if $\delta$ is empty, finish.
5  For each formula $\varphi_l \in L$ obtain a set of formulae by all possible combinations of substitution elements from $\delta$ applied to $\varphi_l$. Add all identified sets to L.
6  $\delta p = \delta p \cup \delta$, proceed to step 2.
```

For each generated specification the following measures can be calculated:

(1) The measure of undesirability indicating how undesirable is the human's state, described by the generated specification. It also reflects the confidence degree of producing an undesirable output from the generated specification.

(2) The minimum and maximum time before the generation of the output state. This measure is critical for timely intervention in human's activities.

These measures serve as heuristics for choosing one of the generated specifications. To facilitate the choice, constrains on the measures may be defined, which ensure that an intervention occurs only when a considerable undesirability degree is determined, but also the minimum time before the undesirable output is above some acceptable threshold. To calculate the measure (1), the degree of undesirability is associated with each output state of the cognitive model (i.e., a number from the interval [0, 1] that expresses how undesirable is the state). Then, it is determined which output states from the cognitive specification can be potentially generated, given that the bodies of the formulae from the generated specification are evaluated to TRUE. This is done by executing the cognitive specification with $\mathrm{body}(\varphi_i)$ = TRUE for all φ_i from the generated specification. Then, the measure of undesirability is calculated as the average over the degrees of undesirability of the identified output states, which can be potentially generated. The measures (2) can be calculated when numerical timing relations are defined in the properties of a cognitive specification.

For the case study from the automatically generated specifications that ensure the creation of the state performed(eat food) the one expressed by property IP4 is chosen. This specification has the highest confidence degree of producing the output (equal to the undesirability measure of the state performed(eat food)), when it is undesirable. It is assumed that the time interval t7-t8 in IP4 is sufficient for an intervention. The predictor state from the chosen specification is i1, as in the most cases it is generated earlier than the state b3. Thus, i1 is included in the internal focus. By a similar line of reasoning, the specification expressed by property IP7 is chosen, in which i2 is the predictor state included into the internal focus. Thus, the internal focus for the cognitive model is the set {i1, i2}.

Representation Relations

A next step is the identification of representation relations for cognitive states from a cognitive model for the human. A representation relation for an internal state property p relates the occurrence of p to a specification Φ that comprises a set of state properties and temporal (or causal) relations between them. In such a case it is said that p *represents* Φ, or Φ describes *representational content* of p. In this section an automated approach to identify representation relations for cognitive states from a cognitive model is described.

The representational content considered backward in time is specified by a history (i.e., a specification that comprises temporal (or causal) relations on past states) that relates to the creation of some cognitive state. In the literature on Philosophy of Mind different approaches to defining representation relations have been put forward (cf. Kim, 1996). For example, according to the classical causal/correlation approach, the representational content of an internal state property is given by a one-to-one mapping to an external state property. The application of this approach is limited to simple types of behaviour (e.g., purely reactive behaviour). In cases when an internal property represents a more complex temporal combination of state properties, other approaches have to be used. For example, the temporal-interactivist approach (cf. Jonker and Treur, 2003) allows defining representation relations by referring to multiple (partially) temporally ordered interaction state properties; i.e., input (sensor) and output (effector) state properties over time.

To automate the process of representation relation identification based on this idea, a procedure has been developed. To apply this procedure, cognitive specification is required to be stratified. This means that there is a partition of the specification $\Pi = \Pi_1 \cup \ldots \cup \Pi_n$ into disjoint subsets such that the following condition holds: for $i > 1$: if a subformula $\mathrm{at}(\varphi, t)$ occurs in a body of a statement in Π_i, then it has a definition within $\cup_{j<i} \Pi_j$.

Algorithm: GENERATE-REPRESENTATION-RELATION

```
Input:   Cognitive specification X; cognitive state specified by at(s, t), for which the representation relation is to be identified
Output: Representation relation for at(s, t)
1 Stratify X:
1.1 Define the set of formulae of the first stratum (h=1) as
{$\varphi_i$: at($a_i$, t) $\leftrightarrow$ $\psi i_p(at_1,\ldots, at_m) \in X \mid \forall k\ m \geq k \geq 1\ at_k$ is expressed using InputOnt}; proceed with h=2.
    1.2  The set of formulae for stratum h is identified as
    {$\varphi_i$: at($a_i$, t) $\leftrightarrow \psi i_p(at_1,\ldots, at_m) \in X \mid \forall k\ m \geq k \geq 1\ \exists l\ l < h\ \exists \psi \in$ STRATUM(X, l) AND head($\psi$) = $at_k$ AND $\exists j\ m \geq j \geq 1\ \exists \xi \in$ STRATUM(X, h-1) AND head($\xi$)=$at_j$ }; proceed with h=h+1.
    1.3  Until a formula of X exists not allocated to a stratum, perform 1.2.
2 Create the stratified specification X' by selecting from X only the formulae of the strata with the number i < k, where k is the number of the stratum, in which at(s, t) is defined. Add the definition of at(s, t) from X to X'.
3 Replace each formula of the highest stratum n of X' $\varphi_i$: at($a_i$, t) $\leftrightarrow$ $\psi i_p(at_1,\ldots, at_m)$ by $\varphi_l\ \delta$ with renaming of temporal variables if required, where $\delta$ = {$at_k$\ body($\varphi_k$) such that $\varphi_k \in$ X' and head($\varphi_k$)=$at_k$}. Further, remove all formulae { $\varphi \in$ STRATUM(X', n-1) | $\exists \psi \in$ STRATUM(X', n) AND head($\varphi$) is a subformula of the body($\psi$)})
4 Append the formulae of the stratum n to the stratum n-1, which now becomes the highest stratum (i.e, n=n-1).
5 Until n>1, perform steps 3 and 4. The obtained specification with one stratum (n=1) is the representation relation specification for at(s, t)
```

In Step 3 subformulae of each formula of the highest stratum n of X' are replaced by their definitions, provided in lower strata. Then, the formulae of n-1 stratum used for the replacement are eliminated from X'. As result of such a replacement and elimination, X' contains n-1 strata (Step 4). Steps 3 and 4 are performed until X' contains one stratum only. In this case X' consists of a formula φ defining the representational content for at(s, t), i.e., head(φ) is at(s, t) and body(φ) is a formula expressed over interaction states and (temporal) relations between them.

In the following it is shown how this algorithm is applied for identifying the representational content for state i1 from the internal focus from the case study. By performing Step 1 the specification of the cognitive model given above is automatically stratified as follows: stratum 1: {IP1((own_position_refrigerator), IP1(food_not_eaten_more_than_2h), IP1(own_position_cupboard), IP1(medicine_box_taken)}; stratum 2: {IP2, IP5}; stratum 3: {IP3, IP6}; stratum 4: {IP4, IP7}.

By Step 2 the properties IP4, IP5, IP6, IP7 are eliminated as unnecessary for determining the representational content of i1. Further, in Step 3 we proceed with the property IP3 of the highest stratum (3) that defines the internal state i1.

$$\exists t6\ [\ t5 > t6\ \&\ at(d1,\ t6)\ \&\ at(b2,\ t6)]\ \Leftrightarrow\ at(i1,\ t5)$$

In Step 3 the property IP8 is obtained by replacing d1 and b2 state properties in IP3 by their definitions with renaming of temporal variables:

$$\exists t6\ [\ t5 > t6\ \&\ \exists t4\ [\ t6 > t4\ \&\ at(b1,\ t4)\ ]\ \&\ \exists t2\ [\ t6 > t2\ \&\ at(observed(\text{own\_position\_refrigerator}),\ t2)\ ]\ \ ]\ \Leftrightarrow\ at(i1,\ t5)$$

Further, the properties IP3, IP2 and IP1(own_position_refrigerator) are removed from the specification and the property IP8 is added to the stratum 2. Then, IP9 is obtained by replacing b1 in IP8 by its definition:

$$\exists t6\ [\ t5 > t6\ \&\ \exists t4\ [\ t6 > t4\ \&\ \exists t15\ [\ t4 > t15\ \&\ at(observed(\text{food\_not\_eaten\_more\_than\_2h}),\ t15)\ ]\ ]\ \&\ \exists t2\ [\ t6 > t2\ \&\ at(observed(\text{own\_position\_refrigerator}),\ t2)\ ]\ \ ]\ \Leftrightarrow\ at(i1,\ t5)$$

After that the properties IP8 and IP1(food_not_eaten_more_than_2h) are removed from the specification and IP9 becomes the only property of the stratum 1. Thus, IP9 defines the representational content for the state i1 that occurs at any time point t5.

Similarly, the representational content for the other state from the internal focus i2 is identified as:

$$\exists t12\ [\ t11 > t12\ \&\ \exists t16\ [\ t12 > t16\ \&\ at(observed(\text{own\_position\_cupboard}),\ t16)\ ]\ ]\ \Leftrightarrow\ at(i2,\ t11)$$

The algorithm has been implemented in Java. The overall time complexity of the algorithm for the worst case is $O(|X|^2)$, where $|X|$ is the length of a cognitive specification X.

Behavioural Monitoring

To support the monitoring process, it is useful to decompose a representational content expression into atomic subformulae that describe particular interaction and world events. The subformulae are determined in a top-down manner, following the nested structure of the overall formula:

$$\text{monitor\_focus}(F)\ \rightarrow\ \text{in\_focus}(F)$$

$$\text{in\_focus}(E)\ \wedge\ \text{is\_composed\_of}(E,C,E1,E2)\ \rightarrow\ \text{in\_focus}(E1)\ \wedge\ \text{in\_focus}(E2)$$

Here is_composed_of(E,C,E1,E2) indicates that E is an expression obtained from subexpressions E1 and E2 by a logical operator C (i.e., and, or, implies, not, forall, exists). At each decomposition step subexpressions representing events are added to the list of foci that are used for monitoring. This list augmented by the foci on the states from the output focus is used for monitoring. For the case study from the identified representation content for i1 and i2 the following atomic monitoring foci were derived:

observed(food_not_eaten_more_than_2h)
observed(own_position_refrigerator)
observed(own_position_cupboard)

Furthermore, the information on the states in the output and internal foci, on the chosen predictors for the output states, and on the identified representation relations is used to constantly monitor. As soon as a an event from the atomic monitoring foci occurs, the component initiates automated verification of the corresponding representational content property on the history of the events in focus occurred so far. The automatic verification is performed using the TTL Checker tool (for the details on the verification algorithm see (Bosse et al, 2009)). For the case study such a history (or a trace) was created using the LEADSTO simulation tool (Bosse et al, 2007).

Another task is to ensure that the goal criteria hold. The satisfaction of the criteria is checked using the TTL Checker tool. Furthermore, to prevent the violation of a criterion promptly, information related to the prediction of behaviour (i.e., predictors for outputs) can be used. More specifically, if the internal states-predictors for a set of output states O hold, and some behaviour or performance criterion is violated under O, then an intervention in human activities is required. The type of intervention may be defined separately for each criterion. In particular, for the case study as soon as the occurrence of the prediction states i1 and i2 is established, the violation of the criteria identified previously is determined under the condition that the predicted outputs hold. To prevent the violation of the criteria, the following intervention rules are specified:

(1) If the human did not consume food during last 5 hours, then inform the human about the necessary food intake. Formally:

$$\forall t1\ current\_time(t1)\ \&\ \neg\exists t2\ t1\text{-}300 \leq t2 < t1\ belief(holds\_at(performed(eat\ food),\ t2),\ pos)\ \Rightarrow\ \text{to\_be\_communicated\_to}(\text{‘Meal time’},\ pos,\ Human)$$

(2) If the human took medicine X less than 2 hours ago (time point t2 in minutes) and the existence of the predictor i1 is established, then inform the human that she still needs to wait (120- t2) minutes for taking medicine. Formally:

$$\forall t1\ current\_time(t1)\ \&\ \exists t2\ t1\text{-}120 < t2\ belief(holds\_at(performed(medicine\ intake),\ t2),\ pos)\ \&\ at(i1,\ t1)\ \Rightarrow\ \text{to\_be\_communicated\_to}(\text{‘Please wait 120-t2 minutes more’},\ pos,\ Human)$$

(3) If the human did not consume food during last 3 hours and the existence of the predictor i2 is established, inform the human that she better eats first. Formally:
∀t1 current_time(t1) & ¬∃t2 t1-180 ≤ t2 < t1
belief(holds_at(performed(eat food), t2), pos) & at(i2, t1)
⇒ to_be_communicated_to('Please eat first', pos, Human)

Discussion and Conclusions

In this paper a computational model was presented incorporating a more in depth analysis based on a cognitive model of a human's functioning. Having such a cognitive model allows relating certain behavioural or performance aspects that are considered, to underlying cognitive states causing these aspects. Often cognitive models are used either by performing simulation, or by temporal reasoning methods; e.g. (Port and van Gelder, 1995). In this paper a third way of using such models is introduced, namely by deriving more indirect relations from these models. Such an approach can be viewed as a form of knowledge compilation (Cadoli and Donini, 1997) in a pre-processing phase, so that the main processing phase is less intensive from the computational point of view. Such a form of automated knowledge compilation occurs in two ways: first, to derive the relationships between considered behaviour or performance aspects to the relevant internal cognitive states, and next to relate such cognitive states to observable events (monitoring foci). These monitoring foci are determined from the cognitive model by automatically deriving representation relations for cognitive states in the form of temporal specifications. From these temporal expressions the events are derived that are to be monitored, and from the monitoring information on these events the representation expressions are verified automatically.

A wide range of existing ambient intelligence applications is formalised using production rules (cf. Christensen, 2002) and if-then statements. Two important advantages of such rules are modelling simplicity and executability. However, such formalism is not suitable for expressing more sophisticated forms of temporal relations, which can be specified using the TTL language. In particular, references to multiple time points possible in TTL are necessary for modelling forms of behaviour more complex than stimulus-response (e.g., to refer to memory states in delayed-response behavioural specifications). Furthermore, TTL allows representing temporal intervals as in the following property: 'if the human was sleeping for x hours and x > 4h and s/he did not take the medicine A during 2 hours after being awake, then support will be provided to the human'. Moreover, using TTL one can refer to histories of states, for example to express that a medicine improves the health condition of a patient; in this case the health conditions in traces with and without the medicine intake are compared.

Another popular approach to formalise recognition and prediction of human behaviour is by using Hidden Markov Models (HMM) (e.g., Sanchez et al., 2007). In HMM-based approaches known to the authors, recognition of human activities is based on contextual information of the activity execution only; no cognitive or (gradual) preparation states that precede actual execution of activities are considered. As indicated in (Sanchez et al., 2007) a choice of relevant contextual variables for HMMs is not simple and every additional variable causes a significant increase in the complexity of the recognition algorithm. Knowledge of cognitive dynamics that causes particular behaviour would provide more justification and support for the choice of variables relevant for this behaviour. Furthermore, as pointed in (Brdiczka et al., 2009) for high quality behaviour recognition a large corpus of training data is needed. The computational costs of the pre-processing (knowledge compilation) phase of the approach proposed in this paper are much lower (polynomial in the size of the specification). Also, no model training is required. However, the proposed approach relies heavily on the validity of cognitive models.

In the future, cases will be elaborated, in which cognitive models based on diverse cognitive frameworks and architectures will be used.

References

Aarts, E., Harwig, R. and Schuurmans, M. (2001) Ambient Intelligence. In: P. Denning (ed.), The Invisible Future, McGraw Hill, New York, p. 235-250.

Bosse, T., Jonker, C. M., van der Meij, L., and Treur, J. (2007). A Language and Environment for Analysis of Dynamics by Simulation. International Journal of Artificial Intelligence Tools, vol. 16, 435-464.

Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A, and Treur, J. (2009). Specification and Verification of Dynamics in Agent Models. Int. J. of Cooperative Information Systems, 18(1), pp. 167-193.

Brdiczka, O., Langet, M., Maisonnasse, J., Crowley, J. L. (2009). Detecting human behavior models from multimodal observation in a smart home. In: IEEE Transactions on Automation Science and Engineering, 6(3), IEEE Computer Society Press.

Cadoli, M., Donini, F. M. (1997). A Survey on Knowledge Compilation. AI Communications, 10(3-4), p. 137-150.

Christensen, H.B. (2002). Using Logic Programming to Detect Activities in Pervasive Healthcare. In International Conference on Logic Programming, Springer, vol. 2401, pp.21 - 436

Jonker, C.M., and Treur, J. (2003). A Temporal-Interactivist Perspective on the Dynamics of Mental States. Cognitive Systems Research Journal, vol. 4, 137-155.

Kim, J. (1996) Philosophy of Mind. Westview Press.

Port R.F., van Gelder, T. (eds.) (1995). Mind as Motion: Explorations in the Dynamics of Cognition. MIT Press, Cambridge, Mass.

Rao, A., Georgeff, M.P. (1991). Modeling rational agents within a bdi-architecture. In Proc. of 2th Int. Conf. on Principles of Knowledge Repr. and Reasoning, 473-484.

Sanchez, D., Tentori, M., Favela, J. (2007). Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments. In: Proc. 8th Int. Mexican Conf. Current Trends in Computer Science, IEEE CS Press, pp. 33–40.

Checking Chess Checks with Chunks: A Model of Simple Check Detection

Richard Ll. Smith (rls@hwyl.org)
School of Social Sciences, Brunel University,
Uxbridge, UB8 3PH UK

Fernand Gobet (fernand.gobet@brunel.ac.uk)
School of Social Sciences, Brunel University,
Uxbridge, UB8 3PH UK

Peter C.R. Lane (peter.lane@bcs.org.uk)
School of Computer Science, University of Hertfordshire,
Hatfield, AL10 9AB UK

Abstract

The procedure by which humans identify checks in check positions is not well understood. We report here our experience in modelling this process with CHREST, a general-purpose cognitive model that has previously successfully captured a variety of attention- and perception-related phenomena. We have attempted to reproduce the results of an experiment investigating the ability of humans to determine checks in simple chess positions. We propose a specific model of how humans perform this experiment, and show that, given certain reasonable assumptions, CHREST can follow this model to create a good reproduction of the data.

Keywords: CHREST; cognitive model; cognitive architecture; chess; check detection

Motivation

In studying the general phenomenon of human perception, we have looked at the specific task of perceiving checks in the game of chess. This task involves a player being presented with a chess position with a requirement to determine whether or not the player's king is being threatened by another piece.

Experiments on human subjects have provided data about how well they perform this task, but we have no good model of how the underlying psychological processes work in this situation. Identifying threats in games is a complex task which explores the process of visual attention when guided by interpretation of higher goals. Understanding these processes may help shed light on a variety of aspects of attention and perception.

Although understanding these processes is a desirable goal in itself, we are also interested in modelling this process as part of a larger project to produce a cognitive model which, whilst operating under human constraints, plays chess in a human-like way. Successfully modelling the check perception process would be a step towards this aim, as well as a verification of the parameters of the model itself.

Background

Saariluoma conducted a series of experiments (Saariluoma, 1984) relating to the perception abilities of humans through the medium of chess. We are concerned with one experiment in particular: this was to measure how quickly players could determine, given a chess position consisting of a white king and one other black piece, whether or not the king was in check. The subjects of the experiment included chess players with a mix of skill levels: two complete beginners, three unrated amateurs, two experts (ELO rating around 2,000 points) and a high-class international Grand Master.

Analysis of the results of this experiment showed a very significant ($p < 0.001$) correlation of reaction speed with chess ability: the Grand Master took around a third of the time to return a decision compared to the mean of the reaction times of the beginners.

Saariluoma noted that experienced players must perform at least some of the operations involved in the task more quickly than less experienced players, but did not predict which ones. It is known that a few of these processes are improved with practice, such as recognition of pieces (Saariluoma, 1984), speed of making moves in the mind's eye (Church & Church, 1977; Milojkovic, 1982); these have been addressed in our model (see below).

Whilst it is plausible that other cognitive processes involved may be improved through practice, we hypothesise that the greater relevant knowledge acquired by more experienced players should account for the main part of the remaining difference.

It is difficult to test this hypothesis on human subjects due to the obvious challenges of controlling for the amount of domain knowledge acquired and isolating the relevant processes. In order to investigate this hypothesis, the use of a cognitive model would be helpful in order to manipulate these factors directly.

A successful model should be able to demonstrate the superiority of experts over novices in the check detection task, and explain why.

CHREST

CHREST (Chunk Hierarchy and REtrieval STructures) (Gobet et al., 2001) is a general-purpose cognitive architecture designed to simulate certain aspects of the human mind[1], including, to the extent that these have been measured or can

[1] For information on CHREST beyond what is presented here, the interested reader is referred to the CHREST website at http://chrest.info

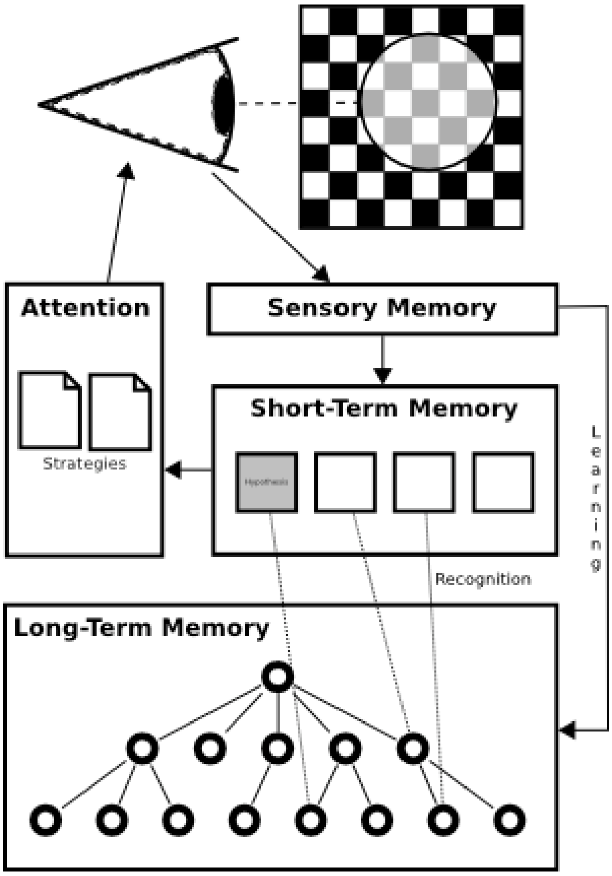


Figure 1: An overview of the main components of the CHREST cognitive architecture.

be estimated, its limitations (an important requirement of a model that aspires to simulate human cognition is that it not make use of any abilities in excess of that of a human (Simon, 1969)).

CHREST has previously been shown to be a successful model of the mind in domains as diverse as physics representation (Lane, Cheng, & Gobet, 2000), language acquisition (Jones, Gobet, & Pine, 2005; Freudenthal, Pine, Aguado-Orea, & Gobet, 2007), and ageing (Smith, Gobet, & Lane, 2007); however, the simulation of perception and memory in chess (de Groot & Gobet, 1996; Gobet & Simon, 2000; Gobet & Waters, 2003) has been CHREST's most studied application.

Figure 1 shows a top-level overview of the CHREST architecture. It simulates the main divisions of memory in humans as is generally agreed upon (Baddeley, 1990): a short-term memory (STM) store, and a long-term memory (LTM) store. In addition, it has an advanced perception/attention system.

CHREST's memory system is based on chunking theory (Chase & Simon, 1973), which holds that information in the human mind is stored as *chunks*. Chunks are discrete collections of *features* that have some meaning when grouped together. In the domain of chess, the features that make up chunks in CHREST are man-on-square combinations such as "White king on square g1". A chunk containing this feature, and representing a standard castled white king could be represented as the set: $\{\text{Kg1}, \text{Rf1}, \text{Pf2}, \text{Pg2}, \text{Ph2}\}$, where the first letter is the first letter of the piece's name.

CHREST's LTM is made up of a hierarchical network of these chunks. Its organisation is primarily tree-based, though the presence of semantic links adds a graph-like flavour. Knowledge is added to LTM through two main learning processes. When a new pattern is encountered, it is compared to previously-learnt chunks: if it does not match any known chunk, then a new chunk is created containing some of the new information (*discrimination*); if it does match a known chunk, then some of the information in the pattern is added to that chunk (*familiarisation*).

CHREST's STM has a capacity of up to four chunks. There is good evidence that this is the approximate STM capacity of young adult humans (Luck & Vogel, 1997; Cowan, 2001). These chunks are references to chunks held in LTM (again, as indicated by recent research (Gobet et al., 2001)).

Attention in CHREST is represented through simulated eye movements (this is a slight simplification of the human attention system, but this approach is relatively easy to simulate and its output can be verified against recorded human data).

CHREST's attention is directed through information previously learnt and added to LTM, and a set of heuristics. The basic heuristics, such as "look at the centre of the board", "look at objects grouped together", and "follow a potential move from an observed piece" guide the perception of basic patterns, which are incorporated into LTM as chunks.

As more information-rich chunks are acquired, this learnt information is used to guide the focus of attention. When an observed pattern is recognised as a previously-learnt chunk, a reference to the chunk is placed in STM, and this selected information may be used to provide a new focus. If a chunk referenced in STM is linked to another chunk in LTM, then CHREST's attention is directed towards locations containing objects in the linked chunk that are disjoint from the objects recorded in the recognised chunk. This process allows CHREST to focus on the distinguishing features of a scene.

Patterns, then, are perceived on the board according to previously-learnt chunks, and chunks are built up out of perceived patterns; this interplay between the learning cycle and the perception system results in complex emergent behaviour. In previous work (de Groot & Gobet, 1996), the eye movements generated by CHREST during a simulated presentation of a chess position have been shown to be comparable to those of Masters.

See (Lane, Gobet, & Smith, 2008) for more details of the attention system.

CHREST Configuration

The version of CHREST used for these experiments was the 3.0 beta version. The code base of this version has been mostly rewritten from the 2.x version. It represents a substantial evolution of the model and a major step forward towards a full release of a complete CHREST 3. As well as being more flexible and better able to make use of modern computing technology, this version of CHREST has a number of new features: notably, it understands chess at a deeper level, includes a customised experiment framework which automatically performs and reports on sets of experiments, and has the ability to perceive and learn from full games rather than selected positions.

In order to simulate the variety of individuals employed in the human experiments, a series of CHREST subjects[2] with varying LTM sizes was produced: 20 each with a network size of one of 100, 1,000, 10,000, 100,000 nodes. (These different network sizes represented players of different skill level).

Most of the training of these subjects was carried out using a set of 10,000 games played during 2008 between players with ELO ratings of above 2000. Each subject was allowed to learn from the state of the game board at random intervals during simulated play-throughs of the games until the required network size was reached.

In addition, each subject was specifically trained on boards with a king and one other piece (all possible configurations of this type were produced to make up a training set). A total of 10% of each subject's LTM network was generated in this way, reflecting the fact that checking positions are very common in rapid and speed chess games, which most chess players use as a form of practice (Gobet & Campitelli, 2007). This figure is necessarily only an estimate of real-life behaviour due to a lack of empirical data at this time, but it was estimated in advance and not fitted to the model's result.

As in the Saariluoma experiment, 60 chess positions were generated for testing. A white king was placed on the board, along with a black queen, rook, bishop, or knight. For each position, the locations of the pieces were randomised, with the constraint that the king was placed on a square in which it was in check in half of the positions.

Timings played an important part in the experiments; time was one limiting factor for CHREST's perceptual cycle, and the time taken for CHREST to decide if a position contained a check was the main dependent variable in the experiments.

CHREST uses an internal clock which accumulates the processing times of simulated operations. These times are (where possible) taken from human experiments, or otherwise (where experimentation has not yet been possible), taken from sensible estimates (see (de Groot & Gobet, 1996) for details).

Unless otherwise noted, timings used were the standard timings which have evolved in CHREST:

- A constant 200 ms was added to all trials to simulate initial reaction to the stimulus, motor preparation, and motor response (i.e. pressing the button). Visual reaction times have been recorded as in the region 180 to 200 ms for university-age students (Brebner & Welford, 1980), though increasing with age (Welford, 1977).

- Saariluoma found, in a previous experiment (Saariluoma, 1984), that novices were slower than experts in recognising chessmen. The mean difference between the two groups was 57.1 ms; this value was added to the clock as time taken to recognise each piece for the 100 and 1,000-node network (this division was slightly arbitrary as it might be expected that the delay would be a gradient rather than binary, but we have no better data).

- From their analysis of experimental results, de Groot and Gobet (1996) proposed definite parameters for the time required to move pieces in the mind's eye. These parameters consisted of a *base* time, the time taken to begin making a move, and a *square* time, the time taken per square to move a piece. The first was estimated as 100 ms, and the second as 50 ms for experts, and 100 ms for novices. We have used these same values.

Modelling Check Perception

We have described the domain of interest, that of the human process of perceiving and determining checks, and the general-purpose cognitive architecture that we are using to investigate it. Now we consider how to specifically adapt the model to the domain.

It has already been shown that the memorisation of chess positions under human constraints — see (de Groot, 1978) — can be improved through prior knowledge of chess positions. We propose that the process of determining whether a king is in check from another piece, given that the location and types of both pieces have been established, benefits from the presence in memory of previously encoded chunks of chess positions.

Our hypothesis for the superiority of experts over novices in detecting checks lies in chunking theory (Chase & Simon, 1973). Following previous work (Gobet & Jansen, 1994), we hypothesise that links are formed between the learnt visuo-spatial chunks and more abstract knowledge; for example, moves associated with the chunk, the goodness of the chunk in positional terms, and, of interest with respect to our particular domain, whether the chunk contains a check or not.

Our model, set up as above, simulates the experiment as follows. The simulated subjects were presented with a test position and allowed to perceive it until they had observed two pieces. Once this was achieved, they attempted to decide whether the king was being attacked by the other piece.

If the two pieces were recognised as a chunk already stored in their LTM, then the subject was assumed to be able to quickly (we have assumed 10 ms — the standard time taken to traverse an LTM link) identify whether the position was a check or not. Essentially, the subject would have exhibited automaticity (Shiffrin & Schneider, 1977).

[2] We use the term subject here to distinguish the computational instantiation of a model (complete with data) from the theoretical model

If no such chunk was recognised, then a simulated attempt to determine check was carried out, by 'moving' the non-king piece towards the king with simulated eye movements (previous work has demonstrated this proportionality to distance effect (Church & Church, 1977)), and checking this by 'moving' the king towards the other piece in a similar fashion. We assume that double verification of the check relation occurs here, but not when a chunk has been identified, as mentally moving pieces in the mind's eye is more likely to generate errors than when a pattern has been recognized (for a discussion of the difficulty of generating moves in the mind's eye and playing blindfold chess, see (Saariluoma, 1984), and (Campitelli & Gobet, 2005)).

Experiment 1: Standard Perceptual Strategies

For the first experiment, our initial model made use of standard strategies to guide perception when studying each position: i.e. the use of LTM guidance, and fall-back general-purpose heuristics as described above and used in the training of the networks. These results are shown in table 1 and figure 2. The human data collected in (Saariluoma, 1984) are shown for comparison[3].

The results demonstrate some success of the approach in modelling the data ($r^2 = 0.92$): specifically, they show the required qualitative interaction of LTM size (acquired knowledge) with time, and are within around 200 ms of the times of the novice players. However, the results diverge from the human data considerably when considering performance of expert-level and above.

These strategies have previously been shown to be an accurate model of expert eye movements in perceiving scenes, but they clearly do not fully capture behaviour in this domain.

Table 1: Time taken to make a check perception decision as simulated by CHREST for players of different skill levels using standard perceptual strategies (Experiment 1).

LTM Network Size (nodes)	Time Taken (ms)
100	1,705
1,000	1,403
10,000	1,301
100,000	1,068

Experiment 2: Simplified Perceptual Strategy

Following the results of the first experiment, we hypothesised that players are using their meta-knowledge about the problem to re-orient their perceptual strategies. As the model consistently overestimated the time taken, we suspected that the perceptual strategies used by CHREST were too involved and that humans used a simpler strategy.

Our revised model was that the subject would automatically perceive a man on the board using far peripheral vision and direct their attention towards it. The subject would then make use of their near peripheral vision (set at ±2 squares from the focal point) to recognise another piece if one was in range. If no piece was in range, the player would detect the other piece using their far peripheral vision, and refocus on that point following a saccade (thus, making one, or a maximum of two, eye fixations; in the previous experiment, the focus could be directed towards empty squares).

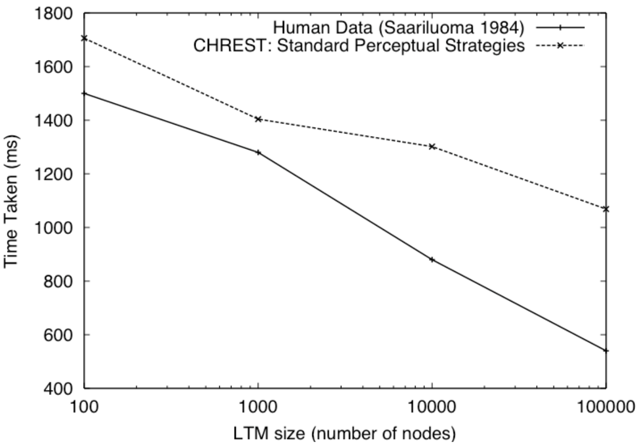


Figure 2: Time taken to make a check perception decision as simulated by CHREST for players of different skill levels using standard perceptual strategies (Experiment 1). The human data are shown for comparison.

The results of re-running the experiment using this strategy are shown in table 2 and figure 3. This time the results are a significantly better fit to the human data ($r^2 = 0.94$), again showing the qualitative interaction, but matching the data quantitatively to within 200 ms at worst. In this experiment, however, the results better match the data for advanced players rather than novices.

Table 2: Time taken to make a check perception decision as simulated by CHREST for players of different skill levels using standard perceptual strategies (Experiment 2).

LTM Network Size (nodes)	Time Taken (ms)
100	1,320
1,000	1,010
10,000	883
100,000	606

Discussion of Results

Before discussing the results, we note that there are some limitations to the study and suggest some other reasons for caution in interpreting the results.

We have assumed above that our choices of four network sizes — {100, 1,000, 10,000, 100,000} — correspond to Saariluoma's categorisation of his subjects — {Fourth

[3]The exact human data were not available and so have been read from the graph supplied in (Saariluoma, 1984)

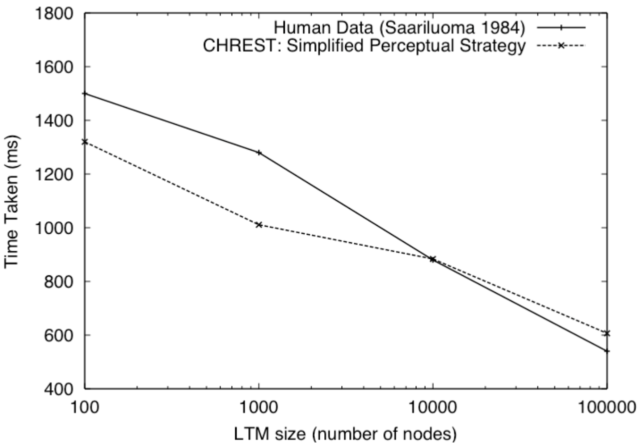


Figure 3: Time taken to make a check perception decision as simulated by CHREST for players of different skill levels using standard perceptual strategies (Experiment 2). The human data are shown for comparison.

Class, Second, Class, Experts, Grand Masters} — but this is not clear. For obvious reasons it is not possible to directly measure the number of chunks learnt by a human subject, so we used a logarithmic progression as an approximation. 100 chunks is probably too many for a beginner who is still learning how the pieces move (though it makes no difference to the result as no chunks were recognised by the 100-node networks) and estimates of the number of chunks learnt by a Grand Master differ, from 100,000 (Simon & Gilmartin, 1973) to 300,000 (Gobet & Simon, 2000) (but this may not be an issue as we argue below that larger networks will probably not show much relative improvement).

The estimate that 10% of a subject's training is on endgame positions is difficult to verify. However, as noted earlier, it is known that players play a large number of speed chess games, where check situations are frequent, and the proportion was estimated in advance of the experiment, so we believe that the figure is reasonable pending other evidence.

A small number of errors were produced by the human subjects: a mean of 3.0%, with a maximum of 4.1% by the experts. CHREST is theoretically able to produce errors (for example, by over-generalising learnt information), but none were produced in these simulations. This may be considered a weakness of the model, but given the proportion of errors made by humans, and that a number of these may have been due to errors of attention (e.g. pressing the wrong key due to fatigue), we do not think this is a serious drawback.

Despite these considerations, we find the results good evidence for our hypothesis. We have proposed a model of how humans carry out simple check detection and found that, with a revision and accepting certain assumptions, it explains the human data well, both quantitatively and qualitatively.

Our revised model shows poorer performance in modelling the perception of weaker players. This may be natural variability, given Saariluoma's small sample size, but we also consider other possible reasons:

Our first model may have been partly right, and though stronger players do use the more efficient, simplified, perceptual strategy described in our revised model, weaker players use (a subset of) the unnecessarily complicated strategies used for perceiving a regular game position.

Alternatively (or in addition), there is some evidence (Reingold, Charness, Pomplun, & Stampe, 2001) that stronger players make better use of their peripheral vision to detect pieces, suggesting that we may have allowed weaker players too much ability in our revised model.

Also, weaker players may spend more time checking their decisions. We have assumed that a "double check" is carried out (checking the relationship between the position of both perceived pieces), but weaker players may find it necessary to make additional checks. It would be expected that stronger players would not feel the need to do this due to their improved confidence in their own ability.

Finally, there may be additional mental processes involved with weaker players which we have not considered. For example, absolute beginners may spend some time trying to remember how each piece moves.

Looking forward, our theory makes predictions that can be tested. Most obviously, players' eye movements could be recorded whilst carrying out this task to determine if our theory of how attention is directed (i.e. very simply and directly) is correct.

Our theory, that chunks are linked to further knowledge, including information about whether a chunk includes a check or not, also leads to some predictions.

First, there should be increased intra-subject variability across different positions compared to the "general exercise" hypothesis of several different mental processes being improved as expertise is acquired: general-purpose processes should not be affected by the specifics of an individual position.

Second, there should be a ceiling to performance on the task. The largest network we tested was 100,000 nodes, but there are only 16,128 separate possible positions containing only a king and one of {queen, rook, bishop, knight} of the opposite colour. Our imposed end game-specific practice of 10% (estimated, but seeming to produce a good match of the human data) of a 100,000 node network covers the majority of these positions. If our theory is correct, performance on this task should rapidly tail off above Grand Master level since there will be fewer additional novel chunks to acquire.

Conclusions and Future Work

We have proposed a hypothesis of how humans perceive and make decisions on checks in simple check positions and from this produced a model that successfully reproduces the experimental human data.

This theory may have wider implications in terms of chunking theory. We have suggested that chunks are linked to extended information — specifically, information about whether the king is in check or not. This theory (if correct) raises a number of questions about the extent and types of information that may be linked to visuo-spatial chunks. Chunks could, for example, be linked to additional semantic information about their strategic value, their relationship to other chunks, or a verbal description. Based on earlier work (Gobet & Jansen, 1994), we are currently attempting to expand this theory by investigating how move sequences in chess may be learnt and attached to visual chunks in a similar manner.

Another, more direct, way to build on this work is to consider checks involving multiple pieces, for example in mid game chess positions. More complicated perceptual strategies would undoubtedly be involved.

Finally, in order to successfully model the human data, we have had to modify the perceptual strategies used, following the assumption that this behaviour would be controlled by conscious processes. Whilst this is a reasonable assumption backed by evidence, it required human intervention; ideally, the model would be able to alter its own behaviour in this way, controlling what information entered STM and directing its own perceptual strategies.

Acknowledgment

This research was funded by the Economic and Social Research Council under grant number RES-000-23-1601.

References

Baddeley, A. (1990). *Human memory. Theory and practice.* Boston: Allyn & Bacon.

Brebner, J., & Welford, A. (Eds.). (1980). *Reaction times.* New York: Academic Press.

Campitelli, G., & Gobet, F. (2005). The mind's eye in blindfold chess. *European Journal of Cognitive Psychology*, *17*, 23–45.

Chase, W., & Simon, H. (1973). The mind's eye in chess. In W. Chase (Ed.), *Visual information processing* (pp. 215–281). New York: Academic Press.

Church, R., & Church, K. (1977). Plans, goals, and search strategies for the selection of a move in chess. In P. Frey (Ed.), *Chess skill in man and machine* (pp. 131–156). New York: Springer Verlag.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*, 87–114.

de Groot, A. D. (1978). *Thought and choice in chess* (2nd ed.). The Hague: Mouton Publishers.

de Groot, A. D., & Gobet, F. (1996). *Perception and memory in chess: Heuristics of the professional eye*. Assen: Van Gorcum.

Freudenthal, D., Pine, J. M., Aguado-Orea, J., & Gobet, F. (2007). Modelling the developmental patterning of finiteness marking in English, Dutch, German and Spanish using MOSAIC. *Cognitive Science*, *31*, 311–341.

Gobet, F., & Campitelli, G. (2007). The role of domain-specific practice, handedness and starting age in chess. *Developmental Psychology*, *43*, 159–172.

Gobet, F., & Jansen, P. (1994). Towards a chess program based on a model of human memory. In H. van den Herik, I. Herschberg, & J. W.Uiterwijk (Eds.), *Advances in computer chess* (Vol. 7). Maastricht: University of Limburg Press.

Gobet, F., Lane, P. C. R., Croker, S. J., Cheng, P. C.-H., Jones, G., Oliver, I., et al. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, *5*, 236–243.

Gobet, F., & Simon, H. (2000). Five seconds or sixty? presentation time in expert memory. *Cognitive Science*, *24*, 651–682.

Gobet, F., & Waters, A. (2003). The role of constraints in expert memory. *Journal of Experimental Psychology: Learning, Memory Cognition*, *29*, 1082–1094.

Jones, G. A., Gobet, F., & Pine, J. M. (2005). Modelling vocabulary acquisition: An explanation of the link between the phonological loop and long-term memory. *Journal of Artificial Intelligence and Simulation of Behaviour*, *1*, 509–522.

Lane, P. C. R., Cheng, P. C.-H., & Gobet, F. (2000). CHREST+: Investigating how humans learn to solve problems with diagrams. *AISB Quarterly*, *103*, 24–30.

Lane, P. C. R., Gobet, F., & Smith, R. Ll. (2008). Attention mechanisms in the CHREST cognitive architecture. In L. Paletta & J. Tsotsos (Eds.), *Proceedings of the 5th international workshop on attention in cognitive science* (pp. 207–220). Graz: Joanneum Research.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, *390*, 279–281.

Milojkovic, J. (1982). Chess imagery in novice and master. *Journal of Mental Imagery*, *6*, 125–144.

Reingold, E., Charness, N., Pomplun, M., & Stampe, D. (2001). Visual span in expert chess players: Evidence from eye movements. *Psychological Science*, *12*, 48–55.

Saariluoma, P. (1984). *Coding problem spaces in chess: A psychological study* (Vol. 23). Helsinki: The Finnish Society of Sciences and Letters.

Shiffrin, R., & Schneider, W. (1977). Controlled and automatic human information processing: 2. Perceptual learning, automatic attending and a general theory. *Psychological Review*, *84*, 127–190.

Simon, H. (1969). *The sciences of the artificial.* Cambridge, MA: MIT Press.

Simon, H., & Gilmartin, K. (1973). A simulation of memory for chess positions. *Cognitive Pyschology*, *5*, 29–46.

Smith, R. Ll., Gobet, F., & Lane, P. C. (2007). An investigation into the effect of ageing on expert memory with CHREST. In *Proceedings of the seventh UK workshop on computational intelligence.*

Welford, A. (1977). Motor performance. In *Handbook of the psychology of aging* (pp. 450–496). New York: Van Nostrand Reinhold.

Computational Models of Human Document Keyword Selection

Michael J. Stipicevic (stipim@rpi.edu)
CogWorks Laboratory, Rensselaer Polytechnic Institute
Troy NY, USA

Vladislav D. Veksler (vekslv@rpi.edu)
CogWorks Laboratory, Rensselaer Polytechnic Institute
Troy NY, USA

Wayne D. Gray (grayw@rpi.edu)
CogWorks Laboratory, Rensselaer Polytechnic Institute
Troy NY, USA

Abstract

Computational models are presented that attempt to mimic how humans select keywords to describe documents. These semantic models are based on data mining techniques applied to large corpora of human writing. A methodology to test the merit of these models is developed; performance at matching author-chosen keywords is the basis of this test. Results indicate topic models and their derivatives outperform traditional semantic models. Finally, it is shown how these models might be incorporated into a system that automatically selects keywords for an academic publication.

Keywords: machine learning; natural language processing; Latent Semantic Analysis; Latent Dirichlet Allocation; Correlated Topic Models; VGEM

Background

After reading a text, humans are able to provide a quick summary of its contents. The smallest summary possible is simply a list of topics. These topics, or 'keywords,' represent the highest levels of human abstraction: they dramatically reduce an entire document to a few words while retaining key information. A computational model of keyword generation would allow researchers to better understand how knowledge is extracted, abstracted, and generalized in the mind.

In this paper we compare the ability of four computational models to pick out author-selected keywords from a larger set of possible keywords. The models are contrasted based on theoretical and technical differences. Finally we propose future research to better understand the underlying mechanics of these models and show how they may be useful as normative tools for automatic keyword generation. The tests of the four models (fit to data) serve as a proof of concept of such a system.

Measures of Semantic Relatedness

Measures of semantic relatedness (MSRs) are techniques that quantify the semantic relationships between two words or documents. They derive a numeric ranking of relatedness from a fitted semantic model. For example, after being trained on a large corpus of English text, an MSR might determine that 'cat' and 'dog' are highly related. The calculation of this ranking depends on each MSR, but the interpretation is the same.

All of the selected MSRs (and the models they are based on) depend on the 'bag of words' assumption (Landauer, Laham, et al., 1997). This means that word order or context does not factor into the relatedness computation. As keywords merely describe the topics of the text (instead of the content), this assumption should not hinder the predictive power of the models.

Although all MSRs are capable of calculating relatedness between words, only a few of the measures are capable of determining relatedness between multi-word terms (e.g. documents, paragraphs). The four MSRs selected for this analysis are able to do this.

Our assumption is that keywords may be selected for a given document from a larger ontology based on document-keyword relatedness values. In other words, for a given MSR, *m*, a document, *D*, a set of appropriate keywords for this document, $k_{1..n}$, and a set of less appropriate keywords for this document (distractors), $d_{1..n}$, we assume that $m(D,k_x)>m(D,d_x)$.

Semantic Models

LSA

Latent Semantic Analysis was first proposed in the late 1980s as a way to extract meaningful relationships between text (Landauer & Dumais, 1997). It has become the basis of numerous applications including educational testing, search engines, and optical character recognition (Zhuang, Bao, Zhu, Wang, Naoi, 2004). LSA uses the singular value decomposition (SVD) to identify the strongest linear relationships within text corpora. The matrices resulting from this analysis can be used to calculate word-to-word, word-to-document, or document-to-document similarity.

Constructing the LSA model

A word-document matrix is constructed from a corpus of natural language. Each element in the matrix is the tf-idf ranking (term frequency–inverse document frequency; Salton and McGill, 1983) of the corresponding word in the corresponding document. Using tf-idf allows LSA to discount frequent words that have low semantic content ('the,' 'what'). The calculation for tf-idf is:

$$\frac{f(x \mid d)}{length(d)} \times \log \frac{m}{f(x)}$$

,

where $f(x|w)$ is the number of times that some word, x, appears in a given document, d, $length(d)$ is the number of words in d, m is the total number of documents in a corpus, and $f(x)$ is the total number of times that x appears in the corpus.

Once the word-document matrix is populated, the singular value decomposition is run. This produces a representation of the original word-document space but realigned to capture important relationships. Restricting the new semantic space to the N most important dimensions provides a set of vectors associated to each word in the corpus an each document in the corpus. For this paper, N was set to 50 to provide a large enough number of dimensions without impeding the usefulness of the SVD.

Calculating a relatedness value

Two methods are available to compare words to documents: appending the document's word count to the original word-document matrix as the start of training, or summing each word's topic vector over the entire document. Since the former requires an expensive SVD computation for every test, we choose the latter to evaluate semantic relatedness.

For every keyword/document pair, a semantic relatedness measure can be calculated as follows:

1. For every keyword/document, look up each word in the reduced LSA semantic space (see above). This produces a vector of length N for every word.
2. Sum these vectors over the entire keyword/document.
3. Take the summed keyword vector and the summed document vector and determine the cosine between the two vectors. This cosine-similarity is the final score provided by LSA (Landauer & Dumais, 1997).

Implementation

LSA was implemented for this paper in custom software. A word-by-document matrix was constructed and populated with corresponding tf-idf values. This matrix was passed to a Matlab SVD routine which computed the reduced semantic model. This model correlates words to the reduced semantic space. A vector for each document or keyword phrase was calculated by summing the individual word vectors, and the final relatedness value is the cosine between the document and keyword vectors.

LDA

In attempting to rework LSA with a strong probability model, Latent Dirichlet Allocation was developed (Blei, Ng, Jordan, 2003). This technique models each document as a probability distribution of topics; each topic is modeled as a distribution of words. By inferring what topic and word distributions exist in a corpus, LDA is able to provide an intuitive notion of topic – and keyword – extraction. LDA and similarly derived methods are called 'topic models' because, unlike methods such as LSA, topics are an explicit component in the model.

Constructing the LDA model

Latent Dirichlet Allocation builds a generative model of text by fitting a proposed model against known data. Specifically, LDA constructs a hierarchical Bayesian model based on Dirichlet priors. Thus, documents comprise a Dirichlet distribution of N “topics,” while topics comprise a multinomial distribution of words. Two corpus-wide parameters govern the model: the Dirichlet prior for topics are controlled by a scalar parameter α and the multinomial distribution for words in topics is controlled by the N-vector β. By estimating α and β for a corpus, a document's topical content may be computed and compared. As with LSA, N was chosen to be 50 for this paper.

1. Initialize a placeholder set of topic probabilities (γ) and a placeholder set of probabilities that each word was derived from each topic (Φ).
2. Using the expectation-maximization algorithm (Dempster, Laird, & Rubin, 1977) determine the best Dirichlet parameters to predict the word-document matrix (as calculated for LSA).
    1. For each word and using the current estimate of γ, α, and β, estimate the probabilities of which topic each word was derived from (Φ).
    2. Normalize Φ so it sums to 1.
    3. For each document and using the updated estimate of Φ, α, and β, calculate the new per-document topic probabilities γ.
3. Once Φ and γ are calculated, estimate α, and β. Repeat steps 2 and 3 until convergence.

Calculating the relatedness value

Using Bayesian inference (with the Dirichlet priors calculated above), a probability for each topic in a document can be calculated. This vector is a topical “fingerprint” of the document, and is similar to the vector created by LSA. Another topic vector is created for the keyword, and the cosine similarity between document and keyword vectors provides a similarity score.

Implementation

For this paper, the LDA-C software package was used. This takes a list of word counts for each document and outputs a fitted LDA model in terms of topic, document, and word probabilities. It also infers the topic probabilities of a new document (when given a fitted model). The similarity between a document and keyword phrase can be determined by first inferring the topic probabilities of each. These can be treated as a vector, and the cosine represents their relatedness.

CTM

Correlated Topic Models extend LDA by allowing topics to be correlated with each other (Blei & Lafferty, 2006). LDA requires topics to be statistically independent, but this may not be true in practice. For example, a document with a topic related to biology is more likely to contain chemistry related topics than topics concerning the French Revolution. CTM allows for this correlation by using a logistic distribution instead of the Dirichlet.

Constructing the CTM model

Correlated Topic Models are constructed in a similar manner to Latent Dirichlet Allocation models, the main difference is the choice of the logistic norm instead of the Dirichlet prior. The logistic norm allows topics to be correlated with each other – specified by a correlation matrix. Inference of these new parameters are performed similarly as in LDA.

Calculating the relatedness value

The results from CTM are computed exactly the same as with LDA – the distribution over topics is treated as a vector and the cosine similarity is computed between the keyword and document. The cross-topic correlation values are ignored.

Implementation

As with LDA, CTM was computed using a software package. CTM-C takes similar inputs and provides similar outputs as LDA-C. Since the cross-topic correlation values are ignored for this paper, the calculation of relatedness values in CTM is the same as in LDA.

VGEM

VGEM (Vector Generation from Explicitly-defined Multidimensional semantic space; Veksler, Govostes, & Gray, 2008) was recently proposed as an alternative to the more computationally-intensive MSRs introduced above. Like LSA, VGEM represents terms as vectors in a multidimensional semantic space, and calculates term relatedness as the cosine between their vectors. However, VGEM does not require construction or computational reduction of a document-by-word matrix (which becomes extremely expensive for sufficiently large corpora). Instead, VGEM requires a set of words to be explicitly chosen as the dimensions of the semantic space, and calculates term vectors dynamically based on term frequencies and term co-occurrences with each of the dimension-words. Various frequency/co-occurrence formulas may be used, e.g. Pointwise Mutual Information (Turney, 2001), or Normalized Google Distance, (Cilibrasi & Vitanyi, 2007).

For the purposes of this paper, VGEM dimensions were taken to be the topics derived from LDA, and frequency/co-occurrence formula used for calculating term vectors was Normalized Similarity Score (NSS), which is a variant of Normalized Google Distance. To be more precise, the value of each word, x, on dimension, y, is derived as follows:

$$\mathrm{NSS}(x, y) = 1 - \mathrm{NGD}(x, y)\ ,$$

where NGD is a formula derived by Cilibrasi & Vitanyi (2007):

$$\mathrm{NGD}(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - \min\{\log f(x), \log f(y)\}}$$

$f(x)$ is the frequency with which x may be found in the corpus, $f(x,y)$ is the frequency with which both x and y may be found in the corpus, and M is the total number of texts in the corpus.

Constructing the VGEM model

1. After training an LDA model(N = 50), select the two highest probability words from each topic in the corpus. This provides the dimension words for VGEM and only needs to be performed once per corpus. Note that VGEM may have up to 100 dimensions, but may have fewer due to redundant words in the LDA-derived dimensions.

Calculating relatedness value

1. For each word in the document, calculate the NSS between the word and each dimension word.
2. Over each document, sum up all word similarity vectors
3. To compare both documents, calculate the cosine similarity between the two summed vectors.

Implementation

VGEM was implemented in custom software for this paper. After the set of word-document counts were computed, LDA was trained on the corpus to provide VGEM dimension words. The two most probable word for each topic were chosen for VGEM dimensions (duplicate words were only represented once). NSS value between a given word and each dimension word generate a VGEM vector. Negative NSS values were clamped to zero. For each document or keyword phrase, VGEM vectors for each word were summed, providing a vector for the entire document or keyword phrase. The similarity between the two is simply the cosine, as for all the measures presented in this paper.

Methodology

Corpus Selection

Our testing and training corpus is the proceedings of the Annual Meeting of the Cognitive Science Society from 2004 to 2008. 100 papers from 2008 were removed to provide test cases, while the rest of the papers were used to train the models as described above. All stopwords as defined in the Python Natural Language Toolkit (Loper & Bird, 2002) were discarded, in addition to all words less than 3 characters long and all words occurring in less than 3 documents.

Segmentation

Each document was processed in two separate modes: by -document and by-paragraph. In paragraph mode, each document was split by software into 50-word non-overlapping segments (the final segment, even if less than 50 words, is kept). Sentence and true paragraph boundaries are ignored (to preserve the 'bag of words' assumption). Document mode leaves the document intact. Thus, in by-paragraph mode, a word-by-paragraph matrix is constructed for LSA, LDA, and CTM instead of the word-by-document matrix mentioned above (splitting up corpora into smaller segments is standard practice, e.g. Landauer & Dumais, 1997).

Segmenting documents by paragraph allows insight into what information these models incorporate. In paragraph mode, each model is only given a small window to process text. This emphasizes locality in word-word association. However, because the windows do not overlap, paragraph mode may break word-word associations that lie across 'paragraph' boundaries. This would lead to an artificially inflated number of topics as there would be less second-order word-word correlations. Future research will examine the use of a sliding window to solve this problem.

Testing Method

Test Cases

In order to evaluate the predictive power of these models, we determine how well each model can select author-chosen keywords. Each test case consisted of a *cue*, *targets*, and *distractors.* The *cue* was one of the 100 selected documents from Cognitive Science 2008 conference proceedings, as mentioned in the Corpus Selection section. The *targets* were the author-picked keywords from the *cue* document, and the *distractors* were random keywords from other documents. The number of *distractors* was twice the number of *targets,* a random guess would be correct 33% of the time. It should be noted that each 'keyword' may actually be a keyword phrase (e.g., 'natural language processing'). This does not pose a problem as each MSR is able to process multi-word terms.

For each MSR, the *targets+distractors* list was ranked and sorted in accordance with MSR's relatedness values between the *cue* and each of the keywords. Finally, the score for each MSR on each *cue-targets-distractors* test case was calculated as follows:

$$Score_{\text{case}} = \frac{\textit{Number of targets in top n words}}{n}$$

where *n* is the number of *targets*, and "*top n words*" refers to the top third of the sorted *targets-distractors* list. Thus, if all target keywords are more related to *cue* than any of the distractor keywords, the score for that test case is 100%. If none of the target words are picked by the MSR to be more related to the cue than any of the distractor words, the score is 0. The overall score for a given MSR is the average of all 100 test case scores.

Results

The mean performance, measured as the ability to select author generated keywords from among distractors, is shown in (Figure 1). We compared the best results from each measure (by-document mode for LSA and LDA; by-paragraph mode for CTM and VGEM) by means of a repeated measures ANOVA. The analysis revealed a significant main effect of Measure, F(3, 395)=13.011, p<.01. Posthoc Tukey HSD comparisons revealed significant differences between LSA (M=.53, SE=.02) and LDA (M=.69, SE=.02), LSA and CTM (M=.63, SE=.02), and LSA and VGEM (M=.68, .02). No significant differences were found between LDA, CTM, and VGEM.

Table 1. Mean and standard error values for the performances of four MSRs on selecting author-generated keywords among distractor keywords, using by-document and by-paragraph modes of training. Chance-level performance is .33.

Mode	Document		Paragraph	
	Mean	Std. Err	Mean	Std. Err
LSA	0.53	0.020	0.38	0.021
LDA	0.68	0.021	0.65	0.019
CTM	0.59	0.022	0.63	0.018
VGEM	0.42	0.021	0.68	0.019

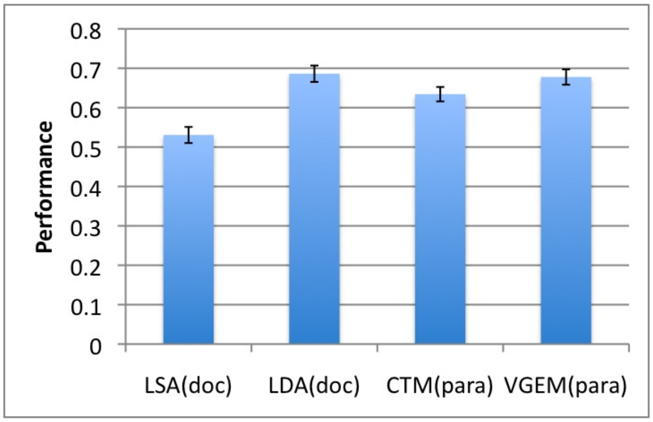


Figure 1. Best mean performances of four MSRs on selecting author-generated keywords among distractor keywords. Chance-level performance is .33. Error bars represent standard error.

Analysis

It is not surprising that topic models (LDA and CTM) consistently perform well in this task. Keywords are nothing more than descriptions of topics; while LSA and VGEM can be interpreted to use topics, they do not model them as explicitly as LDA and CTM.

In paragraph mode, LSA barely outperforms random guessing (0.33). This implies that LSA heavily depends on second and even higher order word correlations, which has been confirmed in several contexts (Kontostathis & Pottenger, 2002). However, the exact opposite has occurred with VGEM – non-overlapping segmentation of text has significantly increased its score. VGEM's underlying MSR – NSS – explicitly uses first order co-occurrence in its model, but nothing else.

LDA and CTM were not statistically different from each other. This is striking because CTM was developed as an improvement upon LDA. Perhaps the utility of CTM is not realized without very disparate topics, the narrow scope of cognitive science papers might render the topic correlation of CTM ineffective.

VGEM is statistically equivalent to LDA and CTM. This is interesting as it is a much simpler measure. Of course, in this paper VGEM used LDA-derived dimensions; this means a full LDA training step must be performed to obtain these results. However, that only needs to be performed once per corpus. With dimensions, VGEM only depends on two tabulations: how many documents contain a given word, and how many documents contain two given words. These can be performed quickly on large or even streaming databases. Additionally, the VGEM approach is adaptable to new vocabulary: as long as the new word appears in a document with a dimension word, a VGEM score can be computed for it. The other three models would require computationally expensive retraining.

Future Development

Sliding Window

As mentioned in analysis, the paragraph model emphasizes locality, but interferes with higher-order word correlation. Preprocessing the text as a sliding window will eliminate artificial barriers by overlapping the selected text. This has the unfortunate side-effect of greatly increasing computation time, which is why it was left out of this analysis. The size of the window (either sliding or non-overlapping) could be modulated to find parameters that best suit the data. This analysis might not benefit LDA (which performs worse on paragraphs) but might boost VGEM performance even higher.

General Corpus

In this analysis, the MSRs were trained on a corpus with a narrow technical focus – cognitive science articles. Although a substantial number of documents were used, the limited breadth of this corpus might be an issue, especially when specialized vocabulary is considered. Lindsey, Veksler, Grintsvayg, & Gray (2007) explore the performance of MSRs when used on different types of corpora, a similar analysis could be performed on the keyword-matching test.

Larger Parameter Search

In addition to the corpus used and the text windowing chosen, there are several variables that affect the outcome of the results. The number of topics/dimensions used, the selection of topics for VGEM, and the choice of MSR all might interact in complicated ways that can only be determined by a more rigorous examination of the parameter space.

Automated Keyword Generation

This work lays the foundation for an automated keyword selection system. Editors of scholarly publications solicit keywords for each submission, mainly to assist in assigning reviewers with relevant interest. However, these keywords can describe topics too narrow or too broad and are rarely consistent across authors (Furnas, Landauer, Gomez, & Dumais, 1987). This is not fully alleviated by a fixed set of keywords: authors may pick too few or too many keywords, and the keyword set may be redundant or omit crucial topics.

A system built upon the keyword-matching test could provide a solution to these two problems. For the first, any of the MSRs described can rank the relationship between each paper and an ontology of keywords. A relatedness threshold can determine which keywords to retain for the document and which to discard. As for the second problem, the ontology of keywords itself can be generated by topic models such as LDA and CTM, much like the VGEM dimensions extracted from LDA as described above. Theoretically, these would represent the 'true' topic distribution.

To test such a system, more human data would be required. Reviewers of each submission are most qualified to judge which keywords are most representative of the document. When reviewing a paper, they could be sent two sets of keywords – author developed and MSR generated – and select which set they feel better suits the paper. This could be expanded to provide keywords from multiple MSRs

Of course, with such sophisticated ranking systems, one questions the usefulness of keywords at all – could we not move straight to an automated matching of submissions to reviewers?

Summary

We describe several MSRs of various theoretical underpinnings and introduce a test that measures the ability of MSRs to match human document keyword selection. We then evaluate each MSR's performance on this test. The results demonstrate that topic models perform well and are robust under text segmentation. Additionally, VGEM (with LDA-derived dimensions) performs as well as topic models with the added benefit of adaptability and speed. Finally, we show how this test is a proof of concept of an automatic keyword generation system.

Acknowledgments

The work was supported, in part, by grant N000140710033 to Wayne Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer.

References

Blei, D., & Lafferty, J. (2006). Correlated topic models. In Advances in Neural Information Processing Systems 18.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (Mar. 2003), 993-1022.

Cilibrasi, R., & Vitanyi, P. (2007). The Google Similarity Distance. IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 3, p. 370-383.

Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society.* Series B (Methodological) 39 (1): 1–38.

Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The Vocabulary Problem in Human System Communication. Communications of the ACM, 30(11), 964-971.

Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E., (1997). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In M. G. Shafto & P. Langley (Eds.), Proceedings of the 19th annual meeting of the Cognitive Science Society (pp. 412-417). Mawhwah, NJ: Erlbaum.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review, 104(2), 211-240.

Lindsey, R., Veksler, V. D., Grintsvayg, A., & Gray, W.D. (2007). Effects of Corpus Selection on Semantic

Relatedness. 8th International Conference of Cognitive Modeling, ICCM2007, Ann Arbor, MI.

Kontostathis, A., & Pottenger, W. M. (2002). A mathematical view of latent semantic indexing: Tracing term co-occurrences. Technical Report, LU-CSE-02-006, Department of Computer Science and Engineering, Lehigh University.

Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. In Proceedings of the Workshop on Effective Tools and Methodologies for Teaching NLP and Computational Linguistics, pages 63–70, Philadelphia, PA, 7 July.

Salton, G. & M. McGill. (1983). Introduction to Modern Information Retrieval. McGraw-Hill.

Turney, P. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In L. De Raedt & P. Flach (Eds.). Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001) (pp.491-502). Freiburg, Germany.

Veksler, V. D., Govostes, R., & Gray, W. D. (2008). Defining the Dimensions of the Human Semantic Space. Accepted to the 30th Annual Meeting of the Cognitive Science Society.

Zhuang, L., Bao T., Zhu, X., Wang, C., Naoi, S. (2004). A Chinese OCR Spelling Check Approach Based on Statistical Language Models. International Conference on Systems, Man and Cybernetics, pp. 4727-4732. Hague, Netherlands: IEEE.

Memory in Clark's Nutcrackers: A Cognitive Model for Corvids

Elske van der Vaart (e.e.van.der.vaart@rug.nl)[1,2]
Rineke Verbrugge (l.c.verbrugge@rug.nl)[1]
Charlotte Hemelrijk (c.k.hemelrijk@rug.nl)[2]

[1]Artificial Intelligence, University of Groningen, P.O. Box 407
9700 AK Groningen, the Netherlands
[2]Theoretical Biology, University of Groningen, P.O. Box 14
9750 AA Haren, the Netherlands

Abstract

Computational modeling has rarely been used to study questions in animal cognition, despite its apparent benefits. In this paper, we aim to demonstrate the value of this approach by focusing on work with Clark's nutcrackers. Like all corvids, these birds cache and recover food, by burying it under ground and returning to it later. With our computational model, we successfully replicate three laboratory experiments investigating this behavior. In the process, we provide the first integrated computational account of several behavioral effects of memory observed in corvid caching and recovery, in addition to a new explanation for a known empirical result.

Keywords: Computational model; animal cognition; corvid; Clark's nutcracker; *Nucifraga columbiana*; caching; memory.

Introduction

Computational models are a favored instrument in the cognitive science toolbox (Sun, 2008). Yet, there is an area of cognitive science where they are rarely used: That of animal cognition research (Penn, Holyoak, & Povinelli, 2008). Although many computational models are built to study other animals (Grimm & Railsback, 2005), the focus tends to be on ecological questions, such as 'what causes dominance hierarchies to form?' or 'how do individuals decide when to migrate?'. In contrast, cognitive questions, that concern animal memory, learning, or problem solving, are seldomly subjected to this approach. This despite the fact that computational models of animal cognition issues can be very useful (Penn et al., 2008).

What we are interested in, is the cognition underlying the caching and recovery behavior of corvids. This family of birds, which includes crows, jays, and nutcrackers, hides food under ground, saving it for later. Recovery can occur after hours, days, or months have passed, and depends on memory for individual cache sites. This behavior has been extensively researched in the laboratory, with a strong focus on its cognitive aspects (de Kort, Tebbich, Dally, Emery, & Clayton, 2006). All these cache and recovery experiments use the same basic paradigm: The birds are offered a bowl of food, a discrete set of sites to cache in, and the presence or absence of a conspecific, and very little else. Nevertheless, the questions asked and the data gathered are diverse, and may concern topics ranging from basic memory mechanisms to higher-level skills, such as future planning and social cognition (de Kort et al., 2006). From a computational modeling perspective, this is excellent: It means that a single computational architecture of corvid cache and recovery cognition can be used to investigate a wide variety of cognitive phenomena.

In this paper, we present a step in that direction by focusing on three experiments with Clark's nutcrackers. These North American corvids are completely dependent on stored food in the winter months, and a single bird may bury up to 33,000 pine seeds a year, spread over thousands of different sites. Observational studies suggest that Clark's nutcrackers may recover their caches up to eleven months after making them, with a recovery accuracy of over 80%. One of the earliest laboratory experiments with these birds demonstrated the role of memory in this process: Like all corvids, a Clark's nutcracker cannot relocate caches by scent or by search, but only by remembering their location (see Kamil & Balda, 1985, for a review).

Since then, other laboratory experiments, in particular by Alan Kamil and Russell Balda, have investigated many more features of the Clark's nutcracker memory system, and it is three of these experiments that we replicate with our computational model (Balda, Kamil, & Grim, 1986; Kamil & Balda, 1990). All three have the same basic setup: The birds are tested in an experimental room, with 180 holes in the floor. These are spaced in a rectangular grid, 12 x 15 in size. Every hole can contain either a sand-filled cup, suitable for caching in, or a wooden plug, rendering it inaccessible. All subjects are always tested individually, and all sand-filled cups are smoothed over between sessions. Every experiment consists of a sequence of caching and recovery sessions. On caching sessions, the birds are offered a bowl of seeds to cache; on recovery sessions, the birds are hungry, and can only eat by recovering the seeds they have previously hidden in the experimental room.

From these three experiments, four patterns are apparent: A decrease in accuracy as recovery progresses, occasional return to already emptied sites, a lack of correlation between caching and recovery order, and a slight preference for re-caching in previously used cups. In this paper, we describe a computational model that successfully reproduces all four of these patterns. Its core component is *memory*, for cache and recovery events. To store these, we draw inspiration from the ACT-R (Anderson, 2007) cognitive architecture, and in particular, from its account of rational memory (Anderson & Schooler, 1991). What we use, is ACT-R's concept of *chunks*: A chunk is a small piece of information, with an *activation* that depends on its own history of use, as well as that of related chunks. Essentially, what our model does, is to encode a bird's *options* for caching and recovery as chunks, and to compute their total activation based on the bird's *memory* of where it has cached and recovered before. With noise in the activation values of chunks, this mechanism is enough to replicate the outcomes of all three Clark's nutcracker experiments under consideration. In this way, we provide the first integrated computational account of different behavioral effects of memory in corvid caching and recovery, and a new explanation for the experimental finding that the recovery accuracy of Clark's nutcrackers declines across sessions. What further strengthens the validity of our model, is that we have extended it to replicate a second set of patterns, concerning cache site choice in the scrub jay, another corvid species (van der Vaart, Hemelrijk, & Verbrugge, to appear). Thus, the idea of constructing a single computational architecture of corvid cache and recovery cognition appears to be a fruitful one.

Model

Our implementation of the Clark's nutcracker experiments consists of two main components: A *simulator* and a *cognitive model*. The simulator runs the experiments, while the cognitive model is a computational theory of the cognitive processes under concern. Motivational processes that govern whether the birds want to cache or recover at all, are not considered; we simply assume that the birds want to cache in caching sessions and recover in recovery sessions.

The Basics of Chunks

Our model features two types of chunks: *Option* chunks and *memory* chunks. Option chunks represent the locations that are *available* for the bird to cache or recover in; memory chunks represent the *actual* cache or recovery events that the bird has experienced. Every chunk has two features: An *identifier* and an *activation*. A chunk's identifier specifies which cup it represents within the experimental room, as determined by its x and y location. A chunk's activation A_i consists of three parts: *Base-level activation* B_i, *spreading activation* S_i, and *noise*; see Equation 1.

A chunk's base-level activation B_i is computed according to Equation 2, following ACT-R's equation for base-level learning (Anderson, 2007). Here, t_j represents the elapsed time t since use j of chunk i, while d is a decay parameter. The weighing factor w_i is determined by chunk i's type, and is considered in detail further on. The effect is that a chunk's base-level activation depends on its frequency and recency of use, and the kind of event it codes for. A chunk's spreading activation S_i depends on the activation of other chunks, and is discussed later. A chunk's noise value is re-computed every time it is evaluated, according to Equation 3, taken from ACT-R, where n is a parameter that we tune, and r is a random value between 0 and 1.

$$A_i = B_i + S_i + noise \tag{1}$$

$$B_i = w_i \cdot \sum_j t_j^{-d} \tag{2}$$

$$noise = n \cdot \ln(\frac{1-r}{r}) \tag{3}$$

$$S_{co_i} = -B_{cm_i} \tag{4}$$

$$S_{ro_i} = B_{cm_i} - B_{rm_i} \tag{5}$$

For the purpose of computing the activations of chunks, time is measured in steps. Every cache or recovery event counts as one step, and every non-experimental day counts as t steps, where t is a parameter that we tune. This simulates the flow of time *outside* of the experimental sessions.

The Structure of Caching and Recovery Sessions

At the beginning of every caching session, the simulator informs the cognitive model which cups are available to cache in. This is our equivalent of a Clark's nutcracker sitting on its perch, overseeing the room and registering its options. Then, every time the cognitive model starts to cache, it computes the activation of all its *cache option chunks*, according to Equation 1, and selects the most active one. This counts as a use of that chunk, and represents a bird's decision to cache in a particular cup. Once the cognitive model has selected its cache site, it caches there, and the corresponding *cache memory chunk* is given a use. Caching continues until the simulator asks the cognitive model to stop; this is determined by the number of caches made by the real birds in the original experiment.

A recovery session works in exactly the same way, except that it revolves around *recovery option chunks* and *recovery memory chunks*. The simulator ends a recovery session when the cognitive model has successfully retrieved as many caches as the real nutcrackers in the corresponding experiment are allowed to do.

The Memorability of Events

For the purpose of calculating a chunk's base-level activation B, according to Equation 2, cache memory chunks are given a weight w_{cm} of 5, while recovery memory chunks are given a weight w_{rm} of 2. This is inspired by the fact that Clark's nutcrackers probe a cup with their beaks about five times when making a cache, but only about twice when attempting to recover (Kamil, Balda, & Good, 1999). Option chunks, regardless of type, always carry a weight w_o of 1, representing the idea that *deciding* to cache or recover is less memorable than *actually* caching or recovering.

Inhibition of Return

To prevent the model birds from returning to recently visited sites, every memory chunk spreads *negative activation*, or *inhibition*, to the corresponding option chunk. See Equation 4 for cache chunks, and Equation 5 for recovery chunks. To work out the case of Equation 4:

The higher the base-level activation *B* of the cache memory chunk cm_i, the lower the spreading activation *S* of the cache option chunk co_i, and the smaller the odds that the model bird will return to that cache site.

Knowing Where to Recover

What allows the cognitive model to relocate its caches, is the fact that every cache memory chunk spreads *positive activation* to the recovery option chunk that codes for the same location; this is included in Equation 5. This has the effect that the cognitive model is *more* likely to try and recover in cups where it has actually cached items.

Experiments

To validate our cognitive model, we test it against three experiments with Clark's nutcrackers: Experiment 1 from Kamil *et al*. (1986), and Experiments 1 and 2 from Balda & Kamil (1990). Here, we describe both the nutcracker experiments and our model's replications of them. Model results are the average of 1000 runs, using the parameters of Table 1; see the Model section for an explanation of each.

Table 1: Parameter values used in the experiments.

d	*n*	*f*	w_{cm}	w_{rm}	w_o
0.1	2	10	5	2	1

Experiment 1 (Experiment 1 in Kamil *et al.* (1986))
In this experiment, the authors measure two aspects of the recovery behavior of Clark's nutcrackers: Their decreasing accuracy as recovery continues, and their tendency to revisit already emptied sites. To this end, four birds are allowed to cache in the experimental room, until they store seeds in about twenty cups. Approximately ten days later, three recovery sessions are held, on alternate days. In each of these sessions, every bird may recover about a third of its caches.

Empirical Results, Kamil *et al.* (1986) To calculate results, recovery accuracy is defined as the total number of caches recovered divided by the total number of cups visited. As Figure 1A shows, the birds' average accuracy declines significantly across recovery sessions, starting at about 55% in session 1 and ending at about 15% in session 3.

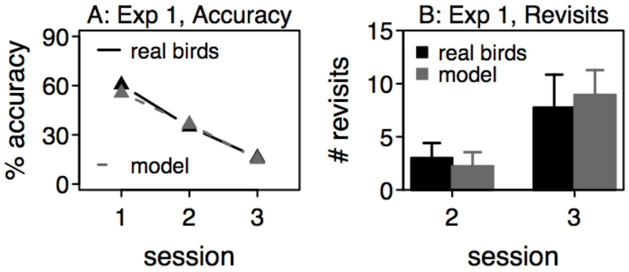


Figure 1: Results of Experiment 1, real birds, Kamil *et al.* (1986) and computational model; 1A: Average recovery accuracy, 1B: Average revisits to previously emptied sites, with standard errors.

Discussion In this experiment, the focus is on the repeat visits to previously emptied sites. Kamil *et al.* (1986) present two possible explanations for this: Either the birds remember their cache sites but not their recovery attempts, *or* they remember both, but continue to make revisits for some reason. For our model birds, the answer lies somewhere in the middle. When deciding where to recover, they follow the recovery option chunk that is currently most active. When determining the activation of a recovery option chunk, a corresponding cache memory chunk *raises* its activation, while a corresponding recovery memory chunk *lowers* it; see Equation 5. In this calculation, the uses of cache memory chunks are weighted five times, while the uses of recovery memory chunks are only weighted twice (Equation 2, Table 1). As stated previously, this is based on empirical observations of caching and recovery events (Kamil et al., 1999). As a consequence of this, recovery option chunks representing already visited cache sites tend to be less active than recovery option chunks representing not yet visited cache sites, but they also tend to be *more* active than recovery option chunks representing sites where the model bird never cached at all. This is what causes the model birds to make revisits.

Experiment 2 (Experiment 1 in Kamil & Balda (1990))
Here, Kamil & Balda (1990) investigate *why* Clark's nutcrackers become less accurate across recovery sessions. Given that these birds successfully locate their caches up to eleven months after making them, it seems unlikely that the two-day delay between recovery sessions is causing their accuracy to decline. Instead, the authors argue, what may be happening is that the birds remember some cache sites better than others, for whatever reason. Then, if they retrieve these 'best remembered' cache sites first, this explains why recovery accuracy drops. To test this idea, ten Clark's nutcrackers are exposed to two experimental conditions: The *quarters* and the *free* condition.

In the quarters condition, the birds are forced to recover their caches by room quarter, while in the free condition, they can recover at will. Each condition consists of one caching session, followed by four recovery sessions. In both conditions, during the caching session, only 32 cups are available for caching, eight in every quarter of the room. The birds may store seeds until they have created at least three caches in every quarter. A week later, recovery sessions begin, conducted on successive days. This is where the conditions vary: In the quarters condition, only one quarter of the room is available for recovery during each session, while in the free condition, all cups are always open. In the quarters condition, the birds may continue to recover until they have retrieved all caches created in the available quarter; in the free condition, they are chased out of the experimental room after they have recovered 25% of their caches. Now the reasoning is that if the birds remember some cache sites better than others, their recovery accuracy should stay the same across recovery sessions in the quarters condition, but decline in the free condition.

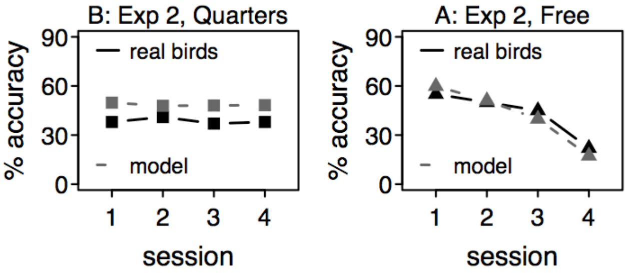


Figure 2: Results of Experiment 2, average recovery accuracy, real birds, Kamil *et al*. (1990) and computational model; 2A: Control condition, 2B: Quarters condition.

Empirical Results, Balda & Kamil (1990) As can be seen in Figure 2, the results are as expected: Recovery accuracy stays the same across recovery sessions in the quarters condition, but declines significantly in the free condition, and more quickly than would be expected by chance. In this experiment revisits to sites already emptied in previous recovery sessions are not counted as errors, because they can only occur in the free condition; instead, these revisits are ignored when calculating accuracy. In further analysis, Balda & Kamil (1990) look for a general relationship between caching and recovery order by calculating Spearman's rank order correlations for the 10 birds. Three of these are significant, but two are positive and one is negative, suggesting that no general relationship exists.

Model Results As can be see in Figure 2, the behavior of our model birds is similar to that of the Clark's nutcrackers: Accuracy does not decrease in the quarters condition, but does decrease in the free condition. Like the real birds, our model birds also show no systematic relationship between caching and recovery order in the free condition; of the 1000 correlations, only 3 are significant.

Discussion From these results, Kamil & Balda (1990) conclude that, in fact, Clark's nutcrackers remember some cache sites better than others. For our model birds, however, the explanation is different. In principle, they remember all cache sites equally, with the exception that the activation of cache memory chunks created earlier will have decayed more than the activation of cache memory chunks created later. However, if this were the explanation for the model birds' decline in recovery accuracy, we would expect to see no difference between conditions in this experiment. So what explains the model birds' constant performance in the quarters condition, but not the free condition?

The answer lies in the fact that, in the quarters condition, the number of caches that can be recovered remains the same across sessions, while in the free condition, it declines. When the cognitive model is deciding where to recover, it calculates the activations of all its recovery option chunks. On average, recovery option chunks representing cache sites are more active than recovery option chunks not representing cache sites, due to the spreading activation coming from cache memory chunks. This is what allows the cognitive model to make accurate recovery attempts, *most* of the time. However, noise may cause an 'incorrect' recovery option chunk to temporarily be more active than all 'correct' recovery option chunks. The lower the ratio of 'correct' to 'incorrect' recovery option chunks, the higher the odds of this occurring. In the quarters condition, the ratio of 'correct' to 'incorrect' chunks remains the same across sessions, because a fresh quarter of cups is available every time. In the free condition, by contrast, the ratio of 'correct' to 'incorrect' chunks decreases across sessions, because the birds continue to recover from the same set of cups. This is what explains our model's performance.

Experiment 3 (Experiment 2 in Balda & Kamil (1990))
In this experiment, Balda & Kamil (1990) further explore the idea of differential memory for different cache sites. They hypothesize that perhaps certain cache sites have physical attributes that make them particularly memorable, such as their placement near certain kinds of landmarks. If this is true, the authors argue, it predicts that if the birds are forced to repeatedly cache in the same sites, they should always cache and recover from them in the same order. After all, if specific sites have physical attributes that make them particularly attractive, they should always be preferred.

To test this theory, seven nutcrackers are exposed to an experiment with three stages. Each stage consists of a caching and a recovery session, with a week between the two, and a week between stages. In stage 1, the birds may freely make 15 to 18 caches. In stage 2, for every subject, only the cups used as cache sites in stage 1 of the experiment are available for caching. In stage 3, this set of cups is again available, together with a second set of cups, that is randomly selected and of equal size. In both stages 2 and 3, the birds are allowed to cache in about nine cups. In all three recovery sessions, the birds can freely recover. Now, the main question is whether or not the birds will demonstrate site preferences by always caching and recover in the same order, thus indicating site preferences.

Empirical Results, Balda & Kamil (1990) To analyze whether or not the birds prefer specific cache sites, Balda & Kamil (1990) calculate four Spearman's rank order correlations: Between caching and recovery in stage 1, between caching and recovery in stage 2, between caching in stage 1 and in stage 2, and between recovery in stage 1 and in stage 2. If a physical attribute is making some sites more memorable or more preferable, the birds should consistently choose to cache and recover in those sites first, producing significant correlations. Instead, the authors find no general relationships between caching and recovery orders; for all measures, they find a few significant correlations for some birds, but they go in both directions.

Another measure of interest is recovery accuracy *within* sessions. If some sites are more memorable than others, recovery accuracy within sessions should decrease, as the better remembered sites are recovered first.

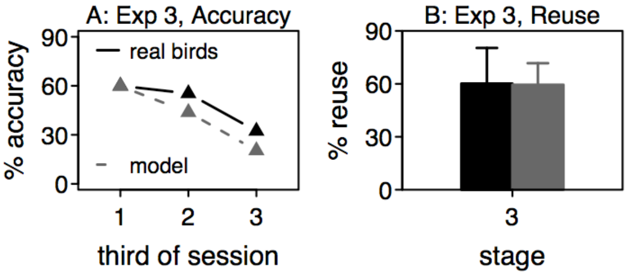


Figure 3: Results of Experiment 3, real birds, Kamil *et al.* (1990) and computational model; 3A: Average recovery accuracy, 3B: Average reuse of cache sites in stage 3, with standard errors, birds on the right, model on the left.

For each stage of the experiment, Kamil & Balda (1990) calculate the mean accuracy of the first three caches recovered, the middle three caches recovered, and the final three caches recovered. They find no significant differences *between* the three stages, but they do find a significant decrease in accuracy *within* the three stages; therefore, Figure 3A plots pooled accuracy results.

A final measure of cache site preference is the proportion of cache sites re-used in stage 3 of the experiment. If the birds choose to cache in particularly preferable sites in stage 1, we might expect them to still strongly prefer those sites in stage 3. Yet, this is not the case: As can be seen in Figure 3B, when offered their chosen set of sites from stage 1, and an equally sized set of new cups to cache in, they choose to cache in old cups only about sixty percent of the time.

Model Results Like the real Clark's nutcrackers, our model birds produce no significant correlations between cache and recovery orders on any of the measures tested by Kamil & Balda (1990). Furthermore, we also find a decrease in accuracy within recovery sessions, in all three stages of the experiment. Pooled accuracy results are plotted in Figure 3A. Finally, as can be seen in Figure 3B, in stage 3, our model birds re-use cache sites at approximately the same levels as the real nutcrackers: They choose to place about 60% of their caches in old sites, and 40% in new sites.

Discussion From this experiment, Kamil & Balda (1990) conclude that Clark's nutcrackers clearly do not have strong site preferences as dependent on *physical* attributes, or they would have consistently preferred to cache and recover from the same sites first. Our model birds show qualitatively the same patterns. The slight preference for old cache sites in stage 3 of the experiment can be explained by the fact that the cache option chunks corresponding to the cups that had already been chosen in stage 1 of the experiment already had two uses by this point, while the 'new' cups had none. As cache memory chunks only spread negative activation to cache option chunks *within* sessions, this means that the average activation of already-used cache option chunks is slightly higher than that of not-yet-used cache option chunks, explaining the model birds' behavior.

General Discussion

Our computational model raises three main questions: First, what does it tell us about Clark's nutcrackers? Second, how robust are its results? And third, how plausible is its design?

Implications of the Model for Clark's Nutcrackers

One of the attractive aspects of our model is that it uses one main mechanism, but fits four different patterns. We assume that both a birds' options and its choices are stored as chunks in memory, and that spreading activation between different chunk types takes care of the rest. This produces all four patterns apparent in the empirical data: A decline in accuracy both within and between recovery sessions, occasional return to already emptied sites, a lack of correlation between caching and recovery order, and a slight preference for re-caching in previously used cups.

In addition, we provide a new explanation for an observed result: The decline in accuracy as recovery proceeds. In a number of different papers, Kamil & Balda (1986; 1990) conclude that this is the result of differential memory for different cache sites, but in our cognitive model, the same effect arises as the result of chance. This seems to be a useful alternative theory, because the attempt to discover what might make certain sites more memorable than others has so far not been successful: As demonstrated by Kamil & Balda (1990) in the original version of our Experiment 3, Clark's nutcrackers do not consistently prefer some sites, suggesting that physical characteristics of particular locations cannot be responsible for different memorability. The birds' familiarity with particular cache sites is also an unlikely explanation, as Kamil, Balda & Good (1999) fail to find any predictors of recovery accuracy in the amount of time the birds spend making each cache.

However, several aspects of the model still need further work. One feature that seems particularly over-simplified is that all the cognitive model's errors are "true errors" – failures to retrieve a correct cache site location. However, for the real Clark's nutcrackers, it is probable that many errors are in facts acts of exploration. For instance, when the costs of making a recovery attempt are increased, the number of errors made drops significantly (Bednekoff & Balda, 1997). This is clearly an aspect of Clark's nutcracker behavior that we should explicitly consider in future.

Robustness of the Model's Results

When a computational model features free parameters, it is important to understand how strongly it *predicts* certain outcomes, and if there are any plausible alternatives that it *cannot* reproduce (Roberts & Pashler, 2000). For our cognitive model, this is certainly the case: As soon as *noise* is set higher than 0, in this type of experiment, it cannot fit anything but a decline in accuracy as recovery progresses (Experiment 2); constant performance is impossible, even if decay is set to 0. That constant performance is a plausible alternative, is demonstrated by Kamil & Balda's (1985) original theory that this was true of Clark's nutcrackers.

Plausibility of the Model's Implementation

Many of our model's core aspects are derived from the declarative memory module of the ACT-R (Anderson, 2007) cognitive architecture, lending it some initial validity. Of course, ACT-R was explicitly designed to model humans, so one might wonder whether our use of it for birds is appropriate. However, we do not think that is a problem in this case; as ACT-R's originator John Anderson himself notes (2007; page 18) many of the adaptive analyses on which ACT-R is based are not specifies-specific. There are many indications that, at a functional level, corvid memory may not be so different from ours; several experiments with scrub jays show that they are capable of flexibly integrating their memories, and have episodic-like "what, where, when" recall of past events (see de Kort *et al*., 2006, for a review.)

Our adaptations of the architecture itself might be more problematic. While ACT-R allows for spreading activation, it is a fixed amount, and it spreads only from 'goal chunks' to 'target chunks', depending on the strength of the association between them. In our cognitive model, it is a chunk's *own* activation that spreads, and this activation can even be negative, inhibiting a chunk's retrieval. The main function of this mechanism is to prevent the model birds from repeatedly caching or recovering in the same location. One might wonder if such a mechanism is necessary at all; if, instead, the real birds might be using a behavioral strategy to avoid revisits, such as 'recover a cache, look away, attempt to recover a cache in the field of view now visible'. This, however, does not appear to be a likely explanation; after successfully retrieving seeds, Clark's nutcrackers fly back to a central perch to eat them (Kamil & Balda, 1985). This means that, when they are deciding where to recover next, a very large portion of the experimental room is visible to them. This makes it very difficult to think of a behavioral strategy that avoids revisits to the extent that the real Clark's nutcrackers do.

Of course, this does not imply that our technique of spreading negative activation is necessarily the best way of implementing an inhibition of return mechanism. It is possible that the same effect could be achieved by instead *increasing* the activations of all other chunks, but we believe our solution is computationally easier, and intuitively plausible. Interestingly, other recent ACT-R adaptations also make similar changes to the architecture: Van Maanen & van Rijn (2007) let activation spread between chunks of different types, and Juvina & Taatgen (2009) attach negative activations to chunks. Although the context and justification is different, this negative activation mechanism serves the same function as ours – suppression of repetition – and operates in a similar fashion, with inhibitory activation that decays over time.

Conclusions

In this paper, we have shown that our computational model of corvid cache and recovery cognition can successfully reproduce the outcomes of three experiments with Clark's nutcrackers, fitting four different patterns. In addition, our computational model has provided a new explanation for the fact that Clark's nutcrackers become less accurate as recovery progresses.

Acknowledgments

We thank Alan Kamil and Russell Balda for further comments on their results, and NWO for funding Elske van der Vaart's doctorate, TopTalent grant 021.001.089.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford: Oxford University Press.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*, 396-408.

Balda, R. P., Kamil, A. C., & Grim, K. (1986). Revisits to emptied cache sites by Clark's nutcrackers (*Nucifraga columbiana*). *Animal Behaviour, 34*, 1289-1298.

Bednekoff, P. A., & Balda, R. P. (1997). Clark's nutcracker spatial memory: Many errors might not be due to forgetting. *Animal Behaviour, 54*, 691-698.

de Kort, S. R., Tebbich, S., Dally, J. M., Emery, N. J., & Clayton, N. S. (2006). The comparative cognition of caching. In E. A. Wasserman & T. R. Zentall (Eds.), *Comparative Cognition: Experimental explanations of animal intelligence*. Oxford: Oxford University Press.

Grimm, V., & Railsback, S. F. (2005). *Individual-based modeling and ecology*. Princeton, NJ: Princeton University Press.

Juvina, I., & Taatgen, N. A. (2009). A repetition-suppression account of between-trial effects in a modified stroop paradigm. *Acta Psychologica, 131*, 72-84.

Kamil, A. C., & Balda, R. P. (1985). Cache recovery and spatial memory in Clark's nutcrackers (*Nucifraga columbiana)*. *Journal of Experimental Psychology: Animal Behavior Processes, 11*(1), 95-111.

Kamil, A. C., & Balda, R. P. (1990). Differential memory for different cache sites by Clark's nutcrackers (*Nucifraga columbiana*). *Journal of Experimental Psychology: Animal Behavior Processes, 16*(2), 162-168.

Kamil, A. C., Balda, R. P., & Good, S. (1999). Patterns of movement and orientation during caching and recovery by Clark's nutcrackers. *Animal Behaviour, 57*, 1327-1335.

Penn, D. C., Holyoak, K. J., & Povinelli, D. J. (2008). Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. *Behavioral and Brain Sciences, 31*, 109-178.

Sun, R. (Ed.). (2008). *The Cambridge handbook of computational psychology*. New York: Cambridge University Press.

van der Vaart, E., Hemelrijk, C. K., & Verbrugge, R. (to appear). Learning Where (Not) To Cache: A Cognitive Model for Corvids. In N.A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31st Annual Meeting of the Cognitive Science Society*. Chicago: Lawrence Erlbaum Associates.

van Maanen, L., & van Rijn, H. (2007). An accumulator model of semantic interference. *Cognitive Systems Research, 8*(3), 174-181.

Second Life as a Simulation Environment: Rich, high-fidelity world, minus the hassles.

Vladislav D. Veksler
(vekslv@rpi.edu)
Cognitive Science Department
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180 USA

Abstract

Second Life™ is a 3D virtual world with unlimited potential as a tool for cognitive modeling. This paper discusses the many advantages of using Second Life versus other simulation environments, the aspects of cognitive modeling that this simulation environment may be appropriate for, the interface setup, and various technical issues. Two simulations are provided as examples of interfacing Second Life with cognitive models, including an example where the high-fidelity complexity and constraints of Second Life may help to distinguish between models and/or parameter values that produce varying performance in different task environments.

Keywords: cognitive modeling, Second Life, 3D virtual worlds, embodiment, ACT-R, task environment, cognitive architectures.

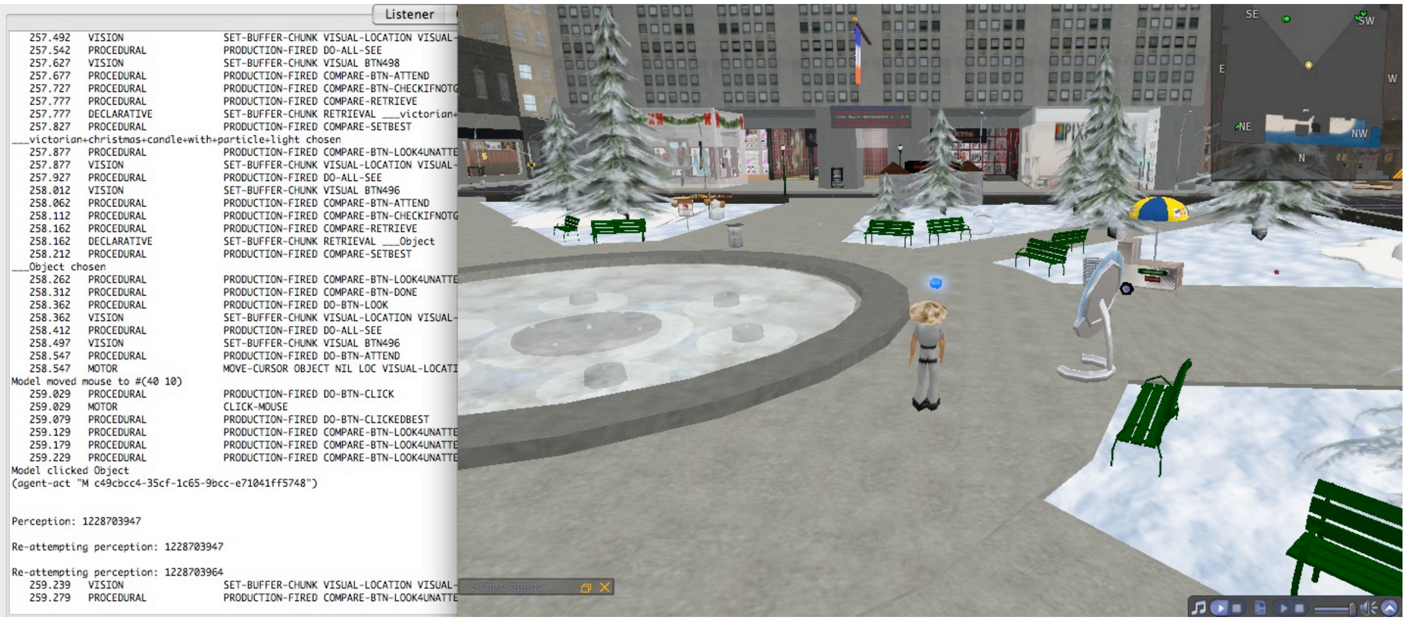
Introduction

The 3D virtual world Second Life™ offers a potential environment for training and testing cognitive models. Second Life is populated by hundreds of thousands of online users, and perhaps millions of virtual objects. This technology may be of interest to the cognitive modeling community for a variety of reasons, including scalability and scope testing, skill transfer simulations and long-term model development, emerging behavioral and social simulations, etc. Second Life provides a very rich, dynamic, and interesting world, (compared with the simple simulation environments that are typical in cognitive modeling), that is well-supported and easy to use and to redesign as needed (compared with robotics), with unlimited tasks, and the opportunity for life-long (rather than simulation-long) learning for a cognitive agent. Some Cognitive Science researchers have already begun to explore Second Life for demo and simulation purposes (e.g. Burden; Merrick & Maher, 2009; Rensselaer Polytechnic Institute, 2008), but more work with this environment is needed to take full advantage of its features.

The rest of this paper discusses the advantages of Second Life over alternative simulation environments for cognitive modeling, the types of simulations that Second Life may be appropriate for, and some key technical issues for modeling in this environment. Finally, two simulations are provided as examples of how cognitive models may be interfaced with Second Life, and how the high-fidelity complexity and constraints provided by the virtual world may be useful in distinguishing between models and/or parameter values that produce varying performance in different task environments.

Why Second Life?

The attraction of Second Life is the same as that of robotics



```
   257.492   VISION              SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   257.542   PROCEDURAL          PRODUCTION-FIRED DO-ALL-SEE
   257.627   VISION              SET-BUFFER-CHUNK VISUAL BTN498
   257.677   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-ATTEND
   257.727   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-CHECKIFNOTG
   257.777   PROCEDURAL          PRODUCTION-FIRED COMPARE-RETRIEVE
   257.777   DECLARATIVE         SET-BUFFER-CHUNK RETRIEVAL ___victorian+
   257.827   PROCEDURAL          PRODUCTION-FIRED COMPARE-SETBEST
___victorian+christmas+candle+with+particle+light chosen
   257.877   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   257.877   VISION              SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   257.927   PROCEDURAL          PRODUCTION-FIRED DO-ALL-SEE
   258.012   VISION              SET-BUFFER-CHUNK VISUAL BTN496
   258.062   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-ATTEND
   258.112   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-CHECKIFNOTG
   258.162   PROCEDURAL          PRODUCTION-FIRED COMPARE-RETRIEVE
   258.162   DECLARATIVE         SET-BUFFER-CHUNK RETRIEVAL ___Object
   258.212   PROCEDURAL          PRODUCTION-FIRED COMPARE-SETBEST
___Object chosen
   258.262   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   258.312   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-DONE
   258.362   PROCEDURAL          PRODUCTION-FIRED DO-BTN-LOOK
   258.362   VISION              SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   258.412   PROCEDURAL          PRODUCTION-FIRED DO-ALL-SEE
   258.497   VISION              SET-BUFFER-CHUNK VISUAL BTN496
   258.547   PROCEDURAL          PRODUCTION-FIRED DO-BTN-ATTEND
   258.547   MOTOR               MOVE-CURSOR OBJECT NIL LOC VISUAL-LOCATI
Model moved mouse to #(40 10)
   259.029   PROCEDURAL          PRODUCTION-FIRED DO-BTN-CLICK
   259.029   MOTOR               CLICK-MOUSE
   259.079   PROCEDURAL          PRODUCTION-FIRED DO-BTN-CLICKEDBEST
   259.129   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   259.179   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   259.229   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
Model clicked Object
(agent-act "M c49cbcc4-35cf-1c65-9bcc-e71041ff5748")


Perception: 1228703947

Re-attempting perception: 1228703947

Re-attempting perception: 1228703964
   259.239   VISION              SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   259.279   PROCEDURAL          PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
```

Figure 1. Cognitive Agent exploring a park in New York. ACT-R model controlling the agent.

– embodiment (minus the many hassles of robotics, discussed below). A large portion of human cognitive abilities is the result of the complexities and consistencies of our environment. Thus, a dynamic, rich world, with physical laws and consistent object properties may provide for more fidelity than simpler simulation environments, and thus, more useful models of cognition.

Second Life's complexity and constraints may help to avoid some 'false positives', as well as 'misses' in cognitive modeling. A false positive may occur when a cognitive model accounts for human data in a simplified task environment, but cannot scale in the real world. A miss may occur when a cognitive model cannot fit human data without the added complexity and constraints of the real world; thus, the use of a simplistic simulation environment may cause for the model to be incorrectly dismissed.

Second Life vs Robotics

If real-world fidelity is so important, why not just use the real world? There are many limitations to working with robots in the physical environment, versus simulated agents in virtual reality. In addition to the financial expenses, one major problem is that robotics work involves disproportionally more work on the 'body' as opposed to the 'mind'. In the end, a slight change to the task environment (e.g. taking a driving robot off-road) may require changes in both sensory and motor mechanisms.

While biological agents are endowed with appropriate sensory-motor systems for their world, and virtual agents for theirs, robotic agents are in no way equipped to handle the dynamics of the real world. For example, the number of sensors on today's robots, compared to the amount of sensors that a biological cognitive agent might have, is simply laughable. Virtual worlds like Second Life provide for environmental complexity and fidelity, as well as proportionally suitable sensory-motor abilities of virtual agents. Said simply, by using Second Life as opposed to a robot platform, researchers may be able to focus on cognitive research, and avoid unnecessary investments of time and finances.

Other Simulation Environments

Many other virtual simulation environments exist, and may be used for cognitive modeling. Some of the alternatives have better graphics, which can be very useful for demo purposes, some have a faster interface for brain-body communication, etc. However, due to the sheer size of the Second Life user community, due to its steadily increasing popularity, it makes for a much richer, ever-growing world. Additionally, the commercial value of Second Life is reflected in greater expansion of its technical capability and technical support. Using Second Life over a less popular simulation environment may be equated to using the World Wide Web over a Bulletin Board System.

What in the world of Cognitive Modeling is Second Life good (and not good) for?

Second Life may NOT be employed for modeling millisecond response times, nor is it appropriate for large-scale parameter exploration. Rather, Second Life is best used for modeling performance, and learning curves. Specifically, Second Life is best employed for (1) testing the scope of models' learning/decision-making mechanisms in complex and dynamic, distractor-full environment, (2) modeling adaptation and skill transfer, and (3) social modeling.

Complexity and Constraints

Spatial navigation is a prime example of a task that requires the complexity and constraints of Second Life for cognitive modeling. When modeling navigation, researchers often unrealistically represent the environment as a flat grid of adjacent spaces (e.g. Braga & Araujo, 2003; Voicu & Schmajuk, 2002). Some alternatives may be to include two-way or one-way wormholes. Different models may thrive in different environments, and so the choice of task-environment is not trivial. Second Life may be employed to provide realistic uncertainties and constraints. Although Second Life bears many geographic properties (e.g. if space A can be reached from space B, usually this means that space B can be reached from space A), it also provides many realistic uncertainties (e.g. object B may be in view when approaching object A from the East, but not from the North; object C may be dynamic, sometimes to be found in proximity with A, and sometimes in proximity with B, etc.). The use of a high-fidelity environment may help to deduce high-fidelity cognitive models and parameter sets.

Task Variety and Skill Transfer

Second Life may be used for simulations of a variety of tasks, from playing with building blocks, to maze-running, to soccer, etc. Most tasks are very easy to set up, require no programming or 3D modeling background, and are reusable by other researchers. Of great importance is the fact that an agent may 'live' and develop in this rich world, learning new skills along the way. The multitude of tasks can also help in modeling skill-transfer – an important qualification of human intelligence. A cognitive agent may adopt their soccer skills to hockey, walking skills to driving, and block-building skills to tower of Hanoi, Tetris, and sculpting.

Technical Setup and Complications

The Second Life programming language, LSL, is required for interfacing Second Life objects with cognitive models. Although the basic algorithm is relatively simple (capture and send sensory information to model; perform any actions returned by the model), some complications are bound to arise.

Land Ownership

There are many parts of the Second Life world where new objects cannot be created because the landowner does not allow this. There are parts of the world, called sandboxes, where users are encouraged to build and script their objects; however, objects usually cannot remain in most sandboxes for longer than a few hours. Thus, sandboxes may be fine for building models and running short simulations, but not for longer lifespans or more controlled simulations. One alternative is to buy land. Another may be to connect an object to an avatar (see a lengthier discussion of this in the Region Restrictions section below).

One last alternative is to use land that may be offered for research purposes by a university or a private research institution. For example, the Second Life AI Laboratory (SLAIL) provides booth size spaces for free to anyone undertaking research in AI (cognitive modeling included), and particularly AI in virtual worlds, providing a permanent exhibition, meeting and collaboration space for the community. The space may be found on Daden Cays in Second Life – (http://slurl.com/secondlife/Daden%20Cays/152/44/22; for more details visit http://knoodl.com/ui/groups/ArtificialIntelligenceGroup/wiki/SLAIL).

Region Restrictions

Scripted objects in Second Life are restricted from entering certain regions. If a modeling simulation requires travel beyond known open regions, it may be necessary to use an avatar (an avatar is a representation of a human user in Second Life, and only exists as long as the user is logged on). One simple way to resolve this issue is to attach the object interfaced with a cognitive model to an avatar. For example, the neon-blue sphere floating above the avatar's head in Figure 1 is an object scripted to interact with a cognitive model. For demo purposes the scripted object can be made see-through, tiny, or made to look like an article of clothing (e.g. a hat).

Firewall Issues

When a computer running a cognitive model is using DHCP, or if it is behind a firewall, a dedicated web server is necessary for interfacing the model with the Second Life world (Figure 2). Alternatively, LSL scripts can answer HTTP requests from the cognitive agent directly through their XML-RPC service (XML-RPC is a standard for XML structure for sending function calls to remote systems). This latter route is sometimes unreliable and may be deprecated ("Category:LSL XML-RPC - Second Life Wiki," 2009), but may be faster than the setup shown in Figure 2, depending on the speed of the researcher-owned web servers.

Asynchronous HTTP Calls

A question may arise when a cognitive model sends a command to its Second Life 'body' (e.g. "move toward the fountain", "raise left arm .2m", "push the block object"), and receives information back about the state of the world, as to the time of the state. The model may require information as to whether its last action has been performed, and whether the HTTP responses are in order. This is easily resolved by sending a timestamp along with the last performed action from the body script to the model.

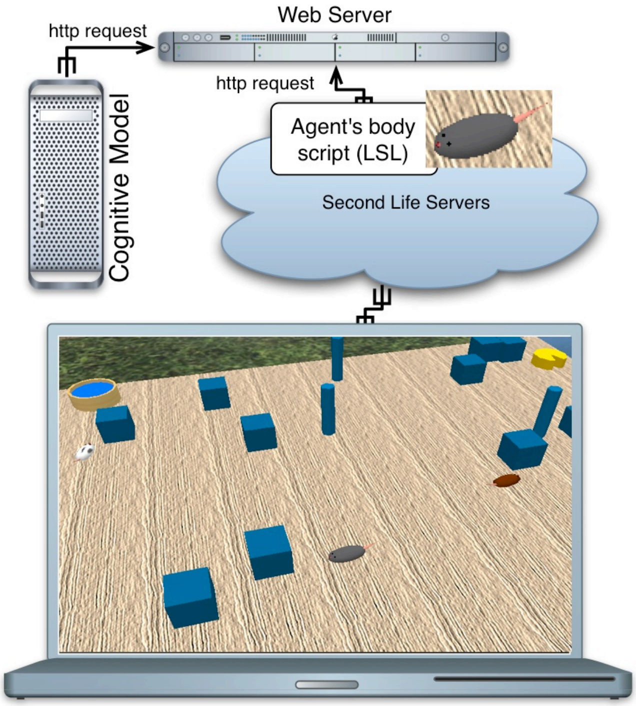


Figure 2. Second Life setup for models on DHCP or behind a firewall. Simulation shown at bottom has 3 models exploring a maze with cheese and water.

Memory Issues

Second Life scripts are relatively restricted in memory (16KB total for Byte-code, Stack, Free Memory, and Heap). This may be a serious restriction for collecting data about the state of the agent and keeping a copy of the prior state (prior state information may be necessary to avoid sending unchanged information to the model, saving both speed and bandwidth). This is not an issue when world-state contains only the last taken action plus the names of a few surrounding objects, but becomes an issue when collecting all possible information (object id, name, description, position, direction, velocity, dimensions, etc.) for a large number of objects.

Scanning

Other complications may arise in the way that a model in Second Life may be allowed to scan around for nearby objects. The scan is performed as a sphere, rather than a cylinder. This may take unnecessarily long for a large radius. A smaller radius may be scanned for a simulated sense of smell, but for long-distance vision, scanning must be restricted from a sphere to a smaller cone.

Speed

The greatest complication is that the perception-action protocol can take a relatively long time. This, of course, depends on the setup of scanning and HTTP requests. The greater bottleneck seems to be the maximum rate of HTTP requests (capped at 25 requests in 20 seconds). The assumption in modern cognitive architectures (e.g. Anderson & Lebiere, 1998) is that visual information is used at most 10 times per second (50ms for attention shift, and 50ms for attending the information). Thus, it seems that Second Life vision is about 10 times slower than may be desired for real-time cognitive models. This is not a major problem for interacting with static objects or other (similarly retarded) models, but it is a problem nonetheless. However, the technical support enjoyed by the Second Life community carries the promise of near-future solutions for these issues.

Specifics of Sample Simulations

Simulation 1

The first simulation was attempted to examine how a cognitive architecture may be interfaced with Second Life. The ACT-R cognitive architecture (Anderson & Lebiere, 1998) was connected with a Second Life script through an intermediary web server, as displayed in Figure 2. A scripted object was created in Second Life that would scan the world every few seconds, and send the state of the world via an HTTP call to the intermediary web server. On the ACT-R side, a cognitive model, in a perceive-think-act loop, would request an updated world-state from the intermediary web server, decide upon an action, and send motor commands back to the server.

Second Life Setup

The Second Life scripted object was attached to an avatar for greater exploratory capabilities (without an avatar scripted objects are restricted from many lands). The script performed a regular scan of nearby objects with a radius of 2m. If less than 5 objects were detected, the radius was increased, and another scan was re-initiated, until at least 5 objects were detected. Much more information was collected and transferred to the ACT-R model than was necessary for this simulation (e.g. object position, velocity, size, etc.), as this helped to examine the technical limitations of the setup. In addition, information sent to ACT-R included a timestamp, and the latest received motor command.

ACT-R Interface and Model

ACT-R visual and motor components were interfaced for Second Life in the following manner. Lisp functions were added to send out motor commands to the intermediary web server, and to retrieve world-state from the server. The ACT-R visual information (visicon) was filled with Button objects, each Button containing the name of its corresponding Second Life object found in the world-state list from the server. Upon clicking one of the button objects, a command would be issued, via a call to the web server, to move toward the corresponding object.

The ACT-R model employed to examine this interface was the Goal-Proximity Decision model (Veksler, Gray, & Schoelles, 2009). The details of the model are not provided here, as this is tangential to the focus of this paper. The general idea of the model is that it attends all objects in the visicon, and clicks the object with the greatest strength of association to the current goal (plus or minus noise). The strength of association between objects, in turn, is updated based on experienced temporal proximity of those objects.

Simulation Results

The Second Life script was first limited to find only the objects that belonged to its owner, which comprised sixteen randomly distributed boxes that served as navigational landmark (Figure 3). The model was presented with each of the sixteen objects as its goal, one at a time, until it successfully found each object.

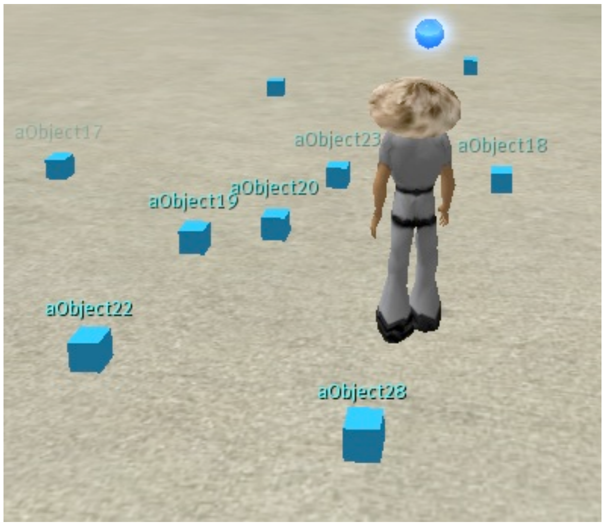


Figure 3. Second Life simulation. Controlled environment, with researcher-owned objects.

Upon the successful completion of this exercise, the scanning restrictions were removed, allowing the model to 'see' all objects within its scanning radius. The model was moved to an object-rich region, Washington Square Park in Manhattan (Figure 1), and manually given various objects as its goals (e.g. fountain, bench, store). Although the model was able to successfully navigate most of the region, some distant objects were unreachable due to the chosen scanning procedure. Thus long-distance vision, as discussed in the Scanning section above, may be necessary for most exploratory agents.

Simulation 2

The purpose of the second simulation was to examine whether Second Life can be set up to help distinguish between sets of model parameters for a Reinforcement Learning model. Reinforcement Learning (Sutton & Barto, 1998) is a widely implemented model of trial-and-error behavior. The specific form of Reinforcement Learning that was implemented in this model was a closed-form version of the ACT-R decision/utility-learning mechanism. The model chose which object to approach based on the utility of that object, given the specific goal (plus or minus noise). Upon reaching its goal, the model updated the utility of all encountered objects for reaching that goal based on the ACT-R utility-learning equation (Bothell, 2008).

The details of the model are relatively tangential to the focus of this work. What is important, however, is that there are several free parameters in this model (e.g. exploratory noise, learning rate, etc.), and that the same parameter values may cause different behavior for different task environments. Thus a high-fidelity task environment, like Second Life, may be necessary to distinguish between different parameter sets.

Different Task Environments

Parameter searches were performed with the model using three different maze structures. Each maze contained sixteen available spaces for the model to explore. The mazes were rated according to the average difficulty of finding each possible maze location from each possible starting point, by means of a random walk. The easy, medium, and difficult mazes required on average 181.39, 369.83, and 793.79 steps, respectively. The easy and medium difficulty mazes were set up in a grid-like fashion, with bidirectional movement allowed between any two neighboring locations. The easy maze, shown in Figure 4A, allows movement in all eight directions to its neighboring cells (N, NE, E, SE, S, SW, W, and NW). The medium difficulty maze, shown in Figure 4B, allows movement in four directions (N, E, S, W). The difficult maze, shown in Figure 4C, was set up with unidirectional and bidirectional connections, without regard for grid consistency.

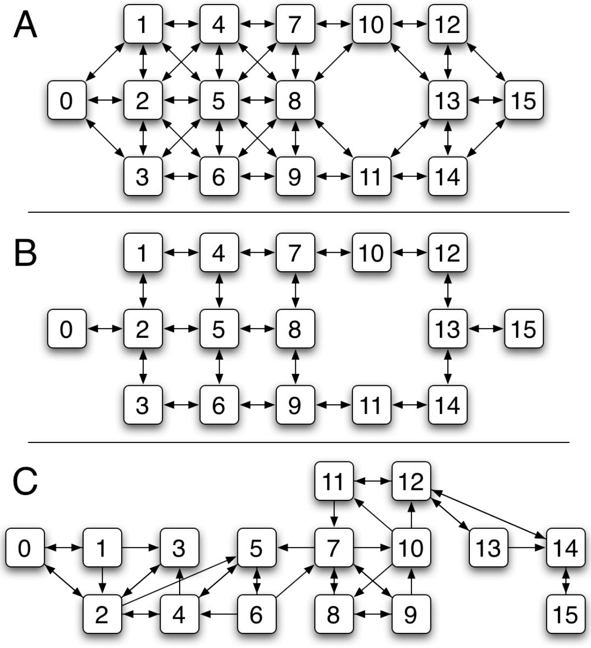


Figure 4. Sample navigation task environments. Numbered boxes signify locations, arrows signify the directions in which an agent may travel.

Different Parameter Sets

The model ran through each of the three tasks 60 times for each parameter set (noise was varied between .01 and 30, learning rate was .001 and .2). Each model run consisted of five bins, where the model had to reach sixteen goal in each bin (every possible location was set as a goal, in random order). The best performance (as measured by the average number of steps taken by the model to reach a goal in bin 5) for each maze was achieved with a different set of values for the free parameters in the model. The best parameters for the easy maze (paramsEasy) was achieved when the noise parameter was set to 5 and the learning rate was .1, for the medium difficulty maze (paramsMed) when the noise parameter was 25 and the learning rate was .1, and for the difficult maze (paramsHard) when the noise parameter was 15 and the learning rate was .01. A 3x3 two-way ANOVA revealed a significant interaction effect of ParameterSet × MazeDifficulty, $F(4, 531) = 115.42$, $p < .001$, a significant effect of ParameterSet, $F(2, 531) = 167.60$, $p < .001$, and a significant effect of MazeDifficulty, $F(2, 531) = 346.52$, $p < .001$. Post-hoc Tukey HSD comparisons revealed significant differences between the performance of all three parameter sets at the $p < .05$ level.

Second Life Simulation

Given the differences between paramsEasy, paramsMed, and paramsHard on the three types of task environments, it may be appropriate, and even essential, to establish which parameter set is best in a high-fidelity task environment. A Second Life simulation was set up as a proof of concept. Figure 2 is an accurate representation of the modeling setup, with a connection through the intermediary web server, with the models being represented as mice in a maze, with random poles and boxes (serving as landmarks), and three possible goals: swiss cheese, cheddar cheese, and water bowl. The complexity of the task, as well as its fidelity, was augmented with a greater number of objects and the presence of dynamic objects (other mice). The model was minimally altered so as to receive perceptual information from Second Life, and send motor commands back (the perception and action functions from Simulation 1 were reused).

The focus here is (1) to point out that choosing a task environment for cognitive modeling is not trivial (2) that Second Life, in theory, is an appropriate environment for task simulations, and (3) that Second Life, in practice, can be successfully interfaced with cognitive models. On the latter point, the model ran once with each of the three parameter sets, each run consisting of ten bins, where each bin comprised finding the three goals, one at a time, in random order. Early results (see Figure 5) suggest that the three parameter sets eventually converge, but this may take an extremely long time (≈27 walks through the maze, which, at worst, is almost 3000 steps). The average number of steps taken to reach a goal is 35.9 for paramsMed, 99.4 for paramsEasy, and 148.5 for paramsHard.

These results are not interpretable without more data, nor, even if the trend should continue, could we assume that the medium difficulty maze shown in Figure 4B may be used in place of high-fidelity task environments. Instead, the suggestion is that these task environments should be used in combination: one to quickly test many models and parameter values, the other to test whether a model could scale up to the complexities of dynamic and uncertain worlds.

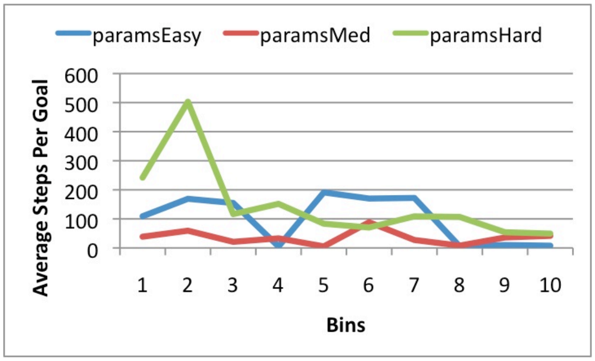


Figure 5. Second Life simulation results from three different sets of parameter values.

Summary

In sum, Second Life may be an important tool for cognitive modeling. It provides a better balance of real-world complexity and constraints than simpler simulation environments, less hassle and financial investment than robotics work, and it stands out from other 3D virtual world with a rich, massive-multiuser environment, and extensive technical support. The Second Life environment may be easily interfaced with cognitive architectures, as described in Simulation 1, or with closed form models, as described in Simulation 2. As Simulation 2 suggests, Second Life modeling work can help to answer questions as to the fidelity of various cognitive mechanisms and/or parameter values whose performance may vary in different task environments.

Future work will address rigorous statistical comparison of model performance in Second Life versus other task environments. Other plans include implementation of long-distance visual scanning coupled with head-movements, and exploration of a greater variety of tasks (e.g. soccer, building blocks, hide and seek).

Acknowledgements

I would like to thank Wayne D. Gray and Michael J. Schoelles for their theoretical and editing contributions to this work. The work was supported, in part, by grant N000140710033 to Wayne Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer. Much of this work was carried out on space provided at the Second Life AI Laboratory (SLAIL) on Daden Cays in Second Life - http://slurl.com/secondlife/Daden%20Cays/152/44/22.

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

Bothell, D. (2008). ACT-R 6.0 Reference Manual: Working Draft. Retrieved April 13th, 2009, from http://act-r.psy.cmu.edu/actr6/reference-manual.pdf

Braga, A. P. S., & Araujo, A. F. R. (2003). A topological reinforcement learning agent for navigation. *Neural Computing & Applications, 12*(3-4), 220-236.

Burden, D. J. H. Deploying embodied AI into virtual worlds. *Knowledge-Based Systems, In Press, Corrected Proof.*

Category:LSL XML-RPC - Second Life Wiki. (2009). *Second Life Wiki* Retrieved April 13, 2009, from http://wiki.secondlife.com/wiki/Category:LSL_XML-RPC

Merrick, K., & Maher, M. L. (2009). Motivated Learning from Interesting Events: Adaptive, Multitask Learning Agents for Complex Environments. *Adaptive Behaviour, 17*(1), 7-27.

Rensselaer Polytechnic Institute. (2008). Bringing Second Life To Life: Researchers Create Character With Reasoning Abilities Of A Child. *ScienceDaily.* Retrieved April 14, 2009, from http://www.sciencedaily.com/releases/2008/03/080310112704.htm

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press.

Veksler, V. D., Gray, W. D., & Schoelles, M. J. (2009). *Goal-Proximity Decision Making: Who needs reward anyway?* Paper presented at the 31st Annual Meeting of the Cognitive Science Society, CogSci 2009.

Voicu, H., & Schmajuk, N. (2002). Latent learning, shortcuts and detours: a computational model. *Behavioural Processes, 59*(2), 67-86.

Group behaviour of virtual patients in response to therapeutic intervention in an agent-based model of dementia management.

Victor Vickland (victor.vickland@unsw.edu.au)
Dementia Collaborative Research Centre, University of New South Wales
45 Beach St. Coogee NSW 2034 Australia

Henry Brodaty (h.brodaty@unsw.edu.au)
Dementia Collaborative Research Centre, University of New South Wales
45 Beach St. Coogee NSW 2034 Australia

Abstract

This project created virtual patients who can respond to hypothetical therapeutic interventions in an agent-based model of dementia management. We evaluated the overall response of patients by collecting statistics and observing their group behaviour. In this model virtual patients were actively seeking treatment for symptoms of depression associated with dementia. Responses to hypothetical therapeutic interventions consisted of both generic (common to all patients) and individual (modified for each patient) components. The preliminary results show that even simple sets of rules governing behaviour of virtual patients can lead to quite complex responses at the group level. Furthermore, the lessons learned from monitoring the group behaviour provided valuable feedback which is now being used to modify the creation of individual virtual patients e.g. implementation of histories of previous successful and unsuccessful treatments.

Keywords: Virtual Patients, Dementia Management, Decision Support.

Introduction

Computer models are now frequently applied in medicine and public health policy. For example forecasting of prevalence and incidence of specific diseases is performed routinely with the aid of computer tools. The application of agent-based modelling is not yet as popular but potential benefits of such approaches have already been recognised in such areas as computational biology, computerised clinical guidelines and modelling of specific symptoms in disease conditions (Kitano, 2002). .

Virtual patients as intelligent reactive agents

Implementation of software-generated agents as virtual patients in computer simulations is now well established (Huang, Reynolds & Candler, 2007). There is still an ongoing debate about what constitutes intelligent behaviour but it is reasonably well accepted that autonomous agents which are able to respond to changing environment can be classified as ‘intelligent'. However, more precise definitions are needed in particular with the onset of modelling of social behaviours (Decety & Grezes, 2006). It has been demonstrated experimentally that important physiological characteristics of real patients can be mapped and modelled accurately (Grinberg, Anor, Madsen, Yakhot & Karniadakis, 2008). Various attempts have been made in the past two decades to also include more complex social behaviours. In such models an expected range of behaviours may include perception of emotional and cognitive states of other agents (Meyer, 2006). In real life, decisions made by individual patients in response to a changing environment and severity of symptoms can be complex and interdependent. In clinical settings for example the onset of depression symptoms in dementia patients may trigger a sequence of events leading to hospitalisation which in turn may trigger further changes to a patient’s life. Such a chain of events may be reversible in some individuals but in others may lead to severe limitation of future life choices. It would be advantageous to have similar complexities reflected by a set of rules describing behaviour of virtual patients in computer modelling projects.

Group behaviour in an agent-based model

The definition of group behaviour is not clear and different researchers put emphasis of different aspects of behaviour that are not predicted beforehand (Wu, Hu, Zhang & Fang, 2008). In its simplest form it is just the ‘average’ behaviour of the group, no more than sum total of the entire population. However if virtual patients become more autonomous e.g. their trajectory reflects their past history of symptoms, then their behaviour may become much less predictable. Health policy makers are predominantly interested in the overall response of larger populations to treatment options. They want to estimate the potential health and economic benefits of future health initiatives (Edge, 2008). It is generally accepted by health policy makers that the group is a collection of “typical” individuals; therefore what is therapeutically beneficial to the group will also be beneficial to the average individual.

Predicting outcomes of therapeutic interventions

Treatment of symptoms of depression in dementia patients is complex and factors causing symptoms are often unknown. Therapeutic interventions fall broadly into two groups: pharmacological, e.g. antidepressant medication, and non-pharmacological, e.g. cognitive-behavioural therapy, environmental improvement or increased interactions with others in daily activities (Zec & Burkett 2008). Depression is frequently associated with dementia and around 20-50 % of patients will suffer from depression at various levels of severity and duration during the course of their decline (Zubenko, Zubenko, McPherson, & Spoor, 2003). It is beneficial to diagnose depression early and treat symptoms effectively. The costs associated with treatment can be modest if a patient is just given an antidepressant. However, delayed or inappropriate treatment can interfere with recovery, which can be costly in personal and financial terms. Therefore only well proven treatments are accepted for implementation. However there is uncertainty about how effective different therapeutic strategies are for individual patients and if they have any cumulative effect or synergistic action when two different interventions are combined. Not all patients respond equally to even well proven pharmacological interventions (Bains, Birks & Dening, 2002). Similarly patient responses to less effective but long lasting treatments such as environmental changes and psychological interventions are even less predictable. Accurate projections of outcomes derived from such interventions are very difficult to make. Therefore clinicians and health policy makers could benefit from forecasts made with an aid of computer models.

Aims

The aim of this project was to test the following assumptions: (a) essential parameters of therapeutic interventions can be implemented into an agent-based model as a cluster of global variables and simultaneously available to all agents in the model, (b) the short and long term outcomes of hypothetical therapeutic interventions can be detected and estimated from the emergent behaviour of a large group of virtual patients. These assumptions were tested in the laboratory setting by using an existing model of dementia management and introducing an optional functionality of virtual treatment intervention. This paper presents interim results and hopes to contribute to the future design of virtual patients.

Methodology

The AnyLogic simulation software was used as a programming tool to build the model (http: //www.xjtek.com). Ten thousand virtual patients were initialised at the start of the experiment with characteristics such as age, gender, severity of dementia and severity of depression. Each patient was initialised with a different set of parameters according to probability distribution tables specific to the population of people with dementia in Australian context. The time-step of the model was 1 week and the model was allowed to run for maximum 1500 steps which is equivalent of around 30 years. The virtual patients behaved with relative autonomy and were able to respond to changes in their environment, most importantly to the introduction of new therapeutic interventions. The computer interface was developed as part of the BPSD management project at Dementia Collaborative Research Centre, Faculty of Medicine, UNSW Sydney (http://bpsd.dementia.unsw.edu.au/models).

Virtual Patient

The blueprint for the patient's behaviour was expressed by statecharts, variables and functions as shown on Figure 1. It covered such characteristics as age, gender, severity of

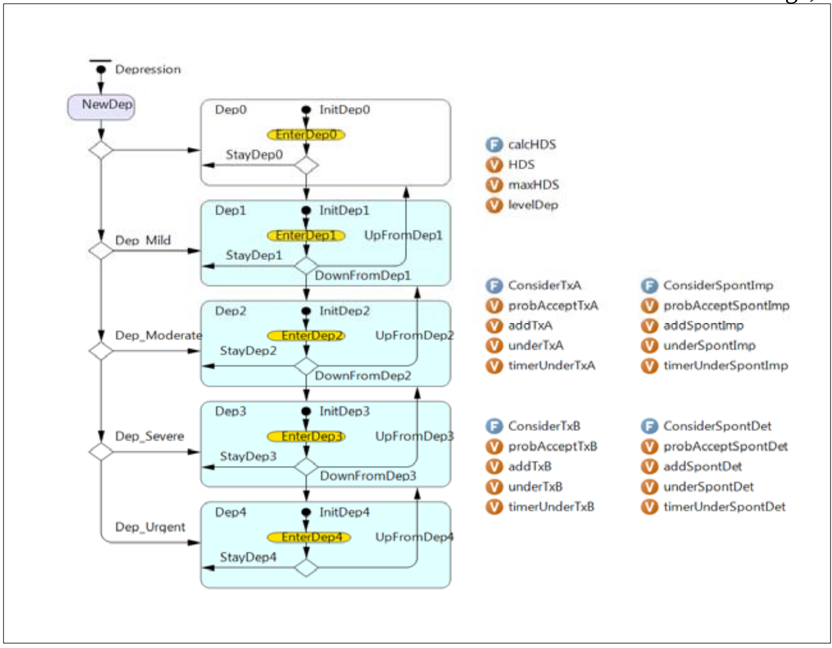


Figure 1. An example of the statechart with variables (V) and functions (F) which govern the behaviour of each individual patient.

dementia, severity of depression, chronic health status and place of residence. At the time of initialisation of the model each patient was allocated with randomly selected characteristics. During the run time of the model patients acted autonomously and they were constantly re-evaluating their own status e.g. rules and all functions were called to recalculate variables and send messages. The AnyLogic simulation engine which underpinned the computer simulation took care of synchronisation and parallel execution of all agents and their interaction with the environment. Figure 1 illustrates some of the components of the virtual patients that relate to the symptoms of depression and acceptance of treatment interventions. The overall design of virtual patients included five other statecharts with numerous functions and variables used to determine agent behaviour and graphical display during animation.

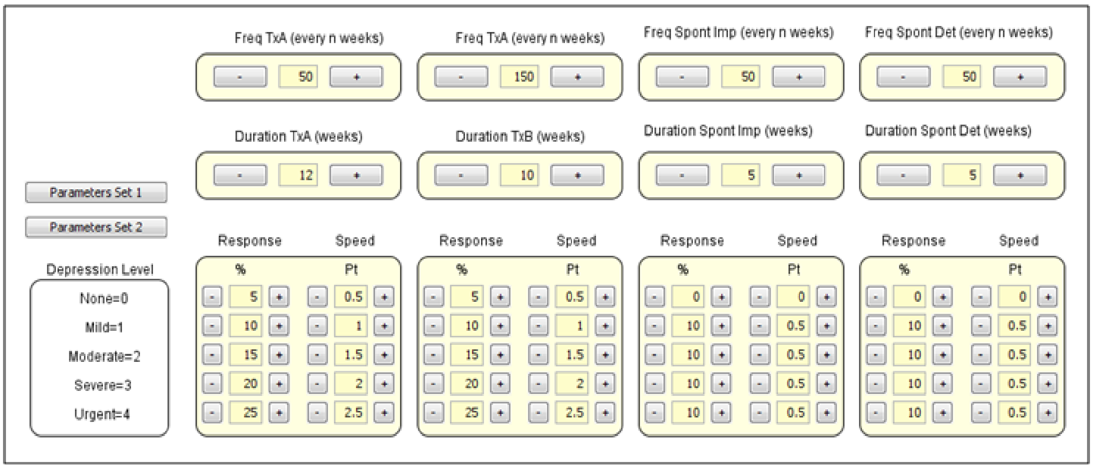
The transitions between states were driven by a set of rules which were identical for each patients therefore leading to a generic response. As time progresses each patient modifies his or her own characteristics according to choices made in previous steps resulting in individualised responses. Therefore the overall response of the patient is a mixture of both generic and individual components with increasingly variable behaviour.

Therapeutic intervention

Dynamic changes in virtual patients' behaviours were triggered by access to therapeutic interventions. The goal of each patient was to reduce the severity of depression if treatment was available. Two hypothetical interventions were available in the model: intervention TxA being equivalent to non-pharmacological treatment of depression e.g. training of nursing staff on how to increase social participation of patients, and intervention TxB representing pharmacological treatment e.g. prescription of an antidepressant such as sertraline (Bains, Birks & Dening, 2002). Intervention TxA had a weaker therapeutic effect (0.3) but was applied for much a longer period of time than intervention TxB, which had larger effect size (0.5) but was available only for a maximum of 12 weeks within a period of 3 years (Bains, Birks & Dening, 2002). The virtual patient had a choice of accepting the treatment and benefiting from it at the rate specified by an initial setting through the user interface as illustrated on Figure 2. The accuracy of modelling therapeutic interventions strongly depended on the accuracy of the parameters that were used to characterise different aspects of these interventions. For example it is known from the literature that depressed patients respond differently to treatment when their symptoms are at different severity levels. The speed of recovery may be initially very fast and then may slow down with the patient remaining mildly depressed for a longer period of time or may even stop responding to treatment (Bains, Birks & Dening, 2002). The user interface also included options to enable spontaneous improvement and/or spontaneous deterioration. Each patient was assigned with a randomly selected probability of responding to such improvement or deterioration.

Interaction between virtual patients and therapeutic interventions

10,000 virtual patients were initialised at the beginning of the experiment and each patient acquired general characteristics common to all. The required characteristics were either taken from look-up tables or were randomly allocated if appropriate. For example the initial severity of depression was randomised but mortality rates were taken from a table according to patient's age group. Once initialised, all agents behaved autonomously. In the current version of the model virtual patients do not communicate with each other but they communicate with the environment. They actively seek treatment if the severity of their symptoms is above a certain threshold. Each virtual patient can accept therapy if the required therapy is available during the patient's lifetime. Figure 3 illustrates



Freq TxA (every n weeks)	Freq TxA (every n weeks)	Freq Spont Imp (every n weeks)	Freq Spont Det (every n weeks)
50	150	50	50

Duration TxA (weeks)	Duration TxB (weeks)	Duration Spont Imp (weeks)	Duration Spont Det (weeks)
12	10	5	5

Depression Level	Response %	Speed Pt	Response %	Speed Pt	Response %	Speed Pt	Response %	Speed Pt
None=0	5	0.5	5	0.5	0	0	0	0
Mild=1	10	1	10	1	10	0.5	10	0.5
Moderate=2	15	1.5	15	1.5	10	0.5	10	0.5
Severe=3	20	2	20	2	10	0.5	10	0.5
Urgent=4	25	2.5	25	2.5	10	0.5	10	0.5

Figure 2. The user interface for setting up attributes of two therapeutic interventions.

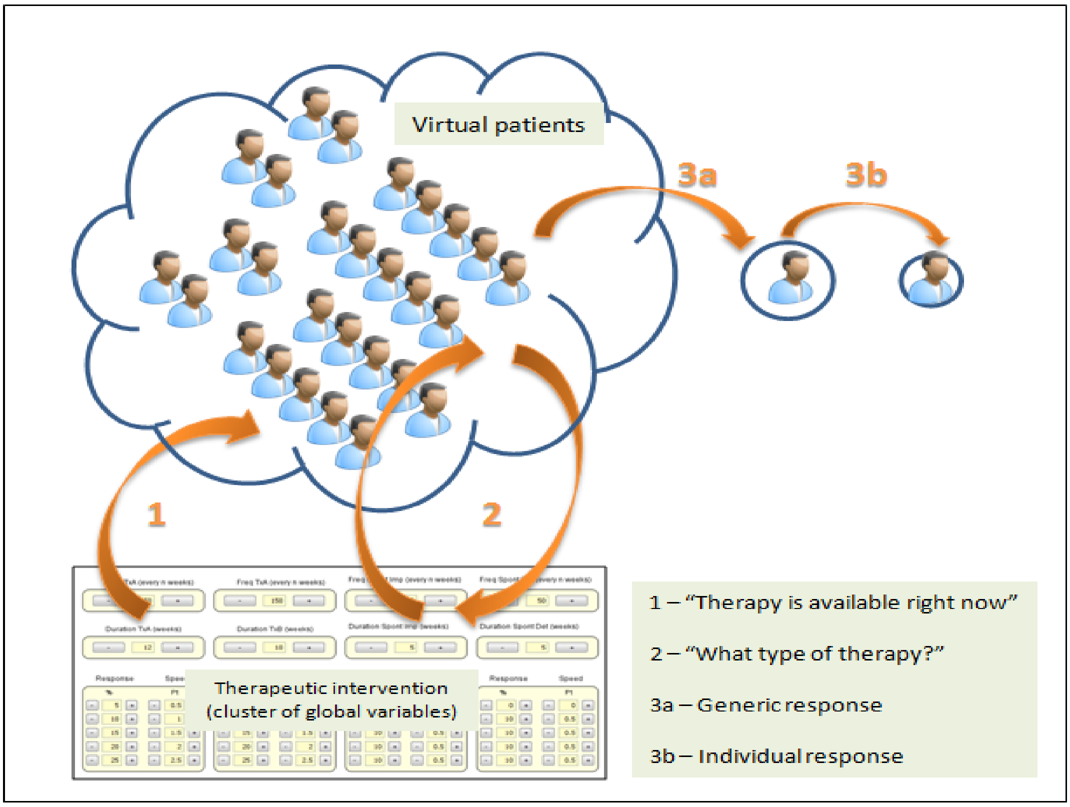

Virtual patients
3a
3b
1
2
Therapeutic intervention
(cluster of global variables)
1 – "Therapy is available right now"
2 – "What type of therapy?"
3a – Generic response
3b – Individual response

Figure 3. Sequence of steps in communication leading to therapeutic response.

this process of periodic checks of the availability of treatment. The group behaviour of patients was monitored continuously during the experiment by acquiring relevant statistics from each patient, for example, the overall number of patients was monitored at each level of symptom severity. Results were plotted simultaneously and analysed for differences. The availability of TxA and TxB was switched on and off via a button on the user screen but it could also be triggered by a timer at specific time intervals. The characteristics of therapeutic interventions were expressed as a cluster of global variables which the experimenter could modify before running the model.

Results

Only preliminary results of the experiments are presented in this paper. They consist of responses of five groups of patients selected by the increased level of severity of symptoms. At the time of initialisation of the model allocation of the patient to each of the severity levels was 39% with no symptoms of depression, 30 % with mild, 20 % with moderate, 10 % with severe and 1% with depression so severe that it required urgent intervention. Each virtual patient who responded to treatment contributed to the statistics for these levels as they either improved or deteriorated with their symptoms. Patient who did not respond to treatment remained at the same level unless they randomly responded to spontaneous improvement or deterioration.

Response without therapeutic intervention

A graph presented in Figure 4 shows the generic behaviour of virtual patients over time in the absence of any therapeutic intervention. The number of patients with a particular level of severity remains almost the same through 1500 steps (weeks) of the model's runtime. Some variability of the numbers is associated with the stochastic nature of the patient's behaviour. For example new patients were constantly initialised according to projected increases in population, while other patients were dying in accordance with age-dependent mortality rates. Spontaneous deterioration and spontaneous improvement in symptoms of depression were also contributing to small changes in baseline percentages. It is important to mention that all virtual patients had the capacity to make decisions e.g. accept the treatment and to decrease symptoms of depression over time. However such decisions could not be made until the therapeutic intervention was available. For example if the patients belief was indicating preference of TxA (non-pharmacological treatment) and no such intervention was made available or only TxB was available, the patient continued without treatment . The same situation would occur in the case of preference for TxB when only

TxA was provided. Therefore the results in Figure 4 would be the same when another therapeutic intervention was available but none of the patients accepted it.

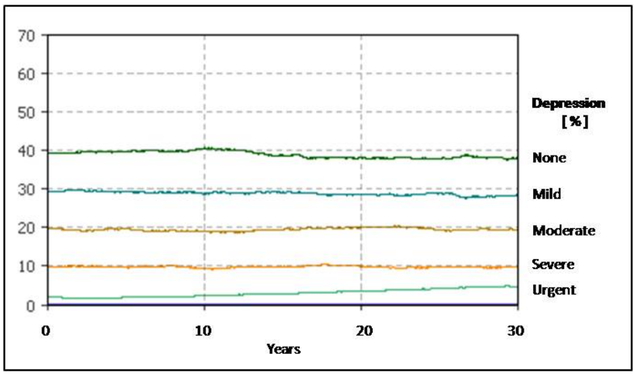


Figure 4. Percentage of virtual patients with different levels of symptom's severity.

Response to therapeutic intervention

The response of virtual patients to the introduction of therapeutic intervention was certainly not homogenous and at the group level changes are clearly visible. Figure 5 shows changes in five groups of patients according to their level of symptom’s severity.

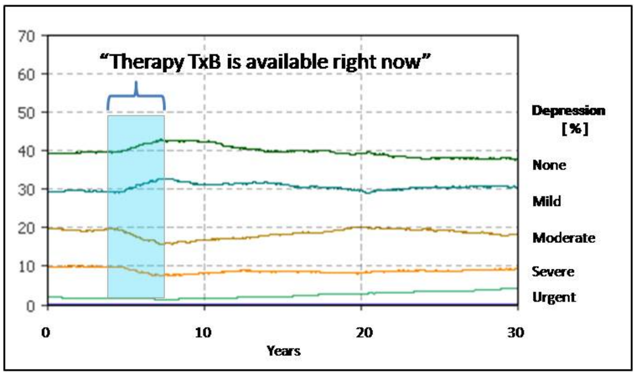


Figure 5. Response of virtual patients to therapeutic intervention TxB.

An improvement in moderate and severe groups (20% and 10% baseline) is indicated by a decrease in number of patients at these levels. However the numbers of patient with mild or no symptoms show strong increases. This can be easily explained when we consider that when patients with severe symptoms improve they “move’ to the moderate group, and if improvement continues they move again to the group with mild depression. A similar situation occurs with patients at moderate and mild levels. However when the patient improves and reaches the mild severity level intervention TxB becomes less effective. Therefore many more patients remain mild instead of reaching the top level without any symptoms. When the intervention is no longer available then there is a slow reversal of improvement and after some considerable delay all levels return to their baseline values. This delay is in itself an interesting phenomenon driven primarily by mortality of the patients who previously improved but no longer contribute to the statistics after dying. Therefore initialisation of newly diagnosed patients with symptom severity assigned according the distribution of 39, 30, 20, 10 and 1 percent will gradually return the distribution to a baseline level.

Discussion

There is an increasing demand for new methods for evaluation of therapeutic interventions and in particular their effectiveness at the population level over time. The incidence of depression is on the increase therefore foreseeing outcomes of potential interventions could have beneficial effects on future policy making and costs. The preliminary results of our experiments indicate that such evaluations are plausible and that estimates could be made long before any real-life clinical trials are implemented. The value of virtual experiments will be in selecting the most probable clinical scenarios for therapeutic interventions e.g. single vs. combined interventions which are implemented over longer or shorter periods of time.

Computer models are effective tools for making forecasts and are routinely used in marketing and economics. However they are less popular in medicine mainly due to much greater complexity and unpredictable nature of human behaviour. We tested the possibility of conducting experiments on populations of virtual patients and foreseeing outcomes of hypothetical interventions. Most exciting was the possibility of monitoring a large population of virtual patients and their group or ‘collective’ response to the same event. We made the distinction between general response and individual response. The difference was in the amount of specific rules by which a virtual patient made the decision of accepting and responding to particular type of treatment. In real life that is indicated by personal beliefs which patients may have e.g. strong preference for one type of therapy.

The agent-based model was stable in performance and fast enough to accommodate a large number of virtual patients. This gives us the possibility of further development of much more complex rules governing patient behaviour and designing much more realistic environments where key players such as doctors, nurses and hospital services are also modelled. The next step in the development of the virtual patients will be introduction of history of responses to therapeutic interventions and linking them with decision making algorithms.

There are number of limitations in the design of this study and ways in which these experiments were conducted. First, there is a question regarding the ‘autonomy’ of patients in this model. It is essential to emphasise that the set of rules governing behaviour was identical for each agent. Individual behaviour was shaped by the decisions made by each agent during the runtime of the model. Some of these decisions where based on randomly assigned values e.g. probability of spontaneous recovery and other decisions were expressions of patient's beliefs e.g. preference for pharmacological interventions when symptoms were moderate or severe. Second, the environmental trigger in a form of a message 'Therapy TxB is available right now' was continuously monitored by each agent but did not automatically invoke change in behaviour.

Third, validation of the model is an issue that can't be easily resolved. Our primary effort was in modelling individual response to the therapeutic interventions. We know quite a lot about individual responses from published medical literature. However there is little understanding of group treatment behaviour in this domain. In contrast validation of consumer behaviour in marketing models can be done by using sales figure and attributes of the purchased products. Unfortunately there is no data which will accurately describe what the group behaviour of real patient choices under a particular treatments should look like. In fact the whole purpose of building the model and conducting virtual experiments was to get better understanding of what this group behaviour might be. Perhaps our effort in this modelling project will be rewarded in future by the next generation of research projects which originated from the results of virtual experiments. By showing clinicians what the plausible future might be we could expect that real-life clinical trials will be strongly influenced and guided by the results of in-silico experiments.

Acknowledgements

This project has been supported by the funding from the Australian National Health and Medical Research Council for 2008-2010.

References

Bains, J., Birks, J., & Dening, T. (2002). Antidepressants for treating depression in dementia. Cochrane Database Syst Rev (4): CD003944.

BPSD Project http://bpsd.dementia.unsw.edu.au/models

Decety J. & Grezes J. (2006). The power of simulation: imagining one's own and other's behaviour. Brain research 1079(1) 4-14.

Edge, L. (2008). Time to confront the global dementia crisis. Lancet Neurology 9(7) 761.

Grinberg L., Anor T., Madsen J., Yakhot A. & Karniadakis G. (2008). Large-scale simulation of the human arterial tree. Clin Exp Pharmacol Physiol 2(36) 194-205.

Huang G., Reynolds R. & Candler C. (2007). Virtual patient simulation at US and Canadian medical schools. Academic Medicine 82(5) 446-451.

Kitano, H. (2002). Computational systems biology. Nature 420(6912), 206-210.

Meyer, J. (2006). Reasoning about emotional agents. Int J Intelligent Systems 6(21) 601-619.

Wu, J., Hu, B., Zhang, J. & Fang, D. (2008). Multi-agent simulation of group behavior in E-Government policy decision. Simulation Modelling Practice and Theory 10(16) 1571-1587.

XJ Technologies http://www.xjtek.com/

Zec RF. & Burkett NR. (2008). Non-pharmacological and pharmacological treatment of the cognitive and behavioral symptoms of Alzheimer disease. NeuroRehabilitation 23(5) 425-438.

Zubenko, G., Zubenko, W., McPherson, S. & Spoor, E. (2003). A collaborative study of the emergence and clinical features of major depressive syndrome of Alzheimer's disease. Am J Psychiatry 160: 8570866.

A Computational Model of Preverbal Infant Word Learning

Guillaume Aimetti (G.Aimetti@DCS.Shef.AC.UK)
Speech and Hearing Group, Regent Court, 211 Portobello
Sheffield, S1 4DP UK

Roger K. Moore (R.K.Moore@DCS.Shef.AC.UK)
Speech and Hearing Group, Regent Court, 211 Portobello
Sheffield, S1 4DP UK

Keywords: Early Language Acquisition; Word Learning; Automatic Segmentation; Cross-Modal Association; Dynamic Time Warping.

Introduction

This work investigates a novel computational model of preverbal infant word learning in an attempt to create a more robust speech recognition system. Currently, the state-of-the-art can be extremely accurate when used in its optimal environment. However, when taken out of its comfort zone accuracy significantly deteriorates and does not come anywhere near human speech processing abilities, even for the simplest of tasks. We take inspiration from the ease with which newborns are able to learn words, with no apparent difficulty, and develop into expert communicators of their native language.

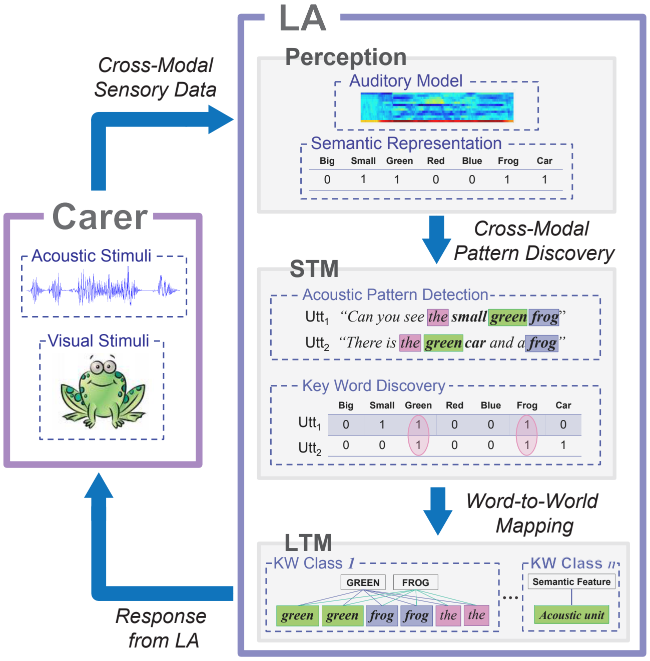
In order to learn words, the young language learner must be able to segment speech into useful units and then associate them to visual referents from within their environment (Smith & Yu, 2008). The model described here, the Acoustic DP-ngrams, attempts to solve the word-to-world mapping problem through cross-modal (acoustic & visual) associative learning set within an interactive framework, as illustrated in figure 1 (for a more technical description of the system see (Aimetti, 2009)).

Initial results show that there is significant potential with the current algorithm, as it segments in an unsupervised manner and does not rely on a predefined lexicon or acoustic phone models that constrain current Automatic Speech Recognition (ASR) methods. The learning process concurs with current cognitive views of early language acquisition (Jones, Hughes, & Macken, 2006; Saffran, Aslin, & Newport, 1996; Saffran, Werker, & Werner, 2006; Smith & Yu, 2008), and the key word detection experiments exhibit similar behaviours apparent in developing preverbal infants (Gomez & Gerken, 2000; Kuhl, 2004; Newman, 2008).

The Computational Model

There are two key processes occurring within our learning agent (LA):

1. Automatic Segmentation: Acoustic DP-ngrams is used to automatically segment the speech, directly from the acoustic signal, into important lexical fragments by discovering *similar* repeating patterns. This approach uses a dynamic programming (DP) technique, dynamic time warping (DTW), to accommodate the temporal distortion present in speech. The advantage of this approach is that it uses an accumulative scoring system to measure the quality and length of the discovered fragments. This method is similar to the Segmental DTW algorithm developed to summarise recordings of academic lectures (Park & Glass, 2008).



Big	Small	Green	Red	Blue	Frog	Car
0	1	1	0	0	1	1

Utt$_1$ *“Can you see the **small green frog**”*

Utt$_2$ *“There is the **green car** and a **frog**”*

	Big	Small	Green	Red	Blue	Frog	Car
Utt$_1$	0	1	1	0	0	1	0
Utt$_2$	0	0	1	0	0	1	1

Figure 1: Word-to-World mapping set within an interactive carer-learner framework. LA’s internal memory is inspired by current cognitive views (Jones et al., 2006).

2. Word-to-World Mapping: Figure 1 shows the interaction between LA and its parent (carer). During training the carer incrementally feeds LA with cross-modal stimuli; the acoustic stream consists of continuous speech, as sampled data, and the visual stream consists of crisp tags, representing the visual referents within the utterance. Internal representations of the visual referents is achieved through the co-occurring events from both modalities, as suggested by Smith and Yu (2008). Each class is therefore emergent and constantly evolves with the accumulation of exemplar tokens, thus allowing the system to gradually become more robust to the variation present in speech.

Experiments

LA is trained with 480 cross-modal utterances from a single female speaker (F1); each utterance is passed to the system as sampled acoustic data in parallel with the crisp visual tag(s), representing the key word(s) that lie within it. To test the emergence and robustness of internal representations, LA is faced with a recurrent key word detection task throughout development. This is carried out as probe moments which occur every 20 utterances. LA is temporarily frozen and tested on 320 unobserved utterances from the known female speaker (F1) and 320 unobserved utterances from an unknown male speaker (M1). Only the acoustic part of the input is processed and LA must recognise the key word(s), responding with the correct visual referent(s).

Figure 2 displays the key word detection accuracy during the learning period, which is shown as the percentage of correct key word detections over the number of utterances observed. The blue plot with circles shows the F1 probe, the green plot with squares shows the M1 probe and the red discontinuous plot shows the chance level of a correct guess.

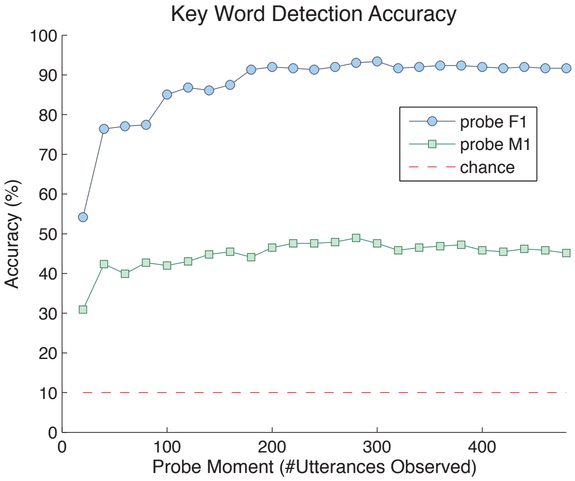


Figure 2: LA's key word detection accuracy throughout development. Probing is carried out every 20 utterances where LA is tested on a known (F1) and unknown (M1) speaker.

Internal representations can be seen to emerge very quickly from the plot in figure 2. After only 20 utterances LA is already able to detect key words well above chance level, achieving 54% for F1 and 31% for M1. Robust representations for F1 develop after 180 utterances, where key word detection accuracy reaches a plateau of 92%(±1%). However, internal representations for M1 seem to plateau after only 40 utterances and limited to a maximum of 49%.

Discussion & Future Work

This paper introduces a computational model of early word learning abilities in preverbal infants. The algorithm is able to successfully learn words in a cognitively plausible fashion. It is clear to see from the results that LA quickly builds up accurate representations to a familiar speaker F1, but is also still able to generalise above chance level to an unknown speaker M1 across gender with 40%to50% accuracy. This shows that without observing other speakers, the system is not able to build robust internal representations that can reliably generalise across speakers, as suggested by Newman (2008).

One downside to this technique is that it is unable to run on a large data-set as the exemplar tokens being stored in memory are unbound and tend to infinity. Currently, the authors are investigating a method to automatically build prototype representations for the most efficient units within the learners native language (i.e. with Hidden Markov Models). This agrees with current thinking that infants begin learning language attending to too much detail within their native language, and that prototype representations (an average of exemplar units stored in memory) occur with experience from a greater variety of speakers (Kuhl, 2004; Newman, 2008).

Acknowledgments

This research was funded by the European Commission, under contract number FP6-034362, in the ACORNS project (www.acorns-project.org).

References

Aimetti, G. (2009). Modelling early language acquisition skills: Towards a general statistical learning mechanism. In *Proceedings of the student research workshop at eacl 2009* (pp. 1–9). Association for Computational Linguistics.

Gomez, R. L., & Gerken, L. (2000). Infant artificial language learning and language acquisition. *Trends in Cognitive Sciences*, *4*(5), 178–186.

Jones, D. M., Hughes, R. W., & Macken, W. J. (2006). Perceptual organization masquerading as phonological storage: Further support for a perceptual-gestural view of short-term memory. *Journal of Memory and Language*, *54*(2), 265-281.

Kuhl, P. K. (2004, November). Early language acquisition: cracking the speech code. *Nature*, *5*, 831-843.

Newman, R. S. (2008). The level of detail in infants' word learning. In *Current directions in psychological science* (Vol. 17, p. 229-232). University of Maryland, College Park.

Park, A., & Glass, J. (2008). Unsupervised pattern discovery in speech. In *Trans. alsp* (Vol. 16, p. 186-197).

Saffran, J. R., Aslin, R. N., & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *SCIENCE*, *274*, 1926-1928.

Saffran, J. R., Werker, J., & Werner, L. A. (2006). Handbook of child psychology. In D. Kuhn & R. S. Siegler (Eds.), (6th ed., Vols. 2, Cognition, Perception and Language, p. 58-108). New York: Wiley.

Smith, L., & Yu, C. (2008). Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, *106*, 1558-1568.

An Extention of Elman Network

Shin-ichi Asakawa (asakawa@ieee.org)
Tokyo Womens' Christian University,
Centre for Information Sciences, 2-6-1 Zempukuji,
Suginami 1678585 Tokyo, JAPAN

Abstract

Although the Elman network is so powerful that it can deal with a variety of language processings, there exist some short comings about its ability. For example, the original Elman net cannot always deal with a long distance dependency appropriately, which is a number agreement between nouns and verbs with many relative pronouns in a sentence. This limitation might cause from the constraints of its structure of the context and the hidden layer, which can preserve only one time previous state of the network. Here, we propose an extension of the Elman network. The extended Elman network can preserve the *n*-th generations of inner states. When the model processed the corpus consisted of many relative pronouns with multi-center embeddings structure, it could deal with the long distance number agreement adequately. This model can be regarded as a natural extension of the Elman network in order to deal with complex structures of language.

Keywords: Elman network; memory capacity; number agreement; long distance dependency.

The structure of the Elman network

The network proposed by Elman (Elman, 1990; Elman, 1991) was a simple recurrent neural network (SRN). This network has an input layer in order to deal with the current input signals, and has a context layer which maintain past states. The contents dealt in the hidden layer at time t depend on both current inputs and past states of the hidden layer at time $t-1$. Therefore, it means that the context layer in the Elman network can maintain the whole history of all the past inputs. As a result, the state of the network at time t depends on both current inputs and the set of all the history of past inputs.

Since there exists a computational limitation (a limitation of precision), SRN can only represent finite regions in a problem space. Therefore, a number of discussions revealed that SRN cannot overcome the limitation of finite state machines. It is known that any attempts to let SRN learn context dependent grammar have, more or less, a limitation that SRN cannot reach the same result as generalized finite state machines. However, here we will try to show one of possible solutions to tackle this problem.

The device to deal with the language which has center embeddings must have memory stores to maintain complex time sequential information. A number of experimental studies revealed that SRN could recognize and learn regular languages (eg. Giles et al.,1992). The Elman network is simple but powerful in order to deal with the context dependent grammar by the ability of time development (see Fig.1). However, there is a limitation in the Elman network. It cannot deal with complex structure such as a long distance dependency like a number agreement between nouns and verbs in sentences with many relative pronouns (a nested center embedding structure). Although Elman found that the Elman network can deal the number agreement in center embedded sentences, it might be impossible to deal with many relative pronouns in multi-center embedded sentences by the limitation of the capacity or the state of the hidden and the context layers. Consider the sentence:

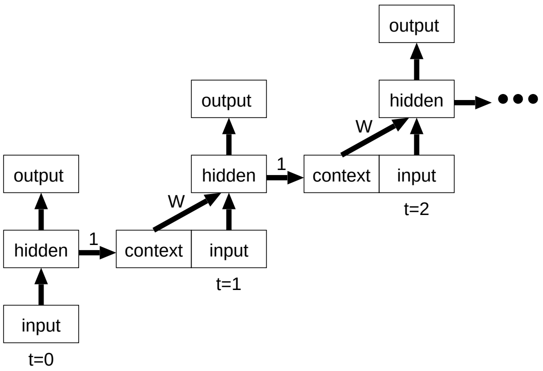


Figure 1: The time development of the Elman network

(1) The girl who chases Mary who feeds the girls who see the cat feeds.

The subject in this sentence 'girl' must agree a number with the verb (the last word 'feeds'). But there are three verbs ('chases', 'feeds', and 'see') between the subject and the predicate. The network to process this sentence properly must maintain the information of the number of the subject at the head of the sentence. The original Elman network might be difficult to maintain this kind of a long distance dependency information because of memory limitation.

Extension of context layers

The possible ways to overcome this problem are to extend volumes and contents of context layers in the network. There are two possibilities to extend context layers: (1) the extension of the number of units in the hidden and the context layers and (2) the extension of generations of context layers (Fig. 2). The extension of the number of units in the context layer is a simple solution for the network to find the solution of the complex time information such as long distance dependencies. But the network cannot necessarily get the precise information which occurred the past. Although the extension of the number of the units in the context layer can enrich the information in the hidden and context layers, this extension might not result in a realistic solution of the long distance dependency. On the other hand, the extension of the number of generations in context layers is a direct way to deal with past n-th generations of contents in the inner state of the network. The n-th generations of context layers can receive exactly the same information of $n-1$-th generations of context layers. The extension of the number of generations in context layers might deal with this problem (multi-center embedded structure) adequately. Since the n-th context layer can receive the contexts of the $n-1$-th context layer directly, the network can deal with the information which occurred long time ago (n-th generation ago).

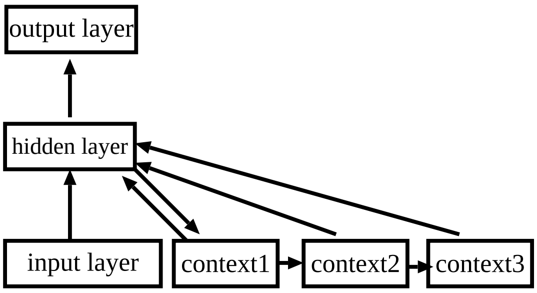


Figure 2: The extension of number of generations in context layers

Numerical experiment

In order to confirm the ability of the network with the extension of the number of generations in context layers, a numerical experiment was performed. The sentences were generated in accordance with the grammar which was almost the same as the one which Elman (1991) used (see Table 1). Total 25 words included EOS (End Of Sentence) consisted of the 25 input and output units.

The normal Elman network which has 20 hidden and context units and the extended Elman network with 5 context layers were compared. The output of the normal Elman network with 20 hidden and context units is for example: *Mary(girl) feeds(who) lives walks. Mary(boy) who(walks). Mary lives. Mary(cats) who(hear)* . The words in the parentheses indicate the correct answers. The number agreement between nouns and verbs preserved in shorter sentences. For example, *Mary(girl) who lives(chases) cat sees.* However, in case of longer sentences which have many relative pronouns, there was a tendency to show incorrect words, which means that the error words did not consist with the parts of speeches as the correct words. On the other hand, the extended Elman network with the 5 generations context layers could deal with the long distance dependency, for example *Mary(boy) who hears cat(Mary) sees Mary.*

Conclusion

Although there was no significant difference in the sense of the quantity in total performance between the normal Elman network and the extended Elman network, there was a quality difference in errors in the sentences with many relative pronouns. The extended Elman network with multi-context layers could process sentences with many relative pronouns properly in the meaning that it could deal with long distance dependencies with multi-center embeddings. This might mean a potential ability of the extension in the number of generations in context layers.

Table 1: The grammar used in the experiment.

S	→	NP VP "."
NP	→	PropN | N | N RC
VP	→	V (NP)
RC	→	who V | who V NP | who V (NP)
N	→	boy | girl | cat | dog | boys | girls | cats | dogs
PropN	→	Mary | John
V	→	chase | feed | walk | live chases | feeds | walks | lives see | hear | sees | hears

Additional restrictions:
1. number agreement between N & V within clause, and (where appropriate) between head N & subordinate V
2. verb arguments:
chase, feed → require a direct object
see, hear → optionally allow a direct object
walk, live → preclude a direct object
(observed also for head/verb relations in relative clauses)

In formal language theory, it is well-known that context free grammar can be processed by finite state automata, therefore, there exists a parsing algorithm. On the other hand, context dependent grammar is in general undecidable. Therefore, it is required to develop a parsing algorithm to deal with context dependent grammar. The extended Elman network with multi-context layers proposed hear could be one of the possible candidates to deal with such complex problems. This model can be regarded as a natural extension of the simple recurrent neural network with multi-memory storage. It could also be analogous with a human model of language information processing.

References

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 195–225.

Giles, C. L., Miller, C. B., Chen, D., Chen, H. H., Sun, G. Z., & Lee, Y. C. (1992). Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, *4*, 393–405.

Perceptual Control Theory Model of the "Beads in the jar" task

Tristan Robert Browne (tristan.browne@postgrad.manchester.ac.uk)
Faculty of Life Sciences, University of Manchester, Oxford Road, Manchester, M13 9PL, UK

Warren Mansell (warren.mansell@manchester.ac.uk)
School of Psychological Sciences, University of Manchester, M13 9PL, UK

Wael El-Deredy (wael.el-deredy@manchester.ac.uk)
School of Psychological Sciences, University of Manchester, M13 9PL, UK

Keywords: PCT; DTD; Hierarchical control.

Introduction

Perceptual Control Theory (hereafter PCT) has been successfully employed in modelling skilled performance (Marken, 2001) and prescribing errors (Marken, 2003). Here we model the draws-to-decision (DTD) behaviour of participants on the "beads-in-the-jar" task (see Fine *et al.*, 2007).

PCT is a control theory approach to explaining human behaviour, derived from negative-feedback loops used in engineering and developed for the application to Psychology since the latter half of the last century (Powers, 1973). The theory states that all behaviour is purposeful and is intended to control specific environmental variables. A system of hierarchical control directs behaviour through interconnected control systems at multiple levels. Higher level systems set reference values for immediately subservient systems and these systems also feedback information regarding their current state. First order systems act on, perceive and feedback the state of the controlled variable to the system hierarchy.

"Beads in the jar task"

Participants were told there were two jars, (*jar R* 60:40 red to green and *jar G* 60:40 green to red beads) and that up to 20 beads would be drawn randomly from one of the jars, with a 50% chance of either jar being chosen. The task required subjects to choose after the first draw and on every subsequent draw either which jar the beads were coming from or to draw another bead. They were instructed only to decide when they were sure which jar the beads were coming from. The number of draws participants chose before deciding was the draws-to-decision (DTD) measure.

Method

Behavioural data was collected from 39 participants in the "beads-in-the-jar" task under three conditions: *High Cost Condition* (*HCC*) where participants could win £4 by deciding the correct jar on the first draw, and then lost 20p for every subsequent draw; *Low Cost Condition* (*LCC*) initial winnings £2 on the first draw and then 10p lost for every draw; and the *No Cost Condition* (*NCC*) where no winnings or drawing costs were applied.

Participants' mean DTD was significantly lower in the *HCC* than in the two other conditions, and significantly lower in the *LCC* than the *NCC* (figure 1).

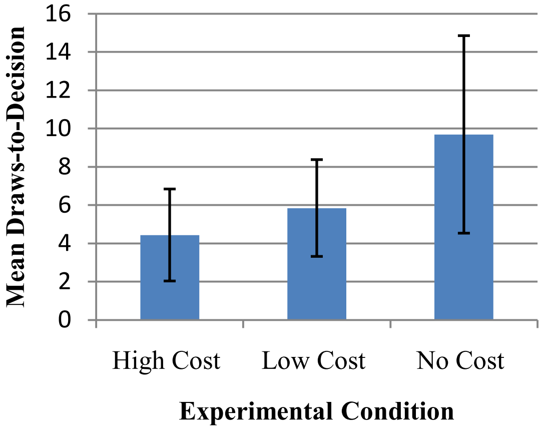


Figure 1: Mean conditional DTDs and associated standard error. Significant differences found between all conditions.

Model

Our PCT model of the DTD behaviour employed two competing control systems at the same level: 1) participants were controlling for how much drawing was costing, 2) participants perceived how sure they were of which jar the beads were being drawn from. This fed into a comparator that outputted a decision when they were surer of it being *jar R* or *jar G* than how much they perceived it cost them to draw another bead. We modelled these using winnings versus the perceived likelihood of the jars (exp. 1) and perceived total cost versus jar uncertainties (exp. 2).

Results

Experiment.1

We accounted for all possible DTD results in the *HCC* and the 18/20 *LCC* using a perceived likelihood measure based on red and green bead counts and optimising the gain on the winnings only using a deterministic linear optimisation for each DTD value (equations.1-3).

$$Red\ Count = No.Red\ Beads - No.Green\ Beads \tag{1}$$

$$Green\ Count = No.Green\ Beads - No.Red\ Beads \tag{2}$$

$$\begin{aligned} &Perceived\ Winnings \\ &= \{Total\ Winnings\ [f(condition)] \\ &- Cost\ per\ Draw\ [f(condition)]\} * Gain \end{aligned} \tag{3}$$

Using 1000 random randomly generated bead sequences 60:40 in favour of red, the model was tested for robustness across all 20 DTD scores in the *HCC* and *LCC*. The mean squared error for DTD values was calculated based on the error for each novel sequence on each DTD draw (figure 2).

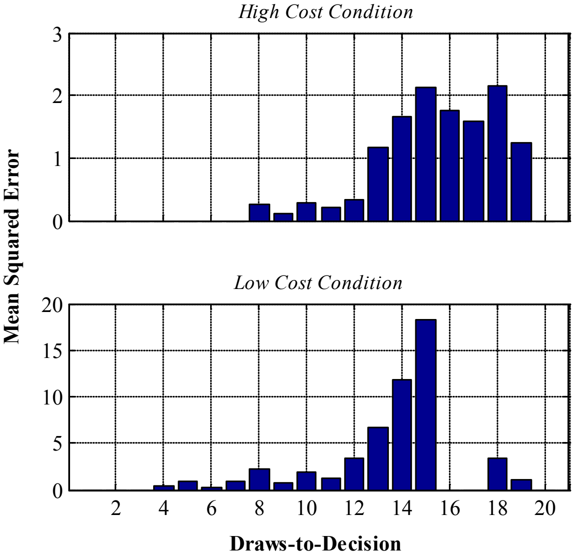


Figure 2: MSE for each DTD using 1000 random bead sequences.

Experiment.2

Here we aimed to model all participants' results in all three conditions. We optimised Gain Factors through iterations of the *HCC* and the *LCC* simultaneously and finally across all three conditions. We calculated jar "uncertainty" using a maximum bead count of 20 for each jar and taking one away each draw depending on the bead colour. We also optimised an "internal cost" value across the conditions simultaneously, for the perceived cost (equations 4-6).

$$\begin{aligned} Red\ Count = Gain &\\ &* (Max\ Red\ Beads \\ &- No.Red\ Beads) \end{aligned} \tag{1}$$

$$\begin{aligned} Green\ Count = Gain * (&Max\ Green\ Beads \\ &- No.Green\ Beads \end{aligned} \tag{2}$$

$$\begin{aligned} &Perceived\ Cost = \\ &(Cost\ per\ Draw * No.of\ Draws\ ) +\ Internal\ Cost \end{aligned} \tag{3}$$

The model produced: exact expected values for 12/39 participants, an error of ±1 DTD for 32/39 participants and accounted for all participants with an error margin of ±2 DTD for the *HCC* and *LCC*. When applied to all three conditions the model perfectly accounted for 2/39 participants and 35/39 with an error of ±6 DTD.

Discussion

The model was more successful in the *HCC* than the *LCC* in experiment 1, both in terms of modelling more DTD results and a lower MSE for each. This could be due to the applicability of the model to these different situations. When the winnings for a correct answer and the cost per draw are higher, participants' behaviour will be more influenced by these factors and less by other factors such as boredom with the task causing them to draw early. We therefore hypothesise that in future experiments if the initial winnings and costs per draw were even higher, then this model would be a better predictor of participants' performance.

This argument is also partially supported by the results from experiment 2: using the same gains and internal costs across conditions, the model was most successful when there was a cost for a draw: in the *HCC* and *LCC*. However when the *NCC* was introduced much larger errors resulted.

It would be unrealistic to suggest that all 39 participants were using the same mental model to compute which draw they would make a decision. However, the reliability of our model in the two cost conditions suggests that cost for drawing is an important factor for determining the DTD.

The reliable effect for delusional subjects to "jump-to-conclusions" (Fine *et al.*, 2007) may then be, to some degree, due to a higher internal cost for making extra draws in the task for these participants. This cost could be anxiety to finish the task early or an over-compensation to want to appear intelligent to the experimenter. Future studies asking participants post-experiment which factors caused them to make their decision will help to clarify this hypothesis.

The partial success of the PCT framework for this task implies its viability for modelling reasoning behaviour. Future work may consider how higher level systems interact with lower systems, and how these higher levels could serve as a regulator for which mental model participants preferentially employ for probability estimation, cost perception and other reasoning processes.

Acknowledgments

I fully acknowledge and gratefully thank the Medical Research Council for supporting me throughout this project.

References

Marken, R. (2001). Controlled variables: Psychology as the centre fielder views it. *American Journal of Psychology, 114*, 259-281.

Marken, R. (2003). Error in skilled performance: A control model of prescribing. *Ergonomics, 46*, 1200-1214.

Powers, W. (1973). *Behavior: The control of perception*. Chicago: Aldine.

Fine, C., Gardner, M., Craigie, J. and Gold (2007). Hopping, skipping or jumping to conclusions? Clarifying the role of the JTC bias in delusions. *Cognitive Neuropsychiatry, 12*, 46-77.

Mathematical Modeling of Human Brain Behavior as an Adaptive Complex System

Maryam Esmaeili (maryam.esmailee@gmail.com)
Department of Informatics, Lugano, Switzerland

Abstract

The aim of this paper is modeling of brain as a multi-agent system and then theoretical study of game-theoretic solution concepts in competitive and cooperative multi-agent interactions in this system. Brain as a cognitive function implementer is composed of large-scale neural networks of cognition (neurocognitive networks) which are considered as expert agents that do what they think in their on best expertness. Neurocognitive networks implement the cognitive functions in brain and thorough understanding of cognition is not possible without knowledge of how they operate individually and socially. In this study dynamic interaction among those expert agents are formulated as competitive and cooperative behaviors. We obtain the equilibrium behavior in the long run, and characterize the collective behavior of these expert agents as responsible of intricacies of cognition. By this work, it was shown how complex collective behavior of brain can emerge from the locally optimal behavior of each agent. In the end we will see how these neural networks organize themselves in a way that the collective behavior will be intelligent. It will be shown that the best structure in brain for having intelligent behavior is multilevel hierarchical organization with nesting structures.

Keywords: multi-agent system, cooperative behavior, competitive behavior, self-organization, neurocognitive network

Introduction

The gap between knowledge of the brain and of the mind can only be bridged with understanding of neural system's behavior that performs cognitive operations. Neurocognitive networks are large-scale systems of distributed and interconnected neuronal populations in the central nervous system organized to perform cognitive functions. We consider neurocognitive networks to be flexibly adaptive to the rapidly changing computational demands of cognitive Processing [1,2,3,4]. The large-scale anatomical connectivity of the cerebral cortex provides a richly intricate structure within which the constituent local area networks have an enormous potential for coordination in a multitude of different patterns. The theory of coordination dynamics [5,6] provides insight into the dynamic characteristics of such interacting complex neural systems.

The ambition of this work is modeling cognitive development through studying the competitive and cooperative behavior in interacted agents (neurocognitive network) in brain. In this study we try to understand how to set up the architecture of an agent as a component of a complex system to be suitable for evolution, how self-interested behavior in every agent evolves to cooperative behavior, and how the goal structure of each agent can be self-modified in order to achieve the common goal of the system. In this work we represent a recursive definition of agent in this way that neurocognitive networks as a particular autonomous entity is considered as agents and also brain as a whole system is considered as an agent. In this way, we put things together and call them an agent and they have a recursive structure. In a hierarchical organization, an agent could be made up of a number of other agents with many different levels. The recursive organization would allow us to build a complex adaptive system like brain at different levels of granularity. In the end we will discuss why and how neurocognitive networks as self-interested agents form their organization.

Modeling of Competitive and Cooperative Behavior of Agents

The notion of self-interested behavior and self-motivated is the foundation of many fields of research. Agents are self- motivated in the sense that they only do the tasks, which are expert in, and are in their own best interest, as determined by their own goals and motivation. Each expert agent has its own expertness or goal, which is expressed in term of a function. In this work, it was supposed that the goal of every agent (cognitive networks) is improving learning process.

The learning progress function of each neurocognitive network as an agent in brain depends on its prediction, state and all the other agent's states. At each time period, each agent faces the problem of choosing strategy and anticipating next state in order to maximize its own learning progress function. To fulfill its long-term interest or expert, every agent seeks a sequence of strategies, which maximizes the accumulated learning progress function defined over an infinite time horizon.

The functions of the neurocognitive network are expressed in real time by the coordinated actions of cooperating areas, with the states of coordination changing dynamically [6]. So cooperation among these agents can be a very important factor for analyzing the behavior of this complex adaptive system. The key element that distinguishes a common goal from an agent's individual goal is that it requires cooperation. By a common goal, we also mean one, which is not achievable by any single agent alone, but is achievable by a group of agents. The self-interested behavior of each agent must be coordinated to achieve globally consistent and efficient collective actions. In this work, we define such a common object as the summation of the strategies of the individual agents in a society.

Intelligent as an Emergent Behavior

The most important point here is that how can extract intelligence by deriving the implicit cooperative behavior of each self-interested and non-intelligent agent. We described the competitive interactions among agents as the basis for cooperative interactions learnable through imitation. In our model every agent (neurocognitve network) makes decisions on the basis of imperfect information about other agents' activities. Now we are going to consider how the evolution of cooperation proceeds?

We need to understand how the competitive behavior of each agent evolves to implicit cooperative behavior. Implicit cooperative behavior of each agent is defined in terms of the effect on other agents. At each time the expert based on the current state of corresponding expert and the other expert make one prediction. Every agents try do make an action that can minimize the prediction error which makes its competitive behavior. The cooperative behavior of agents is modeled as the set of strategies optimizing the summation of the action functions of all agents.

Regarding the collective behavior emerging from competitive interaction, we have the following interesting observation. If the number of agents is small, the summation of the learning progress increases as the number of the agents increases, after a certain number of agents is reached it decreases if the number of agents increases, and it converges to zero as the number of agents becomes very large. It means that special number of neurocognitive networks can learn a cognitive function and the objective interaction between these agents is limited to the number of agents. We can approximate the number of the neurocognitive networks, which can have interaction to learn a specific skill. It is possible to approximate the number of the agents, which can learn a specific skill. By having this number and making a group based on this number and considering the sum of progress in learning of every agent and the sum of learning of all agents in that group and subtracting them the amount we achieve represents the effect of cooperative between the agents of this group. The emergent Intelligence is based on this cooperation between the neurocognitive networks that are not able to produce intelligent behavior alone.

The Self-Organization Process

In this section, we investigate why and how neurocognitive networks form their organization and produce especial structures in brain and cortex. The first question is that why every high level cognitive function is done by a special part of brain with a specific shape and structure? The answer is that neurioocognitive networks do cognitive functions and as was shown before for having intelligent behavior they need to make organization and have cooperation with each other. They may form an organization because of their joint interest in efficient resource acquisition or allocation [16]. We show that their organization can emerge through competitive interactions motivated by self-interested agents. We consider two types of organization, the flat organization, and the hierarchical organization. The collective learning progress at competitive equilibrium and cooperative equilibrium in these two different organizations is computed and the results are compared. By comparing these results we consider that each agent receives a higher utility by forming a hierarchical organization. They may form a hierarchical organization because of their joint interest in efficient resource allocation, and the self-interested agents benefit from a hierarchical organization with a nesting structure where they can improve their own objects.

Conclusions

We have argued here that the neural underpinning of cognition is best understood through the study of neurocognitive networks. We tried to model the behavior of these neural networks by some classic rules in social science and game theory. When examined from this perspective, cognition is seen as a dynamic process that rapidly evolves through a series of informational consistent coordination states. In each moment of cognitive processing, there are two types of behavior that cause transition from one cognitive state to another. These two types of behavior are two common behaviors in social science and society: Competitive behavior based on self-benefit and interest and cooperative behavior.

We understood that simple local interactions between neurocognitive networks could produce complex and purposive global behavior as a cognitive skill. We formulated and analyzed the competitive and cooperative behaviors of these self-interested agents in a dynamic environment. We described a way of organizing the set of multiple agents into a structured organization. Based on this model we can say that every neurocognitive network has a simple goal that in this model was progressing of learning that we modeled it by a linear and simple activation function. By this local goal the agents try to have interaction and the cooperation behavior emerge by these local simple interactions. We showed that with a hierarchical structure the behavior of organization can be more intelligent and so neurocognitive networks should organize themselves in hierarchical structures to produce more intelligent behavior. In a hierarchical organization, an agent could be made up of a number of other agents with many levels. We can conclude that the growth of neural system starts from the set of the unstructured flat organization of neurocognitive networks that we considered them as some self-interested agents.

These self-interested agents are left to self organize themselves into the whole organization to have more and more progress in learning.

References

[1] Mesulam, M.-M., 1990. Large-scale neurocognitive networks and distributed processing for attention, language and memory. Ann. Neurol. 28, 597–613.

[2] Mesulam, M.-M., 1998. From sensation to cognition. Brain 121, 1013–1052.

[3] Bressler, S.L., 2003a. Cortical coordination dynamics and the disorganization syndrome in schizophrenia. Neuropsycho-pharmacology 28, S35–S39.

[4] Bressler SL, 2006. Tognoli, E. Operational principles of neurocognitive networks. International Journal of Psychophysiology 60:139-148

[5] Kelso, J.A.S., 1995. Dynamic Patterns: the Self-Organization of Brain and Behavior. MIT Press, Cambridge, MA.

[6] Bressler, S.L., Kelso, J.A.S., 2001. Cortical coordination dynamics and cognition. Trends Cogn. Sci. 5, 26–36.

[7] Grossberg, S., 2000. The complementary brain: unifying brain dynamics and modularity. Trends Cogn. Sci. 4, 233–246.

[8] Haken, H., 1977. Synergetics—an Introduction: Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry and Biology. Springer, Berlin.

[9] Friston, K.J., 2000. The labile brain. III. Transients and spatio–temporal receptive fields. Philos. Trans. R. Soc. Lond., B Biol. Sci. 355, 253–265.

[10] Friston, K.J., 1997. Imaging cognitive anatomy. Trends Cogn. Sci. 1, 21–27.

[11] Fingelkurts, An.A., Fingelkurts, Al.A., 2004. Making complexity simpler: multivariability and metastability in the brain. Int. J. Neurosci. 114, 843–862.

[12] Basar T, Olsder G. 1982, Dynamic noncooperative game theory. Academic Press, New York.

[13] Namatame A, Sasaki T. 1998, Self-organization of complex adaptive systems as a society of rational agents, ISAROB, 189-195

[14] Fundenberg D, Tirole J, 1991, Game theory. MIT Press, Cambridge

[15] Marschak J, Randner R., 1972. Decentralized theory of teams. Yale University Press, New Haven

[16] Carley K, Prietula M, 1994. Computational organization theory. Lawrence Erlbaum, London

Reimplementing a Diagrammatic Reasoning Model in Herbal

Maik B. Friedrich (maik.friedrich@dlr.de)
German Aerospace Center, Braunschweig

Frank E. Ritter (frank.ritter@psu.edu)
College of IST, Penn State

Keywords: Herbal, Diag, Diag-H, reimplementing, cognitive model

Introduction

This paper builds upon a study of how people find faults in a simple device and a corresponding cognitive model (Ritter & Bibby, 2008). This existing model, Diag, was implemented in Soar 6 and is based on the idea that learning consists of procedural, declarative, and episodic learning. Diag was developed to analyze human behavior while solving a simple diagrammatic problem (Ritter & Bibby, 2008), a task with similarities to many important real world problem solving tasks. Because Diag predicted astonishing results and is implemented in a version of Soar that is no longer supported, an implementation in an up-to-date cognitive architecture is necessary to make the model available again and more flexible to future changes.

We maintained Diag's basic structure while reimplementing it in a high-level behavior representation language, Herbal, that generates Soar models and can generate different variants more quickly that in Soar directly. Herbal compiles into Soar 9, which allows not only that the model can be used again for further research with current Soar models but it is also made accessible to more researchers. This newly implemented model, called Diag-H, was validated by comparing its predictions to the existing data. It could be shown that Diag-H creates almost the same results as Diag but also incorporates the advantages of Herbal.

Diag task and results

The Diag task is called fault-finding task (FFT) and builds upon an interface with 4 switches and 7 light that represent an electrical circuit with 7 different components that are connected via switches. The task consists of a combination from interface information and circuit condition to determine which component is faulty.

Diag was implemented with the effort to predict human reaction times and learning behavior while solving the Diag task. The models strategy is based on the energy flow running through the circuit. A light gets selected based on its position in the circuit and tested by the position of the switches and if its lit up or not. On the Problem Space Computational Model (PSCM) (Lehman, Laird, & Rosenbloom, 1996; Newell, Yost, Laird, Rosenbloom, & Altmann, 1991) level, Diag consists of problem spaces that are hierarchically ordered to solve the FFT by testing the components stepwise.

For validating the Diag model, a user study with 10 participants was run. The participants were instructed how the circuit components are connected, how the components are represented on the interface, and what their task is. While solving the FFT the participants had to recall the circuit diagram from memory, combine it with the presented interface constellation, and identify the faulty component. The results showed that the average proportion of variability in problem-solving time per participant was 79%. The task, the study, and the results are described in detail in Ritter and Bibby (2008).

Diag-H

The reimplementation of Diag was done in Herbal (Haynes, Cohen, & Ritter, 2009), a high-level language based on the PSCM that produces models that can run in Soar and Jess. Because of the use of Herbal the reimplementation required an understanding of the PSCM and visual modeling techniques. This serves as an example of how Herbal can provide modelers that have no strong programming background access to the complicated machinery used by cognitive architectures that may traditionally be out of their reach.

Because Diag-H is a reimplementation of the Diag structure, the most important effort was to copy the structure accurately. Diag-H uses the same structure of problem spaces and strategy to solve the FFT. The reimplementation process was supported by Herbal because of the direct implementation of the PSCM. This means Herbal models implement problem spaces directly and assign them hierarchically.

The task knowledge in Diag-H is stored in operators. An operator in Herbal is a combination of generic conditions and actions that can be combined as required. 93 conditions and 56 actions were modeled and combined to 85 operators. Herbal compiles Diag-H into 187 Soar rules.

Diag-H predictions and the existing data

To validate Diag-H, we used the data from Ritter and Bibby (2008). The number of Soar model cycles with learning turned on was used to predict solution times from Diag-H. Using linear regression between Diag-H predictions and the existing user data, an average motor output time (B = 1.42 s) and an average time as slope of decision cycles (0.187 ms) was calculated. To determine how accurate the model predicts individual behavior, the predicted times (as slope of decision cycles * decision cycles + intercept = 0.187 ms * decision cycles + 1.42s) were compared to the observed problem solving times.

Each participant saw a different order of the 20 faults. Figure 1 shows the individual problem-solving time for participant 8 and the predicted times aggregated over this stimulus predicted by Diag-H. This example shows how well the Diag-H predictions fit to the user data.

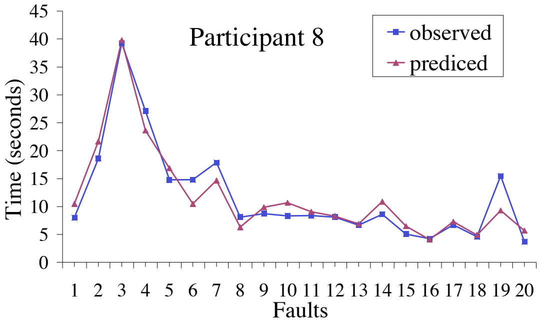


Figure 1: The observed and predicted problem-solving times over 20 trials for participant 8.

To compare the Diag-H predictions further to the user data, each set of model cycle per run was regressed to the problem-solving times for each participant individually. The average proportion of variability in problem-solving time per participant accounted by Diag-H was $r^2 = 72.2\%$. By removing two non significant participants from the analysis the significance reaches $r^2 = 87\%$.

These comparisons showed that Diag-H was able to predict the existing participant performance to a good extent. Similar to Diag, Diag-H also has problems in predicting the performance of participants P5 and P7. However, when comparing the correlations for the predictions per fault, per trial, and per participant Diag-H is constantly 5% less accurate than Diag.

Summary

We have described the use of a high level behavior representation language, Herbal, to reimplement Diag, a model that solves a diagrammatic reasoning task. The reimplementation, Diag-H, was validated by testing whether it creates the same predictions as Diag. Diag-H uses the same strategy and reaches almost the same results by predicting human behavior and combines this with Herbal advantages. A Herbal model can predict similar results to a Soar model but has a shorter implementation time. The generic Herbal structure allows quick adaptations to future requirements and further development of models. These results allow proceeding with research on the Diag task supported by the Diag-H model.

Diag-H offers several new possibilities for research. One aspect is implicated by two participants (P5 & P7) that did not fit either the existing Diag predictions or the updated Diag-H predictions. Because these participants' error rates were not significantly higher than the average, the results suggest that they used a different strategy than Diag-H. Therefore, the development of several strategies will be necessary for a detailed analysis of the performance of these two participants. Through the use of Herbal as implementation language the process of creating new strategies will be simplified. In the future even Herbal compiled ACT-R models will be available (Paik, Kim, & Ritter, 2009).

Acknowledgements

DLR and ONR provided resources for this work and ONR supported the development of Herbal.

References

Bibby, P. A., & Payne, S. J. (1993). Internalisation and the use specificity of device knowledge. *Human-Computer Interaction, 8*, 25-56.

Bibby, P. A., & Payne, S. J. (1996). Instruction and practice in learning to use a device. *Cognitive Science, 20*(4), 539-578.

Haynes, S. R., Cohen, M. A., & Ritter, F. E. (2009). A design for explaining intelligent agents. *International Journal of Human-Computer Studies, 67*(1), 99-110.

Lehman, Laird, & Rosenbloom. (1996). A gentle introduction to Soar, an architecture for human cognition. In S. Sternberg & D. Scarborough (Eds.), *Invitation to cognitive science* (Vol. 4). Cambridge, MA: MIT Press.

Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S., & Altmann, E. (1991). Formulating the problem space computational model. In R. F. Rashid (Ed.), *Carnegie Mellon Computer Science: A 25-Year commemorative* (pp. 255-293). Reading, MA: ACM-Press (Addison-Wesley).

Ritter, F. E., & Bibby, P. A. (2008). Modeling how, when, and what learning happens in a diagrammatic reasoning task. *Cognitive Science*.

The Need of an Interdisciplinary Approach based on Computational Modelling in the Study of Categorization.

Francesco Gagliardi (francesco.gagliardi@libero.it)
Department of Philosophical and Epistemological Studies
University of Rome *La Sapienza* — Via Carlo Fea — I-00161 Roma, Italy

Categorization in Cognitive Psychology and the Prototypes-Exemplars Debate

The main theories (Murphy, 2002) concerning the study of categorization and the nature of concepts are: the classical theory also known as Aristotelian, the prototypes theory, the exemplars theory and theory-theory. The theories of prototypes and exemplars, jointly taken, constitute the so called *typicality view* on concepts. In fact, both theories, even if in contrast, are based on experimental evidences that as a whole they show the existence of a "phenomenon" of typicality in categorization processes (see *"Typicality as phenomenon"* in Murphy; 2002 pg. 28). Prototypes and exemplars theories supersede the limitations and the experimental inadequacy of the classical theory, based on logical predicates, but when considered separately turn out to be incomplete and unsatisfactory (Murphy, 2002; pg. 4). Nevertheless, in the past thirty years some literature concerning experimental psychology focused on the comparison between prototypes theory and exemplar theory and carrying out experiments in order to demonstrate the correctness of one theory or the other one. For example we can consider the following two papers in conflict (Minda, Smith, 2002) and (Zaki et al., 2003). In the former is supported prototypes theory, while in the latter the exemplars theory, even if they make use of the same data set. As matter of fact, the research line related to the diatribe of prototypes vs. exemplars appears to be a dead end because it is fruitless and not decisive and also because it is based on the *naïve* epistemology of pursuing a so called *experimentum crucis*.

Theories of Categorization and Machine Learning

The ultimate aim of the researches about categorization is the understanding of representations of categories (Murphy, 2002; pg. 3) that we build, the concepts, and by which we perform different cognitive tasks. A common aspect of prototypes theory and exemplar theory is the idea that each category is represented by instances belonging to the class: in one case the instances are the prototypes abstracted from observations, and in the other case are the same previously observed instances. In the field of machine learning (Witten, Frank, 2005) (Duda, Hart, Stork, 2000) and automatic classification, one of the learning methodologies known in literature is the so called *instance based learning,* for which the classes, learnt by the classifier system, are represented by instances of the corresponding class. Therefore, the field of machine learning, and in particular of instance-based learning, is the natural context where to study the theories of human categorization based on prototypes or exemplars, from both the theoretical viewpoint of the computational statistics, and the empirical viewpoint of the synthetic method (Cordeschi, 2001), consisting in the realization of classifier systems which embody theories of categorization. Within instance-based learning it is possible to connect the characteristics of robustness and sensibility of a classifier system with categories representation based, respectively, on prototypes or exemplars. In fact, prototypes based classifiers, such as the *Nearest Prototype Classifier* (*NPC*) and the *Nearest Multiple-Prototype Classifier* (*NMPC*), construct the representative instances of the class, called *prototypes*, as the barycentres of an observations subset. These systems obtain robust classifications, that is, not sensitive to noisy and atypical observations. On the other way, classifiers based on exemplars, such as the *Nearest Neighbour Classifier* (*NNC*) and its well known generalization *k-NNC,* use as the set of representative instances the whole set of observations of classes, without any elaboration or abstraction. These systems, which are entirely based on the ability to save all observations in memory, obtain classifications extremely sensible and not at all robust. In the family of instance-based systems the classifiers *NPC* and *NNC* represent the limit cases of maximum robustness and maximum sensibility respectively and they use types of classes representations that can be related to the theories of prototypes and of exemplars, respectively. As it is well known in computational statistics a classifier system, whether natural or artificial, is the result of the trade-off between the two contrasting requisites of robustness and sensibility. More formally this problem is linked with the *Bias-Variance theorem* and with the *Bias-Variance dilemma*, e.g. (Duda, Hart, Stork, 2000, Chap.9). Thus *Prototype-Theory* and *Exemplar-Theory* have not to be considered as two conflicting theories, but they are two limit cases of a same technique to categorize called *Instance-Based Learning*. This technique is used both by natural systems as human minds, and by some artificial systems as instance-based classifier systems. From these simple theoretical considerations it is then clear that it is absolutely groundless to assert the correctness of one of the two theories against the other; a theory which subsumes both of them should be sought just in the trade-off between robustness and sensibility. In fact there are some classifier systems, such as the *Varying Abstraction Model* (Vanpaemel, 2005), the *Mixture Model* (Rosseel, 2002) or the *Prototype-Exemplar Learning Classifier* (Gagliardi, 2008), which are able to subsume both the prototypes and exemplars theories and, hence, they can help to realize a *theory of typicality* which would explain the phenomenon of typicality. In summary, we can affirm that when framing the problem of categorization in the field of machine learning, the prototypes-exemplars diatribe reveals completely unfounded for the general theoretical considerations about the bias-variance dilemma, and also for experimental evidences due to the existence of some hybrid classifier systems. Therefore, the aforementioned diatribe is ill-posed, because of a poor formalization of the subject and the *naïve* epistemology of *experimentum crucis*. These drawbacks could be superseded, by using results of machine learning and computational statistics, and by embracing the synthetic method, as it would be required by the interdisciplinary nature of the categorization problem.

Machine Learning and Cognitive Plausibility of Representations

Classification algorithms strongly depend on the kind of classes' representation that they infer from data, known as *concepts description* (Witten, Frank, 2005; pg. 42) and that they then use to classify new instances. In fact, in the field of machine learning, one can distinguished different family of classifier systems according to the kind of used representations (e.g. instances, decision trees, logical predicates, support vectors, etc.). As it is known in cognitive psychology, the instance-based representation is the only one that coheres with both the prototypes and exemplars theories and therefore, it is the representation to be used in accordance with the *typicality view*. Instead, the most used type of knowledge representation in the machine learning is the one based on rules or decision trees: *"Induction of decision trees is probably the most extensively researched method of machine learning used in data mining"* (Witten, Frank, 2005; pg. 199), although these kinds of representations lack of a true cognitive plausibility, in fact they can be thought as models of the classical theory of categorization, since they represent concepts as logical predicates. As matter of fact, many researches in machine learning as well as machine learning handbooks completely neglect the connections with cognitive psychology and ignore concepts theories, or they do it, let say, in a "superficial" manner. This attitude is well exemplified by Witten and Frank who affirm, with regard to the different possible categories representations that: *"instances do not really «describe» the patterns in data"* (Witten, Frank, 2005; pg. 79) and with regard to the instances based categories learning that: *"in a sense this violates the notion of «learning»"* (Witten, Frank, 2005; pg. 79). This position, followed till its extreme consequences, leads to the paradoxical idea that humans, since represent categories by instances, do not have real learning abilities and do not really have concepts; conversely these abilities are hold only by machines that represent the classes in a anti-psychological way, as for example, with rules and decisions trees. Machine learning researches underestimate possible theoretical and applicative involvements with cognitive sciences although it seems natural that who studies artificial learning of categories should do it in parallel with, or at least not ignoring, the studies about natural learning of categories.

Concluding Remarks

In the previous sections we put in evidence how the "mono-disciplinary" use of cognitive psychology and machine learning produces disappointing results. In fact, from a hand, cognitive psychology produced thirty years of an unfruitful prototypes-exemplars diatribe, which could be avoided if one had not limited oneself to a superficial use of mathematics for the development of cognitive theories, instead of a more foundational use of it, based on machine learning and synthetic method. On the other hand, the field of machine learning disdains the experimental evidences produced by the psychological research. These errors have to be ascribed to a very disciplined and closed methodological praxis, inside the respective scientific communities, in an almost "corporatist" way. Instead, the problem of categorization, as for many of the problems dealt in cognitive sciences, is the same whether one considers natural systems, as human minds, or artificial systems so an interdisciplinary approach in the study of categorization is the natural setting to conduct researches and it is able to progress both field.

References

Cordeschi, R. (2001), *The Discovery of the Artificial. Behavior, Mind and Machines Before and Beyond Cybernetics*. Kluwer, Dordrecht.

Duda, R., Hart, P., Stork, D. (2000) *Pattern Classification (2nd Ed)*. John Wiley & Sons, New York, NY.

Gagliardi, F. (2008) *A Prototype-Exemplars Hybrid Cognitive Model of "Phenomenon of Typicality" in Categorization.* In Proceedings of the 30th Conference of the Cognitive Science Society. Austin, TX. Pp. 1176–1181.

Minda, J.P., Smith, J.D. (2002) *Comparing prototype-based and exemplar-based accounts of category learning and attentional allocation*. Journal of Experimental Psychology: Learning, Memory, Cognition. 28:275–292.

Murphy, G.L. (2002). *The big book of concepts*. MIT Press, Cambridge, MA.

Rosseel, Y. (2002) *Mixture models of categorization.* Journal of Mathematical Psychology. 46: 178-210.

Vanpaemel, W., Storms, G., Ons, B. (2005) *A Varying Abstraction Model for Categorization*. In Proceeding of the XXVII Conference of the Cognitive Science Society. Pp. 2277-2282.

Witten, I.H., Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. (2nd Ed.)* Kaufmann, San Francisco, CA.

Zaki, S.R., Nosofsky, R.M., Stanton, R.D., Cohen, A.L. (2003) *Prototype and Exemplar Accounts of Category Learning and Attentional Allocation: A Reassessment.* J Exp Psychol Learn Mem Cognit. 29(6), pp. 1160–1173

Unifying syntactic theory and sentence processing difficulty through a connectionist minimalist parser

Sabrina Gerth (gerth@ling.uni-potsdam.de)
Department of Linguistics, Karl-Liebknecht-Str. 24-25
14476 Potsdam, Germany

Peter beim Graben (p.r.beimgraben@reading.ac.uk)
School of Psychology and Clinical Language Sciences, Whiteknights, PO Box 217,
Reading, RG6 6AH, United Kingdom

Keywords: Computational psycholinguistics; Modeling human sentence processing; Minimalist Grammars; Connectionist architecture;

Introduction

Syntactic theory provides a rich array of representational assumptions about linguistic knowledge and processes. Such detailed and independently motivated constraints on grammatical knowledge ought to play a role in sentence comprehension. However most grammar-based explanations of processing difficulty in the literature have attempted to use grammatical representations and processes per se to explain processing difficulty. They did not take into account that the description of higher cognition in the mind encompasses two levels: On the one hand, at the macrolevel, symbolic computation is performed, and on the other hand, at the microlevel, mathematical computation is achieved through processes within a dynamical system. One critical question is therefore how linguistic theory and dynamical systems can be unified to provide an explanation for processing effects. Here, we present such a unification for a particular account to syntactic theory: namely a parser for Stabler's Minimalist Grammars, in the framework of Smolensky's Integrated Connectionist/Symbolic architectures. In simulations we demonstrate that the connectionist minimalist parser produces predictions which mirror empirical findings from psycholinguistic research.

Method

Materials In contrast to English, the word order in German is relatively free, which offers the opportunity to vary syntactic processing difficulties for the same lexical items by changing their morphological case. For this study mild garden-path sentences in German (subject-object vs. object-subject) sentences were used which are known for eliciting a P600 in an event-related brain potential (ERP) experiment (Frisch, Schlesewsky, Saddy, & Alpermann, 2002). Consider the following example sentences in German:

(1) *Der Detektiv* *hat* ***die***
The detective$_{\text{MASC|NOM}}$ has the
Kommissarin *gesehen.*
investigator$_{\text{FEM|ACC}}$ seen.
'The detective has seen the investigator.'

(2) *Die Detektivin* *hat* ***den***
The detective$_{\text{FEM|AMBIG}}$ has the
Kommissar *gesehen.*
investigator$_{\text{MASC|ACC}}$ seen.
'The detective has seen the investigator.'

(3) *Den Detektiv* *hat* ***die***
The detective$_{\text{MASC|ACC}}$ has the
Kommissarin *gesehen.*
investigator$_{\text{FEM|NOM}}$ seen.
'The investigator has seen the detective.'

(4) *Die Detektivin* *hat* ***der***
The detective$_{\text{FEM|AMBIG}}$ has the
Kommissar *gesehen.*
investigator$_{\text{MASC|NOM}}$ seen.
'The investigator has seen the detective.'

The sentences (1)-(2) have subject-object order whereas (3)-(4) have object-subject order. Previous work (Weyerts, Penke, Münte, Heinze, & Clahsen, 2002) has shown, that sentence (3) is harder to process than sentence (1) due to the scrambling operation which has to be applied to the object of sentence (3) and leads to higher processing load. A second effect for these syntactic constructions in German is that (2) and (4) contain a case ambiguous nominal phrase (NP). Bader and Meng (1999) found that readers assume that the first NP is a subject when it is case-ambiguous; Frisch et al. (2002) showed in an event-related brain potentials study that sentences like (4) lead to a mild garden-path effect. This work is able to model both effects - the scrambling operation as well as the disambiguation effect.

Symbolic Representation The symbolic representations of human sentence processing are well-established in the linguistic literature covering a wide range of grammatical formalisms e.g. lexical-functional grammars (LFG), head-driven phrase grammars (HPSG), tree-adjoining grammars (TAG), Minimalist Grammars (MG) and so on. Until now, the present work is the first study which uses the Minimalist Grammars formalism for German, so far it has been only applied to English (Stabler, 1997; Harkema, 2001; Hale, 2003). In order to use MG for a language with relatively free word order, a new pair of features was introduced into the formalism. These scrambling feature expands the movement operation, thereby accounting for the possibility to rearrange arguments of the sentence signaled by morphological case.

Mathematical Representation The second part of this study deals with the encoding of the particular parse steps carried out by the grammar formalism. The minimalist tree of each parse step is mapped onto the fractal tensor product encoding as follows: role vectors represent the positions in the binary minimalist tree (root, left child, right child), while fillers account for the symbols of the tree and the minimalist features of the lexicon entries (e.g. $>$, $<$, +acc, -acc, d, =d etc.). The tensor product (Smolensky & Legendre, 2006) is calculated by the binding of role and filler which results in a tensor product representation of each parse step. In other words each symbolic representation will be presented as a numerical value in an activation space and can be visualized in a coordination system by trajectories. These trajectories visualize the sentence processing difficulties by exploring different areas in the vector space.
Finally the numerical values of the encoding are used as input to a neural network. This study will use Tikhonov-Hebbian learning to simulate the underlying language processes with the help of autoassociators.

Results

Figure 1 shows the trajectories of sentence (1) and (3) which only differ in the scrambling operation for (3). Both graphs start with different initial conditions and converge until parse step 6. At this point the second NP is moved (scrambling) at which the trajectories diverge significantly reflecting the disambiguation process and a high syntactic processing difficulty.

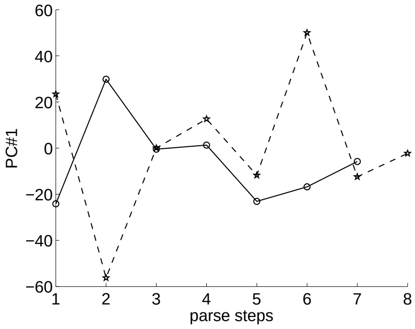


Figure 1: Time series for the scrambling operation.

Figure 2 shows the trajectories for the sentences (2) and (4). The trajectories start with the same initial conditions and proceed equally because both sentences are parsed equally (following the subject preference strategy) until parse step 5. At that point the graphs diverge significantly which can be interpreted as processing difficulties as encountering the second NP (disambiguation). The scramble operation becomes inevitable for sentence (4) and requires a reanalysis of the built syntactic structure. Further the trajectory for sentence (4) breaks down at parse step 7 simulating the garden-path effect.

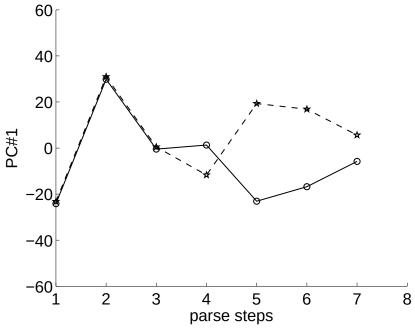


Figure 2: Time series for the garden-path effect.

By modeling these kinds of processing difficulties (Gerth & beim Graben, submitted; beim Graben, Gerth, & Vasishth, 2008) on both levels–macrolevel and microlevel–this approach bridges the gap between the symbolic computation and the mathematical representation and combines the functionalities of established linguistic theories.

References

Bader, M., & Meng, M. (1999). Subjekt-object ambiguities in german embedded clauses: An across-the-board comparison. *Journal of Psycholinguistic Research*, *28*(2), 121-143.

beim Graben, P., Gerth, S., & Vasishth, S. (2008). Towards dynamical system models of language-related brain potentials. *Cognitive Neurodynamics*, *2*(3), 229 - 255.

Frisch, S., Schlesewsky, M., Saddy, D., & Alpermann, A. (2002). The P600 as an indicator of syntactic ambiguity. *Cognition*, *85*, B83-B92.

Gerth, S., & beim Graben, P. (submitted). Unifying syntactic theory and sentence processing difficulty through a connectionist minimalist parser. *Cognitive Neurodynamics*.

Hale, J. T. (2003). *Grammar, uncertainty and sentence processing*. The Johns Hopkins University.

Harkema, H. (2001). *A recognizer for minimalist grammars*. University of California, Los Angeles.

Smolensky, P., & Legendre, G. (2006). *The Harmonic Mind: From neural computation to optimality-theoretic grammar vol. 1: Cognitive architecture* (Vol. 1). MIT Press.

Stabler, E. P. (1997). Derivational Minimalism. *Lecture Notes in Computer Science*, *1328*, 68-95.

Weyerts, H., Penke, M., Münte, T. F., Heinze, H.-J., & Clahsen, H. (2002). Word order in sentence processing: An experimental study of verb placement in german. *Journal of Psycholinguistic Research*, *31*(3), 211-268.

Do the Details Matter?
Comparing Performance Forecasts from Two Computational Theories of Fatigue

Kevin A. Gluck (kevin.gluck@us.af.mil)
Glenn Gunzelmann (glenn.gunzelmann@us.af.mil)
Air Force Research Laboratory, 6030 S. Kent St.
Mesa, AZ, 85212 USA

Michael A. Krusmark (michael.krusmark@mesa.afmc.af.mil)
L-3 Communications, 6030 S. Kent St.
Mesa, AZ, 85212 USA

Keywords: Computational Cognitive Modeling; Fatigue; Alertness; Performance Forecasts; Comparison

A Tale of Two Theories

Activation and Micro-Lapses

We have been developing a computational theory of the effects of fatigue (especially sleep-related fluctuations in alertness) on the human cognitive system, implemented through mechanisms that impact existing components of the ACT-R architecture (Gunzelmann, Gluck, Kershner, Van Dongen, & Dinges, 2007; Gunzelmann, Gross, Gluck, & Dinges, 2009). These mechanisms include the suppression of *activation* in the declarative knowledge system, as well as brief breakdowns in the central production execution cycle, which we call *micro-lapses*.

Through an iterative series of mechanistic architectural modifications, model implementations, and goodness-of-fit evaluations in task contexts like the Psychomotor Vigilance Test (PVT – Dinges & Powell, 1985) and the Walter Reed Serial Addition/Subtraction Task (SAST – Thorne, Genser, Sing, & Hegge, 1985), the theory has evolved to a state in which we have some confidence in its appropriateness. In other words, we feel increasingly confident that the mechanisms we are using to replicate and explain relevant empirical results are both sufficient and necessary for that purpose (Estes, 2002). This gives us a measure of confidence that it is reasonable, perhaps even advisable, to use the theory to make novel performance predictions in task contexts beyond those used for originally developing and evaluating the theory. So far, we have promising results from fatigued performance predictions in both the context of dual-tasking (Gunzelmann, Byrne, Gluck, & Moore, 2009) and also in the context of simulated driving (Gunzelmann, Moore, Salvucci, & Gluck, submitted).

Cognitive Slowing

A popular alternative theory of fatigue is one commonly referred to as *cognitive slowing*. Though typically presented as a verbal-conceptual theory that describes an important category of empirical results from the sleep research community, cognitive slowing has inspired at least one prior computational implementation that explicitly moderated the processing of a simulated cognitive system by literally slowing it down (Jones, Laird, & Neville, 1998). To introduce fatigue effects into their model, Jones et al. modified Soar's mechanisms to introduce artificial delays in processing, thereby having the effect of slowing overall system performance. Indeed, one of our very first conjectures regarding plausible mechanisms for implementing a theory of the effects of fatigue on cognitive processing involved a focus on "cognitive slowing" implemented as changes in the Default Action Time (DAT) of the production cycles in ACT-R, which controls the speed of central cognition in the architecture.

Does It Matter?

Despite what we consider to be convincing theoretical and empirical evidence that a cycle time-based account is less valid than our preferred "activation and micro-lapses" theory, we have been left to wonder whether the different theories would actually produce meaningfully different predictions in a more complex, dynamic, realistic context like aircraft maneuvering. This is more than just idle curiosity. It speaks to the core justification for pursuing basic computational cognitive modeling research – that the details matter – not only in the arena of theoretical constructs, but also in the arena of applied cognitive technologies.

Sleepy Pilot Performance Forecasts

We incorporated the fatigue mechanisms into a cognitive model that flies basic maneuvers with a Predator Synthetic Task Environment, in order to simulate the effects of extended sleep deprivation on pilot performance. Gluck, Ball, and Krusmark (2007) described the basic maneuvering task and cognitive model implementation in detail, and space considerations preclude repeating that material here. We will note, however, that for purposes of the pilot performance forecasts reported here we used Maneuver 7 (which requires simultaneous constant rate of change adjustments to airspeed, altitude, and heading over a 90-second trial) and we used the Control Focus and Performance variant of the pilot model, which is our most valid replication of expert-level performance on the basic maneuvering tasks.

With that model as a baseline, we implemented our set of mechanisms in to the model, and used parameter values derived from previous fatigue modeling efforts using the SAST, to arrive at principled values for the "Activation and Micro-Lapses" account. We also derived predicted DAT values for the "Cognitive Slowing" account using values estimated to account for dual-task performance. Though imperfect, the mechanisms and parameter values reflect an honest effort to faithfully implement and parameterize both accounts. To evaluate the alternatives, we ran the model 110 times at each of four levels of sleep deprivation: Baseline (no sleep deprivation), 1, 2, and 3 days of sleep deprivation. Results are presented in Figure 1.

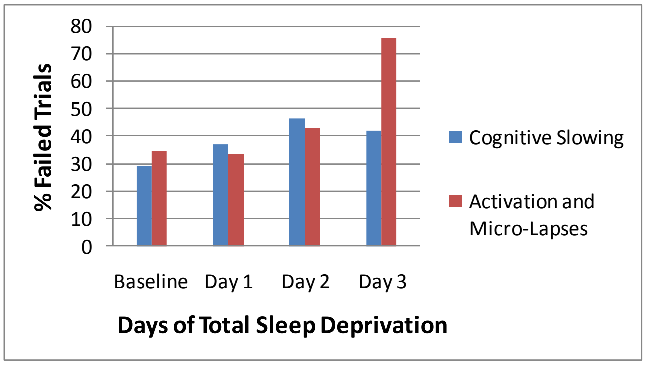


Figure 1: % failed basic maneuvering trials by fatigue theory, across four levels of sleep deprivation

The forecasts show nearly identical performance up to two days of sleep deprivation, followed by a dramatic difference in predicted performance level after three days without sleep. The obvious implication of this result is that it suggests that it *does* matter what the details are in your implementation of a theory of fatigue in the human cognitive system, at least in the extreme. However, this result also raises an assortment of more subtle issues associated with the challenges we face as we begin trying to make real, no kidding, a priori performance predictions in transfer contexts. Some of these questions include:

1. How sensitive are the predictions to variations in the model parameters?
2. How valid are the results?
3. Would we be comfortable using these results to inform policy decisions?

We hope to discuss and debate possible answers to these questions with attendees at ICCM 2009.

Discussion

The good news story is that we have reached a state in our research where we can make forecasts of this sort in complex, dynamic domains and have some confidence in the accuracy of those predictions. This is a desirable state for cognitive science in general, and for us in particular.

The bad news story is that we have no expectation of being able to directly evaluate the accuracy of the model predictions against empirical human data. It is logistically difficult and expensive to run the necessary sleep protocols with this task. It is an interesting conundrum that we are just beginning to face in computational cognitive science.

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The authors thank Kenny Jackson for his assistance with portions of this research. This research was sponsored by the Air Force Research Laboratory's Warfighter Readiness Research Division and by grant 07HE01COR from the Air Force Office of Scientific Research (AFOSR).

References

Estes, W. K. (2002). Traps in the route to models of memory and decision. *Psychonomic Bulletin & Review, 9*(1), 3-25.

Dinges, D. F., & Powell, J. W. (1985). Microcomputer analyses of performance on a portable, simple visual RT task during sustained operations. *Behavior Research Methods, Instruments, & Computers 17(6)*, 652-655.

Gluck, K. A., Ball, J. T., & Krusmark, M. A. (2007). Cognitive control in a computational model of the Predator pilot. In W. Gray (Ed.), *Integrated models of cognitive systems* (pp. 13-28). New York, NY: Oxford University Press.

Gunzelmann, G., Byrne, M. D., Gluck, K. A., & Moore, L. R. (2009). Implications of computational cognitive modeling for understanding fatigue. *Human Factors*.

Gunzelmann, G., Gluck, K. A., Kershner, J., Van Dongen, H. P. A., & Dinges, D. F. (2007). Understanding decrements in knowledge access resulting from increased fatigue. In D. S. McNamara and J. G. Trafton (Eds.), *Proceedings of the Twenty-Ninth Annual Meeting of the Cognitive Science Society* (pp. 329-334). Austin, TX: Cognitive Science Society.

Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*.

Gunzelmann, G., Moore, L. R., Salvucci, D. D., & Gluck, K. A. (submitted). Fluctuations in alertness and sustained attention: Predicting driver performance. Under consideration for presentation at the *9th International Conference on Cognitive Modeling*.

Jones, R., Laird, J., & Neville, K. (1998). Modeling pilot fatigue with a synthetic behavior model. *Proceedings of the Seventh Conference on Computer Generated Forces and Behavioral Representation*, Orlando, FL.

Thorne, D. R., Genser, S. G., Sing, G. C., & Hegge, F. W. (1985). The Walter Reed performance assessment battery. *Neurobehavioral Toxicology and Teratology, 7*, 415-418.

MindModeling@Home
. . . and Anywhere Else You Have Idle Processors

Jack Harris (jack.harris@us.af.mil)
Kevin A. Gluck (kevin.gluck@us.af.mil)
Air Force Research Laboratory, 6030 S. Kent St.
Mesa, AZ, 85212 USA

Thomas Mielke (thomas.mielke@mesa.afmc.af.mil)
L-3 Communications at AFRL, 6030 S. Kent St.
Mesa, AZ, 85212 USA

L. Richard Moore, Jr. (larry.moore@mesa.afmc.af.mil)
Lockheed Martin at AFRL, 6030 S. Kent St.
Mesa, AZ, 85212 USA

Keywords: MindModeling; Computational Cognitive Process Modeling; Volunteer Computing; Distributed Computing

Introduction

As we, the ICCM community, continue to expand the scope of our cognitive modeling ambitions, we increasingly face computational requirements that are an impediment to progress. Computational complexity grows quickly with increases in the granularity of models, the fidelity of the models' operating environment, and the time scales across which these models interact. Additional processing demands are encountered when studying the breadth of a cognitive model's performance capabilities such as through observing the model's sustained fitness while varying the environment or conducting sensitivity analyses of interactions between internal model parameters in a controlled experiments. Such computational demands are not unique to the cognitive modeling community. Other scientific fields (bioinformatics, meteorology, physics, etc.) have already pioneered a variety of platforms and methodologies for dealing with similarly computationally complex problems. We will achieve faster progress toward the broader scientific objectives of cognitive modeling and the specific goals of particular research projects if we pay attention to the lessons learned and capabilities developed in other computational sciences.

Volunteer Computing

An exciting methodological development of the past decade has been the rise of volunteer computing as a means of acquiring access to large numbers of computer processors distributed across the internet. Volunteer computing represents a huge and increasingly powerful computational resource due to the continuous growth rate of end-user processing capability around the world. The first volunteer computing project was SETI@Home. It was established in 1999 for the purpose of demonstrating the utility of "distributed grid computing" by providing a mechanism for analysis of massive amounts of observational radio telescope data. The scientific Search for Extra-Terrestrial Intelligence rapidly caught and held the public imagination, and SETI@Home remains the longest running and one of the most popular volunteer computing projects in the world. This actually is an impressive feat, given that the volunteer computing concept has caught on in an assortment of other scientific communities and there are now approximately three dozen volunteer computing projects available to those interested in donating their idle processor time to scientific pursuits. Most of them, including SETI@Home, run on a software architecture called the Berkeley Open Infrastructure for Network Computing (BOINC). Some of the other large BOINC-based scientific volunteer computing projects include: Climateprediction.net (sensitivity analyses of models that predict Earth's climate up to 2080), MilkyWay@Home (investigating optimization methods for Internet-based computing and developing 3-dimensional models of the Milky Way galaxy), and Einstein@Home (searching for pulsars in data from the LIGO gravitational wave detector). In total, as of April 14, 2009 (the submission date for ICCM 2009), BOINC-based volunteer computing projects include 291,956 active volunteers, offering 531,174 computer hosts. That level of volunteerism is producing an average computational throughput, across all projects, of 1,368 TeraFLOPS.

The largest existing volunteer computing project does not run on the BOINC platform. It is called Folding@Home and is dedicated to understanding protein folding. Folding@Home currently, as of the submission date, has more than 4 million volunteered computers and is producing a throughput of 4,782 TeraFLOPS. By comparison, the world's current fastest centrally-managed High Performance Computing system, at the United States Department of Energy's Oak Ridge National Laboratory, has a peak processing capacity of 1,640 TeraFLOPS, so clearly there is a lot of computational power and potential scientific return available through distributed, volunteer grid computing. It will benefit computational cognitive scientists to begin taking advantage of this platform. Happily, there are now two cognitive science-related volunteer computing projects. One, called Artificial Intelligence System, is an AI project hoping to achieve large scale artificial intelligence by reverse engineering the brain. The other is MindModeling@Home.

MindModeling@Home

Launched in March of 2007, and still (intentionally) in Beta status, MindModeling@Home focuses on utilizing computational cognitive process modeling to better understand the human mind and to improve on the scientific foundations that explain the mechanisms and processes that enable and moderate human performance and learning. It accomplishes this goal using the BOINC software augmented with in-house development of web-based user interfaces and community portals. Together these tools attempt to bridge the gap between the complex engineering challenges of large scale computing and the domain specific requirements of the cognitive science research community.

Since its inauguration two years ago, the system has completed over 50 jobs — most of which involve millions of model runs — which substantially contributed to various research efforts both within and external to our organization. Most of these jobs were completed exclusively by volunteers donating computing time from over 3000 machines (typically 200 to 900 at any given moment in time). However, MindModeling@Home is not limited to volunteers, as we have also achieved integration with local resources as well as several high performance computer clusters.

There are many lessons learned and remaining challenges associated with using heterogeneous computational resources beyond those faced when attempting to use homogenous computing clusters. Some challenges include how to schedule work with virtually no consistent expectation of availability, how to gauge progress and report status to customers, how to ensure that models are written properly and behave appropriately on Linux, Windows and Mac OS X, how to recover when those resources fail, and how to ensure the level of data integrity required for scientific publications. These are non-trivial engineering efforts, and it behooves the computational cognitive science community to leverage existing work in this space.

MindModeling@Home currently supports Common Lisp-based cognitive models. Work has begun to support models written in other languages such as Java, Python and Scheme. The hope of this expansion is to open the door for different types of cognitive research to be supported by this framework; thereby making the computational resource pool available to a broader cross-section of the cognitive modeling community. Other future work includes the exploration of special purpose processors such as the Graphics Processing Units (GPUs) which have enormous computational ability but do not support general process calculations currently used in the models of MindModeling@Home. Also, an effort is underway to better parallelize model component pieces. The ability to parallelize not only experiment parameterization configurations but also the cognitive model and environment itself provides the ability to support modeling efforts at a depth not possible in single processing environments.

Our attempts to address cognitive modeling’s growing computational demands are not limited to acquiring computational resources. The problem is also being tackled through research in exploration and search algorithms. MindModeling.org already incorporates an experimental “adaptive mesh refinement” (AMR) algorithm to intelligently prune and interpolate parameter spaces, and as a result of some very recent research efforts (Best et al., submitted), a smoothing algorithm has been added to help reduce resampling requirements. Integrating intelligent algorithms in the context of large scale computing has proven to be surprisingly challenging, but we see these challenges as opportunities for active involvement by the broader cognitive modeling and computational sciences communities.

A long range goal of MindModeling.org is to abstract away the challenges of using large-scale resources (and using large-scale resources well) for cognitive modelers. Unlike most volunteer computing projects, in which a single research project/team/center drives all of the computational demands, we would prefer to turn MindModeling@Home into a cognitive modeling community resource, enabling cognitive modelers all over the world to harness the power of the system by submitting their own cognitive model batch runs. We believe the distributed power of computational resources available to a distributed MindModeling community will facilitate new advances in computational cognitive process modeling otherwise not possible.

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was sponsored by the Air Force Research Laboratory’s Warfighter Readiness Research Division.

References

Best, B. J., Furjanic, C., Gerhart, N., Fincham, J., Gluck, K. A., Gunzelmann, G., & Krusmark, M. A. (submitted). Adaptive mesh refinement for adaptive exploration of cognitive architectures and cognitive models. Under consideration for presentation at the 9th *International Conference on Cognitive Modeling*.

Harris, J. (2008a). MindModeling@Home: A large-scale computational cognitive modeling infrastructure. In *Proceedings of the 6th Annual Conference on Systems Engineering Research* (http://cser.lboro.ac.uk/CSER08/).

Harris, J. (2008b). Maximizing the Utility of MindModeling@Home Resources. In *Proceedings of The Eleventh World Conference on Integrated Design and Process Technology.*

High-Level ACT-R Modeling based on SGT Task Models

Marcus Heinath (mhe@zmms.tu-berlin.de)
Berlin University of Technology, Center of Human-Machine Systems,
Franklinstraße 28-29, FR 2-7/2, D-10587 Berlin

Keywords: ACT-R; high-level modeling, macro compilation, cognitive activity pattern.

Introduction

Today's most important constraint on practical application of cognitive modeling as evaluation method is the high cost/low benefit ratio caused by a lack of support tools for modeling and by high requirements on sophisticated knowledge in cognitive psychology as well as artificial intelligence programming. The development of high-level languages to model human cognition based on low-level cognitive architecture is a current matter in the cognitive research community (see Ritter et al., 2006).

To open cognitive modeling to a wider user group and make the developing task easier and more accessible, the *Hierarchical Task Analysis Mapper* approach (HTAmap) has been developed. The main idea behind HTAmap is modeling on a higher level of abstraction. The foundation of the HTAmap approach is determined by:

- *HTAmap modeling process*: A formalized modeling process to minimize the transformation-gap between semi-formal, high-level sub-goal template task models (SGT; Ormerod & Shepherd, 2004) and formal, low-level ACT-R models (Anderson et al., 2004).
- *HTAmap modeling language*: An XML-based representation of task- and device-related knowledge within an integrated ACT-R model as well as strategies for model reuse and adaptation with regard to dynamic task environments and user skills.
- *HTAmap editor*: A Java-based software tool for systematic and semi-automated ACT-R modeling based on predefined model fragments.

The HTAmap modeling approach delivers cognitive modeling functionality based on predefined and modifiable *cognitive activity patterns* (CAP). Options for integrating separately defined device models of complex dynamic task environments (ACT-R graphical interface, AGI) as well as interfaces for embedded perception and action-models (AGI strategies) are also possible.

HTAmap modeling process

A large part of the cognitive model building process is transformed into a more simple pattern-oriented modification task. Figure 1 shows an overview of the different modeling steps. The output of the SGT method is the formal decomposition of the task and its re-description in terms of sub-goal templates, which represent a nomenclature for stereotypical operator tasks (TM_{SGT}). This re-description is the starting point for building the XML based HTAmap model (TM_{HTAmap}) and its automated transformation into ACT-R code (TM_{ACT-R}). In comparison to existing macro compilation approaches like ACT-Simple, G2A or ACT-Stitch (see Ritter et al., 2006 for an overview), that are widely limited to model user behavior in static task environments, HTAmap focuses strongly on simulations of dynamic task environments and their simplified coupling with ACT-R (IM_{AGI} →IM_{AGImap}→ IM_{ACT-R}).

HTAmap modeling language

The transformation gap between both levels of description is closed by the *HTAmap modeling language*. CAPs are being used to solve the mapping-problem from high-level to low-level description (SGT and ACT-R models). A CAP represents a generalized solution for the execution of an operator task (e.g. observe, scan, monitor, de-/activate, adjust, evaluate) using resources (i.e., declarative and

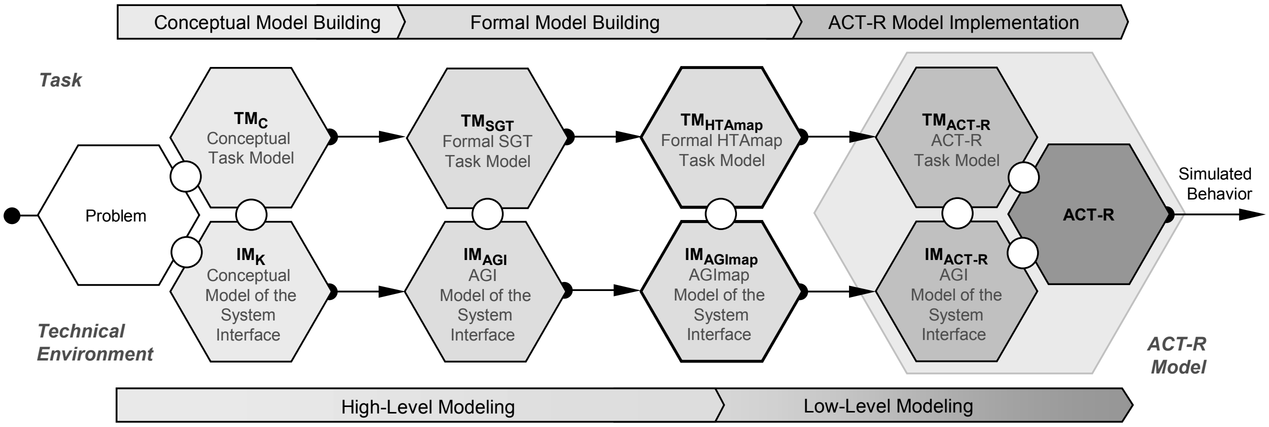
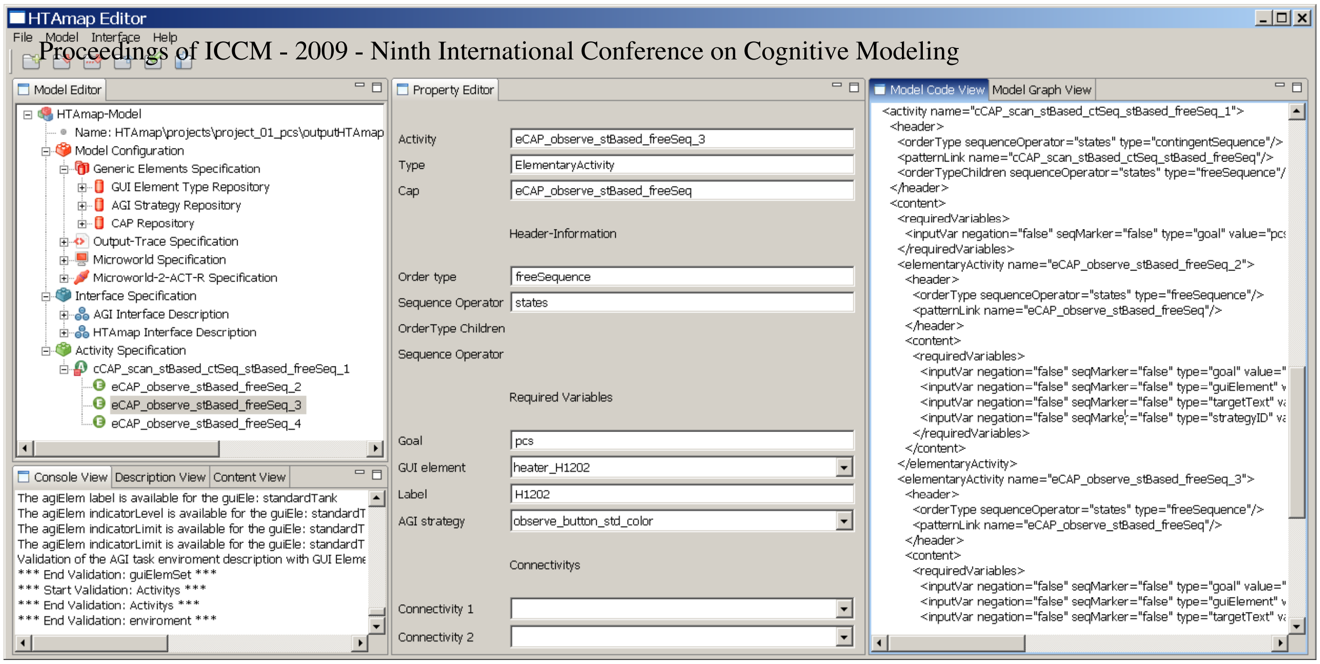

Conceptual Model Building
Formal Model Building
ACT-R Model Implementation
Task
Problem
TMC Conceptual Task Model
TMSGT Formal SGT Task Model
TMHTAmap Formal HTAmap Task Model
TMACT-R ACT-R Task Model
ACT-R
Simulated Behavior
IMK Conceptual Model of the System Interface
IMAGI AGI Model of the System Interface
IMAGImap AGImap Model of the System Interface
IMACT-R AGI Model of the System Interface
Technical Environment
ACT-R Model
High-Level Modeling
Low-Level Modeling

Figure 1: Overview of the HTAmap Modeling Process.



```
HTAmap-Model
  Name: HTAmap\projects\project_01_pcs\outputHTAmap
  Model Configuration
    Generic Elements Specification
      GUI Element Type Repository
      AGI Strategy Repository
      CAP Repository
    Output-Trace Specification
    Microworld Specification
    Microworld-2-ACT-R Specification
  Interface Specification
    AGI Interface Description
    HTAmap Interface Description
  Activity Specification
    cCAP_scan_stBased_ctSeq_stBased_freeSeq_1
      eCAP_observe_stBased_freeSeq_2
      eCAP_observe_stBased_freeSeq_3
      eCAP_observe_stBased_freeSeq_4
```

```
The agiElem label is available for the guiEle: standardTank
The agiElem indicatorLevel is available for the guiEle: standardT
The agiElem indicatorLimit is available for the guiEle: standardT
The agiElem indicatorLimit is available for the guiEle: standardT
Validation of the AGI task enviroment description with GUI Eleme
*** End Validation: guiElemSet ***
*** Start Validation: Activitys ***
*** End Validation: Activitys ***
*** End Validation: enviroment ***
```

Activity: eCAP_observe_stBased_freeSeq_3
Type: ElementaryActivity
Cap: eCAP_observe_stBased_freeSeq

Header-Information

Order type: freeSequence
Sequence Operator: states
OrderType Children
Sequence Operator

Required Variables

Goal: pcs
GUI element: heater_H1202
Label: H1202
AGI strategy: observe_button_std_color

Connectivitys

Connectivity 1:
Connectivity 2:

```
<activity name="cCAP_scan_stBased_ctSeq_stBased_freeSeq_1">
 <header>
  <orderType sequenceOperator="states" type="contingentSequence"/>
  <patternLink name="cCAP_scan_stBased_ctSeq_stBased_freeSeq"/>
  <orderTypeChildren sequenceOperator="states" type="freeSequence"/
 </header>
 <content>
  <requiredVariables>
   <inputVar negation="false" seqMarker="false" type="goal" value="pcs
  </requiredVariables>
  <elementaryActivity name="eCAP_observe_stBased_freeSeq_2">
   <header>
    <orderType sequenceOperator="states" type="freeSequence"/>
    <patternLink name="eCAP_observe_stBased_freeSeq"/>
   </header>
   <content>
    <requiredVariables>
     <inputVar negation="false" seqMarker="false" type="goal" value="
     <inputVar negation="false" seqMarker="false" type="guiElement" v
     <inputVar negation="false" seqMarker="false" type="targetText" va
     <inputVar negation="false" seqMarker="false" type="strategyID" va
    </requiredVariables>
   </content>
  </elementaryActivity>
  <elementaryActivity name="eCAP_observe_stBased_freeSeq_3">
   <header>
    <orderType sequenceOperator="states" type="freeSequence"/>
    <patternLink name="eCAP_observe_stBased_freeSeq"/>
   </header>
   <content>
    <requiredVariables>
     <inputVar negation="false" seqMarker="false" type="goal" value="
     <inputVar negation="false" seqMarker="false" type="guiElement" v
     <inputVar negation="false" seqMarker="false" type="targetText" va
```

Figure 2: Parts of the HTAmap Editor: *Model Editor* (left), *Property Editor* (middle) and *Model Code View* (right).

procedural memory as well as perception and action modules) to tackle a recurrent problem in a specific context. In general, CAPs describe the specific applications of identified SGT information handling operation at a less abstract level of ACT-R. In detail, a CAP comprises the necessary ACT-R declarative and procedural structures (i.e., production and chunks) and provides interfaces for parameterization regarding various tasks and environments. *Elementary cognitive activity patterns* (eCAP) are the basic building blocks for tasks. *Compound cognitive activity patterns* (cCAP) are used to represent the hierarchical structure of tasks. Complex task structures are represented via nesting of cCAPs: a cCAP can contain several eCAPs and/or other cCAPs (see Heinath & Urbas, 2007 for an overview). An activity is the combination of a CAP and task relevant knowledge. The HTAmap model itself is the combination of activities and task control knowledge (see Figure 2, right side). The kind of execution of activities is specified by their *order-types* which strongly refer to the plan concept of the SGT method: fixed, free or contingent sequence and simple choice.

HTAmap modeling editor

To proof the concept and reduce the entry level in cognitive modeling**,** a *HTAmap editor* has been implemented. Figure 2 shows a screenshot. The editor consists of three main parts: A *Model Editor* is used to define the repositories with the predefined model fragments (e.g., CAPs, AGI strategies, etc.), the AGI interface specification of the technical environment and the activity structure as result of the SGT task analysis. Secondly, task specific information are set in the *Property Editor*. Thirdly, the resulting HTAmap model - in XML syntax - is shown in the *Model Code View*. The transformation of HTAmap models into ACT-R code is realized automatically via the software tool *HTAtrans* that has been also integrated in the HTAmap modeling editor.

Conclusion and Outlook

HTAmap extends the modeling process to a wider user group. Main scope is the simplification of ACT-R modeling, the reuse of model fragments and consequently an increase of model application in the context of usability evaluation of Human-Machine Systems. The HTAmap model can be seen as a first approximation to model human behavior in ACT-R and provides a good starting point for further refinements. Currently, only a subset of CAPs is specified. The transfer of more ‘associated’ production rules into CAPs requires further verification and validation steps in the future.

Acknowledgments

This PhD project was funded by Deutsche Forschungsgemeinschaft (research training group GRK 1013 prometei).

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C. & Qin, Y. (2004). An integrated theory of the mind. Psychological Review 111 (4), 1036-1060.

Ormerod, T. C. & Shepherd, A. (2004). Using Task Analysis for Information Requirements Specification: The Sub-Goal Template (SGT) Method. In D. Diaper & N. Stanton (Hg.), *The handbook of task analysis for human-computer interaction* (p. 347–365). Mahwah, NJ: LEA.

Heinath, M. & Urbas, L. (2007). Simplifying the Development of Cognitive Models using a Pattern-based Modeling. In *Proc. of 10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*. Seoul, Korea.

Ritter, F. E., Haynes, S. R., Cohen, M., Howes, A., John, B. & Best, B. (2006). High-level Behavior Representation Languages Revisited. In *Proc. of the 7th ICCM* (p. 404–407). Trieste, Italy: Edizioni Goliardiche.

Dual-Task Strategy Adaptation: Do we only Interleave at Chunk Boundaries?

Christian P. Janssen (c.janssen@ucl.ac.uk)
Duncan P. Brumby (d.brumby@cs.ucl.ac.uk)
UCL Interaction Centre, University College London
Gower Street, London WC1E 6BT UK

How do people interleave attention when multitasking? One dominant account is that the completion of a subtask serves as a cue to switch tasks (e.g., Salvucci, 2005). But what happens if switching solely at subtask boundaries led to poor performance? In this paper, we investigate how drivers allocate their attention to a secondary phone dialing task while driving. We use a computational model to explain why we expect a particular pattern of task interleaving. These predictions are collaborated with empirical data of how participants dialed a UK-style number while driving.

A number of studies have investigated how drivers interleave dialing and driving (e.g., Brumby, Salvucci, & Howes, 2009; Salvucci, 2005). These studies have found that drivers dial in bursts, dialing several digits at a time before returning their attention to the road in between each burst. The manner in which the digits are dialed corresponds to the representational structure of the number in memory (i.e. xxx-xxx-xxxx for a US number). This supports the idea that the completion of a discrete subtask acts as a natural cue to switch from one task to another (Salvucci, 2005).

An alternative account of this behavior is that drivers complete as much of the secondary dialing task as possible while maintaining a stable lane position. Brumby et al. (2009) show that dialing three or four digits at a time is a particularly efficient task interleaving strategy: Any more interleaving incurs additional costs without significant improvement in lane keeping performance, and less interleaving sacrifices safety.

A limitation of previous data though is that it has focused almost exclusively on having participants dial US numbers. This is problematic because these numbers are made up of chunks of three and four digits each. Here we redress this issue by having participants dial a telephone number that has many more digits per chunk; namely, the xxxxx-xxxxxx representational structure that is used in parts of the UK. If drivers were to dial this number by interleaving only at the chunk boundaries, then they would have to dial five or six digits at a time. In the next section, we use Brumby et al.'s (2009) model to derive predictions for different task interleaving strategies, which are then compared to data from a study that investigates how participants dial a UK-style number while driving.

Model Exploration of Strategies

The model focuses on how different strategies for interleaving tasks affect critical performance metrics, namely, dial time and driver safety. We model a situation where the driver has to dial an eleven-digit number with chunks of five and six digits. We fit a parameter in the model that represents the amount of time it takes to dial each digit based on the human single-task baseline data (described below). Based on these data, key presses took 800 ms to execute, with the exception of the first key press of a chunk of digits, which took 1,200 ms to execute. We assumed that switching attention from driving to dialing, and back again, took 200 ms to execute. Furthermore, we assumed that disrupting the chunk structure of the dialing task carries an additional time cost of 100 ms to retrieve relevant state information from memory.

We used the above model to predict how a relevant subset of task interleaving strategies would perform. Each strategy differed in the number of digits that was dialed before attention was returned to driving. We use a simple convention to describe each strategy. A cross represents a key press and a dash represent a point where the model would interrupt dialing to return attention to the road. The strategies we evaluated were:

S1: xxxxxxxxxxx
S2: xxxxx-xxxxxx
S3: xxxxx-xxx-xxx
S4: xx-xxx-xxx-xxx
S5: xx-xxx-xx-xx-xx
S6: x-xx-xx-xx-xx-xx
S7: x-x-x-x-x-x-x-x-x-x-x

Of these S2 is notable because it interleaves only at the chunk boundary of the telephone number. Whereas, S3-S7 disrupt the chunk structure by interleaving more frequently, and as a result incur additional switch costs. We next give performance predictions for each strategy.

Model Predictions

Figure 1 shows the predicted lane deviation and dial time for each task interleaving strategy. There is a clear speed/accuracy trade-off between the time taken to complete the dialing task and vehicle lateral deviation. The important

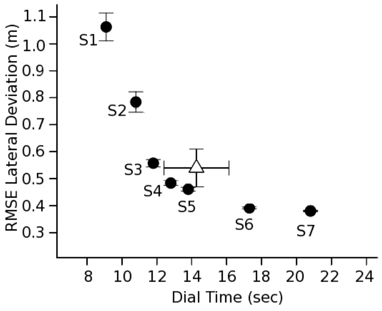


Figure 1: Model (dots) and human (triangle) data for total dialing time against lateral deviation

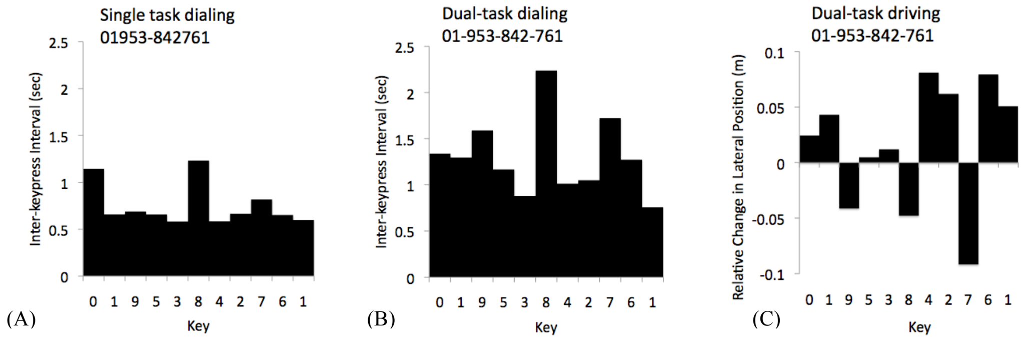


Figure 2: Data from empirical study. (A) inter-key press intervals on the single task, (B) inter-key press intervals on the dual task, and (C) relative change in lateral position on the dual task.

point to note here is the shape of the trade-off curve (linking S1 through S7). More frequent task interleaving carries the benefit of improved lane keeping but at the cost of increased time. However, at some point in this trade-off curve the improvements that are to be had in lane keeping become smaller with increased interleaving. Strategies that disrupt the chunk structure of the telephone number to dial three or four digits at a time (i.e., S3-S5) appear to cluster around the point where relatively safe driving performance is achieved while completing the dialing task relatively quickly. In contrast, the performance of the strategy that interleaves only at the chunk boundaries (S2) is in a region of relatively unsafe driving performance. These modeling results show that incurring the additional costs of disrupting the chunk structure of the telephone number is clearly worthwhile in terms of the improvement that is to be had for safety on the primary driving task. In the next section, we test the prediction that drivers should break up the chunk structure of a UK-style number in dual-task settings for improved safety.

Experiment

Twelve participants drove at a constant speed of 55 mph in a desktop based driving simulator that was projected on a 30-inch monitor. The driving environment consisted of a three-lane highway with safety cones placed on both sides of the centre lane to encourage staying inside lane boundaries. For dialing participants used a real mobile phone (Nokia 6300).

Participants started the experiment by learning the to-be-dialed number in a way that reinforced the intended chunk structure (i.e., xxxxx-xxxxxx). The number was shown on the monitor, but only digits from the current chunk were visible. Xs replaced the digits of the other chunk.

After training, participants completed 10 single-task dialing trials where they entered the number as quickly as possible (from memory). Participants then completed 10 single-task driving trials, and 20 dual-task trials (dialing while driving). For the dual-task trials, participants were instructed to drive as safely as possible while dialing. Each trial ended once the participant had dialed the number correctly, or after 60 seconds. To reinforce safe driving, feedback on average lane deviation was given after each trial. Error trials were excluded from the analysis.

Results

Figure 2a shows the average time to dial each key in the single-task context. These data suggest that when participants dialed the number as quickly as possible there were extended delays when entering the first and the sixth key of the number. These extended delays correspond to the first key of each new chunk of digits, presumably reflecting the time taken to retrieve the chunk of digits from memory. In the dual-task condition this pattern changes, however.

Figure 2b shows that while all key presses become elevated in the dual-task condition, there are extended delays at the third, sixth, and ninth digits. Figure 2c provides evidence that participants were choosing to suspend dialing at these points in order to bring the car back to the centre of the road (i.e., negative values indicate movement towards lane centre). Taken together these data suggest that participants were choosing to interleave dialing and driving in a manner akin to strategy S4.

Conclusion

Model and human data combined suggest that secondary subtask structure can be actively reconfigured to allow for more interleaving. Dialing is not necessarily interleaved at chunk boundaries instead people are willing to disrupt the explicit chunk structure of a secondary task when it is beneficial to do so in dual-task settings. This study is part of our ongoing effort to identify the influence of cognitive and environmental constraints on strategy adaptation in multitask situations. Future work should point out how increased or decreased demands on both types of constraints alter interleaving.

References

Brumby, D. P., Salvucci, D. D., & Howes, A. (2009). Focus on Driving: How Cognitive Constraints Shape the Adaptation of Strategy when Dialing while Driving. *Proc. of CHI 2009*.

Salvucci, D. D. (2005). A Multitasking General Executive for Compound Continuous Tasks. *Cognitive Science, 29*(3), 457-492.

"Cognitive Plausibility" in Cognitive Modeling, Artificial Intelligence, and Social Simulation

William G. Kennedy (WKennedy@GMU.Edu)
Krasnow Institute, George Mason University, 4400 University Avenue
Fairfax, VA 22030 USA

Keywords: cognitive plausibility, cognitive modeling, artificial intelligence, social simulation.

Introduction

The claim of "cognitive plausibility" is applied to cognitive models, Artificial Intelligence systems, and social simulations. All three research communities use the term but have different grounds for justifying their use. Can the semantics of the term be rationalized across all three disciplines?

The purpose of this poster is to advance the discussion of the meaning of cognitive plausibility started at the Cognitive Science Society annual meeting in 2008 (W. G. Kennedy, 2008) and continued within the International Journal for Social Robotics (William G. Kennedy, Bugajska, Harrison, & Trafton, 2009).

Different Views of Cognitive Plausibility

The three fields of cognitive modeling, Artificial Intelligence, and social simulation have different views of what constitutes acceptable justification for the use of the desired trait descriptor cognitive plausible.

Cognitive Plausibility in Cognitive Modeling

In cognitive modeling, the focus is primarily on replicating the observed behavior of a single individual and researchers believe theories, experiments, and models matching experimental data are needed to claim cognitive plausibility. With an interest in the make up of cognition, cognitive modeling is focused on experiments that demonstrate overall performance and experiments that isolate components of cognition, such as memory and reasoning. For cognitive modeling, matching human performance data includes matching the errors humans make.

Cognitive Plausibility in Artificial Intelligence

The argument of researchers in the field of Artificial Intelligence is that if the inputs and outputs of the system are comparable to those of humans, then the system is cognitively plausible. The field is less concerned with the cognitive plausibility of the internal components or processes because eventually all the components or processes are implemented in silicon. Hence the black box analogy with no cognitive plausibility claims about the inner working/components/subsystems, i.e., how the outputs are generated. The focus here is on the functional performance of the system. Artificial Intelligence is also not limited to demonstrating the performance of an individual, but is quiet happy to apply multiple and distributed intelligent agents to obtain cognitive performance. Finally, it should also be noted that the goal of AI research is not simply replicating human performance, but understanding the mathematical principles behind it as demonstrated by the building of systems that match and may one day surpass human performance.

Cognitive Plausibility in Social Simulations

The social sciences have the challenge that they cannot conduct experiments on real societies. As a result, social simulations have long relied on functions describing the behavior of rational individuals and behavior of small and large groups as a whole. These formulations go back to difference equations describing the effects of the number of combatants and weapons (e.g., swords and shields or bows and arrows) on one side reducing the number of combatants on the opposing side in each of a series of exchanges (Lanchester, 1916). However, even with the development of much more sophisticated social simulations, the "homo economicus" assumptions of perfectly rational behavior have been criticized by many including Herbert Simon and the community now recognizes a need for better cognitive plausibility in their models of human behavior (Sun, 2006), but is without a definition of what that means.

Common Ground

To find common ground, Nobel prize winner Richard Feynman is instructive. Richard Feynman lectured that "All other aspects and characteristics of science can be understood directly when we understand that observation is the ultimate and final judge of the truth of an idea." (Feynman, 1998) But cognitive plausibility would then be dependent on "observing" cognition. While we may be getting close to observing cognition directly (Anderson, 2007), simulation has been suggested as a third branch of science, adding to theoretical and experimental branches. Herbert Simon wrote that simulation can be of help to understand the natural laws governing the inner workings of a system from the top down "because the behavior of the system at each level is dependent on only a very approximate, simplified, abstracted characterization of the system at the level next beneath" (Simon, 1969). He also noted that this approach is similar to the foundations for the entire subject of mathematics.

In proposing a unified theory of cognition, Allen Newell proposed several levels within the human cognitive architecture (Newell, 1990) which Ron Sun, and others, simplified to: the sociological level, the psychological level, and the physiological level (Sun, 2006). Finally, John Laird presented an organization to cognitive architectures based on their goal and basis in his plenary presentation at the Cognitive Science Society in 2007. Combining these concepts provides a basis for unifying the various uses of cognitive plausibility for these three areas of research.

Differentiating Cognitive Plausibility

The old problem with the definition of intelligence was that if it was defined in terms of something human did, then no artifact could ever be intelligent and intelligence was not acceptably defined without reference to humans. Similarly, for a cognitive model or system to be worthy of belief, i.e., plausible, is needs to convince us that it is performing cognition. To avoid the arguments about the validity of the Turing Test, a basis for differentiating the uses of cognitive plausibility is proposed here based on observed performance and system levels.

Consider a cognitive system as being made up of one or more layers of systems. I propose defining the cognitive plausibility of any system or layer as:

Proposal for discussion: to be considered “cognitively plausible,” a system must be capable of performing as well as humans do on cognitive tasks or be plausibly built on components that have met this test.

To perform “as well as humans do” means matching human performance data. Of course, what it means to match human data is a separate discussion and has been discussed elsewhere, see (Fum, Del Missier, & Stocco, 2007) and (Gluck, Bello, & Busemeyre, 2008) Ron Sun (Sun & Ling, 1997) has proposed three “types of correspondence between models and cognitive [systems]”: behavioral outcome modeling (roughly the same behavior), qualitative modeling (same qualitative behavior), and quantitative modeling (“exactly the same” behavior).

Note that this does not address matching human errors in performing cognitive tasks. Being able to match human behavior, both successes and errors, is proposed to be beyond the basic concept of cognitive plausibility. I suggest describing the ability of a system to match human performance <u>including</u> errors as being **“genuinely cognitive plausible”**. Further, to address construction of systems from cognitively plausible subsystems, I propose that cognitively plausibility can be “deep” or “shallow”. **“Shallow cognitive plausibility** is cognitive plausibility at only one layer of a cognitive architecture and **“deep cognitive plausibility”** is cognitive plausibility across more than one layer.

For social simulations, cognitive plausibility can be based on using cognitively plausible models for individuals at the next lower level, i.e., for the individuals that make up the society. Using the proposed definition of cognitively plausible, the field of AI can base its use of the term on meeting or exceeding human-level performance. Finally, cognitive model researchers can base their use of the same term on the cognitive plausibility of matching human performance or on a plausible construction of cognitively plausible modules. All fields can clarify their cognitive plausibility as shallow, deep, or genuine. This is the subject of discussion for this poster.

Acknowledgments

This work has been informed by discussions with many individuals including John Anderson, Ken De Jong, Bonnie John, John Laird, Greg Trafton, an anonymous ICCM reviewer, and others over the past year. The work is supported by the Center for Social Complexity of George Mason University and by the Office of Naval Research (ONR) under a Multi-disciplinary University Research Initiative (MURI) grant no. N00014-08-1-0921. The opinions, findings, and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the sponsors.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford: Oxford University Press.

Feynman, R. P. (1998). *The Meaning of it All*. Reading, MA: Perseus Books.

Fum, D., Del Missier, F., & Stocco, A. (2007). The cognitive modeling of human behavior: Why a model is (sometimes) better than 10,000 words. *Cognitive Systems Research, 8*, 135-142.

Gluck, K., Bello, P., & Busemeyre, J. (2008). Introduction to the Special Issue [on Model Comparison]. *Cognitive Science, 32*(8), 1245-1247.

Kennedy, W. G. (2008). *Cognitive Plausibility of Cognitive Maps and Modeling of Others.* Paper presented at the 30th Annual Conference of the Cognitive Science Society, Washington, DC.

Kennedy, W. G., Bugajska, M., Harrison, A., & Trafton, J. G. (2009). "Like-Me" Simulation as an Effective and Cognitively Plausible Basis for Social Robotics. *International Journal of Social Robotics, 1*(2), 181-194.

Lanchester, F. W. (1916). *Aircraft in Warfare: The Dawn of the Fourth Arm*. New York: D. Appleton & Co.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Simon, H. A. (1969). *The Sciences of the Artificial*. Cambridge, MA: The MIT Press.

Sun, R. (Ed.). (2006). *Cognition and Mulit-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge: Cambridge University Press.

Sun, R., & Ling, C. X. (1997). Computational cognitive modeling, the source of power and other related issues. *AI Magazine, 19,* 113-120.

Automated Data Analysis for Operator Modeling

JC LeMentec (le_mentec_jc@cat.com)
Caterpillar Inc.
Technical Center, Bldg. E
P.O. Box 1875
Peoria, Illinois 61656-1875 USA

Abstract

IDA is a new architecture that combines data analysis and operator model creation and simulation. It enables the development of operator models by finding causality patterns in recorded machine operation data. A new symbolic constraints based reasoning system is used to accurately and robustly segment data. These segments can then be used to discover an operator strategy expressed as hierarchical set of Tasks.

Keywords: Operator Modeling, Cognitive Model, Data Analysis, Data Segmentation, Autodig, Wheel Loader.

Introduction

To assess the design of a new machine design, Caterpillar relies on an interactive simulation of machine, soil and human operator. The operator model must reproduce the levers, steering and pedals commands in response to environmental inputs, as a human operator would do when performing particular tasks.

The expertise for conventional cognitive models is usually obtained through an interview process with domain experts. While still useful for operator models, interviews alone are not sufficient as a large part of the behavior is unconscious. Fortunately, operator activities can be scrutinized in great details through recorded data of machine operation. Machines are routinely instrumented with various sensors, and a large amount of data is available.

Data Collection

To develop an operator model, we need to collect all data related to the operator-machine interactions. Operator commands are usually directly measured and many perception stimuli can be inferred from various sensor data. The bucket position of a wheel loader, for example, can be determined by the various cylinders extension of the linkage. Other visual information, such as the position of the bucket in relation to a pile of dirt, can be deduced from the force exerted by the soil on the bucket (calculated from cylinder pressure). When analyzing the data with the appropriate tool some patterns start to emerge. These patterns, in turn can be used to identify some general principles of machine operation that can be used to create an operator model.

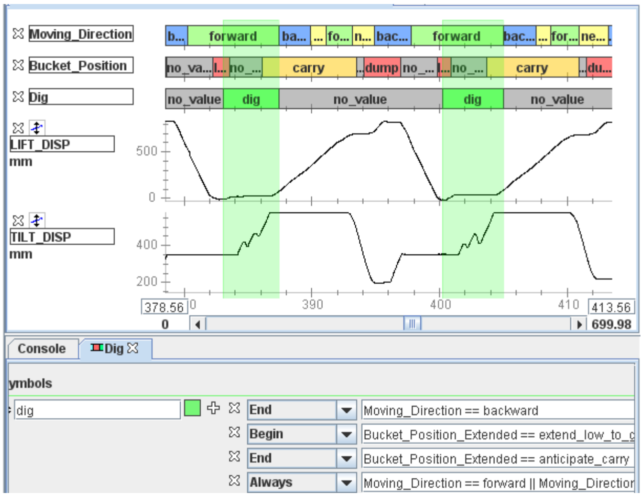
Looking at gigabytes worth of data can be an intimidating task. Common practice is to isolate one occurrence of the particular action intended to be modeled (for example a dig operation) and analyze it in detail. This approach however does not provide the distinction between random actions and regular patterns. To really understand the trends, a large number of digs must be analyzed and compared side by side.

Data Segmentation

The first step is to identify the part of the data that relate to the operation to be studied. A skilled engineer can do this by recognizing some patterns of cylinders extension and a few other sensors in time series data. Yet, automating this segmentation has been an elusive goal, suffering in time accuracy and having false positive or false negative detections. Recognizing the importance of such automated data segmentation, we designed our operator-modeling tool, named IDA, to integrate a new data visualization and segmentation tool.

Constraints Based Segmentation

IDA uses a concept of constraints based reasoning to perform segmentation. A segment is defined by a set of constraints chosen from eight possible types of constraints. IDA uses a special display to show numerical data channels and symbolic data channels simultaneously. Each Symbolic channel is displayed as a thick line where each segment appears as a colored rectangle corresponding to the color associated with this type of segment. Figure 1 shows an example of the data display and constraints definition for a dig segments during truck loading operation with a wheel loader.



dig 🟩
End — Moving_Direction == backward
Begin — Bucket_Position_Extended == extend_low_to_c
End — Bucket_Position_Extended == anticipate_carry
Always — Moving_Direction == forward || Moving_Directior

Figure 1: symbolic channels for data segmentation.

Numerical channels can also be calculated with user provided or predefined functions, such as filters or derivatives. Complex segmentations involve dozens of symbolic channels, each defined by its own set of constraints that refer to other symbolic channels or numerical channels. By combining numerical and symbolic processing it becomes possible to create segmentation that are both accurate and robust.

Finding Patterns In Data

Once particular segments of interest have been identified, IDA provides another view to show all the segments of a given type side by side. Figure 2 show a segmented view of dig segments for an operator. In this view, a regular pattern is visible for both tilt and lift command.

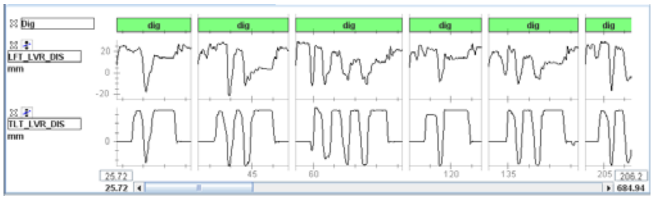


Figure 2: Tilt and Lift commands for an operator during dig

The view has been reduced to fit the available space in this paper. A large display would show more segments at a time and thus reinforce the side-by-side comparison. Finally, the horizontal scroll bar, at the bottom of the display, provides a quick way to browse hundreds of segments. Any irregular segment would stand-out and could be analyzed further. We believe that showing the data with an appropriate view enables the operator model developer to discover new patterns that would not be apparent with a more conventional data display.

Automated Pattern Discovery

A special type of function, called an aggregate function, is used to calculate a single value for each segment. This kind of data reduction can be used, for example, to calculate the energy spent on each dig by using an integral function (an aggregate function) applied to the power delivered by the cylinders over time. Another example of an aggregate function is the "begin_value" function that simply reports the value of a numerical channel at the beginning of a segment. When applied to the gear channel for dig segments, it shows the gear at the beginning of each dig. Figure 3 shows the begin values for gear, speed and throttle. Data channel values at the beginning of a segment present a particular interest when searching for causality of the initiation of a type of task (like digging in our example). If the operator starting the dig action once the machine had reached a specific speed during the deceleration caused by contacting the pile, we would expect to see a consistent value at the beginning of each dig.

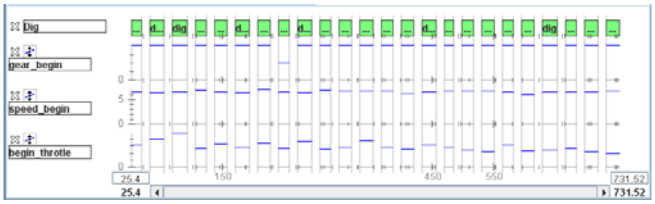


Figure 3: gear, speed and throttle values at the beginning of each dig segment

By calculating automatically the begin values for all the numerical channels available and sorting them by increasing order of their coefficient of variation, the most promising candidate for causality would start to emerge. By applying this calculation on a very large number of data, it is possible to automatically discover some general trends.

Conclusion

IDA, which stands for Integrated Development and Analysis, is currently used as an internal tool at Caterpillar for data analysis and operator model creation and simulation. The combination of both analysis and operator model building in the same architecture opens new possibilities for the creation of operator models. The models are, in a sense, created from the data and compared to actual data at different stages of their development.

The operator modeling approach in IDA relies on serial process definitions related to COGNET (Zachary, LeMentec & Loiederman 1999) and GOMS (Card, Moran & Newell 1983). The concept of data driven modeling could also be applicable to other cognitive modeling systems that rely on production rules such as ACT-R (Anderson & Lebiere 1998) or SOAR (Newel 1990) as long as observable data are available.

In the future, the IDA architecture could be extended to implement knowledge representation for different purposes.

Acknowledgments

The author thanks Caterpillar and its Virtual Product Development division for supporting this work.

References

Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Newell, A. (1990). *Unified Theories of Cognition.* Harvard University Press. Cambridge, Massachusetts.

Card, Stuart; Thomas P. Moran and Allen Newell (1983). *The Psychology of Human Computer Interaction.* Lawrence Erlbaum Associates. ISBN 0-89859-859-1.

Glenn, F., Zachary, W., Le Mentec, J. C., & Loiederman, E. (1999). Modeling perceptual and motor behavior in a cognitive context. *Proceedings of SAE International Conference on Digital Human Modeling for Design and Engineering*. Warrendale, PA: Society of Automotive Engineers.

Incremental processing and resource usage

Deryle Lonsdale, Jeremiah McGhee, Ross Hendrickson, and Carl Christensen
Corresponding author: lonz@byu.edu
Department of Linguistics and English Language
Brigham Young University
Provo, UT 84602 USA

Keywords: Soar; incremental parsing; verb valency; resource usage.

Background

Within the community engaged in Soar-based cognitive modeling (Newell, 1990), some work has focused on parsing natural language input text. An early version of the system (Lewis, 1993) performed syntactic analysis based largely on the Government & Binding (aka Principles & Parameters) framework, including X-bar theory for constituency.

The system, called NL-Soar, receives lexical input word-by-word, and lexical access is performed for each word in turn. The system then attempts to integrate the incoming words and their related information incrementally into linguistic models: a syntactic X-bar parse tree (which will be the focus of this paper), as well as semantic and discourse structures (that will not be addressed here). All potential and possible syntactic material is considered in piecing together licit constructions. Constraints operate to rule out attachments that do not follow standard principles. In certain cases, some types of limited structure can be undone and reformulated when ongoing hypotheses prove untenable in the presence of new incoming words.

NL-Soar was later updated to retrieve knowledge from WordNet (Fellbaum, 1998) that provides relevant morphological and syntactic information for all of the senses and homographs of the word in question (Lonsdale & Rytting, 2001).

In more recent work (Lonsdale, 2006) following on directly from Lewis' prior contribution, we replaced the GB-style syntactic model with one more closely reflecting assumptions of the Minimalist Program (MP) (Chomsky, 1995). Though cognitive modeling has been pursued with other syntactic theories, Minimalism has not seen the same scrutiny, though some parsing and psycholinguistic research has been done within the MP (e.g. (Fong & Hirose, 2005)) and cognitive modeling within MP has been called for (Edelman & Christiansen, 2003). Our work has included adding more functional projections, feature checking, and movements. In addition, two hierarchies of projections—one for clausal structure and one for nominal structure—are available to specify and license construction of hierarchical layers in the syntactic model.

To simplify the work for this paper, no sentences with adjunction, coordination, or complex clauses are considered. We ignore movement of constituents, such as a subject's putative movement from its original position in the specifier of vP to its final position in the specifier of TP, due to the Extended Projection Principle. Finally, intransitives are treated identically whether unergative or unaccusative.

Ongoing work has focused on whether the new syntactic mechanism is capable of supporting incremental processing, and at what cost. In this paper we summarize work done to assess how two different parsing control strategies support parsing in as incremental a fashion as possible. We also attempt to quantify resource usage necessary to parse different types of sentences according to the two different processes.

General remarks

Our study of this question involved running several sentences through the system and running various statistical profiling processes to measure processing load. We are using the newest version of the Soar cognitive modeling system, which represents a substantial revision of the basic NL-Soar code base, not all of which has been converted to date.

The system is agent-based, and information enters from the exterior environment. In our case, incoming words are collected serially into a buffer until they are attended to. Attention involves a lexical access operator which retrieves associated information from WordNet and other lexical resources at the system's disposition. After lexical access, processing proceeds differentially depending on the strategy employed.

At the current time in this version of the system, learning is turned off. Hence the pursuit of hierarchical goals is not enabled, and the agent's only task is to solve the sentence.

The project/attach strategy

The first strategy retains some of the assumptions of the original GB-based theory. For example:

- Lexical categories are projected as completely as possible as soon as possible. Zero-level nodes are projected to XP nodes via one operator.
- Projections (except v) only grow when lexically licensed .
- Structures are extended via the hierarchy of projections as soon as possible.
- Attaching complements and specifiers into pre-existing structure is performed as a last resort, and only when licensed.
- New words aren't attended to until all possible structure is built incrementally.

Thus, in processing a simple intransitive sentence, the agent posits structure as soon as possible, completing the subject's NP and then DP structure before the verb is encountered. Once the verb is attended to, it is projected up to a TP per information provided via operators that consult the clausal hierarchy of projections. The subject is then attached into the specifier position of the TP node.

Processing is similar for transitive verbs, with an additional step to attach the direct object into the complement position of the V-bar node. For ditransitive verbs, the first object is attached into the specifier of the VP node as soon as it is completed; the second object is attached into the complement of the V-bar node as is done with transitive verbs.

The bottom-up merge strategy

This strategy follows recent assumptions for minimalist analysis (Adger, 2003). In particular:

- Structure is only projected when licensed at any stage.
- Projections (except v) only grow when lexically licensed .
- There are separate operators for projecting nodes at the intermediate (X-bar) and phrasal (XP) levels.
- Separate operators perform First Merge (incorporating complements) and Second Merge (incorporating specifiers).
- Projection to XP is only possible when licensed.
- Merge can only occur when licensed via features that need to be checked and deleted.
- New words aren't attended to until all possible structure is build incrementally.

In this case, the agent projects only as much structure as possible, one node at a time, as licensed. Intransitive verbs are constructed in a fashion largely similar to the previously mentioned strategy. However, with transitive verbs the V-bar node is not built until the direct object's structure has been completed. Similarly, the TP node is not constructed until the subject can be combined into the specifier position of a T-bar root node. More interestingly, ditransitive instances require that no V-bar node can be constructed until the second object has been completed. Only then can it undergo First Merge to combine with the lexical verb. Then the first object combines with the V-bar node (i.e. in its specifier) via Second Merge to create a VP.

Results

The second strategy required substantial resource usage, especially for ditransitive constructions. This is because verb phrasal structure must be held in abeyance until both internal arguments are completed.

A post-hoc analysis of the processing statistics showed that almost all of the changes in working memory are due to lexical access and the data retrieved at that time.

Figure 1 shows memory usage over time (measured in decision cycles) for the three canonical types of sentences (intransitive, transitive, and ditransitive).

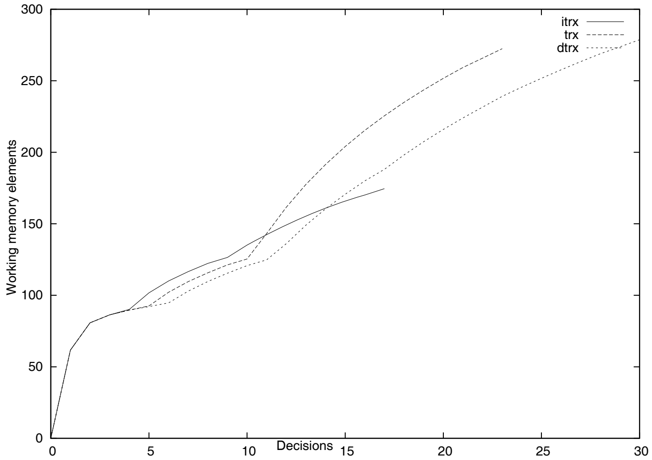


Figure 1: Working memory usage across decision cycles for intransitives, transitives, and ditransitive sentences.

Conclusions

Though only representing a core set of syntactic possibilities for sentences, this work has shown that the two strategies entail different amounts of resource usage, which can be quantified via profiling in the cognitive modeling system.

References

Adger, D. (2003). *Core syntax: A minimalist approach*. Oxford University Press.

Chomsky, N. (1995). *The Minimalist Program*. MIT Press.

Edelman, S., & Christiansen, M. H. (2003). How seriously should we take Minimalist syntax? *TRENDS in Cognitive Sciences*, *7*(2), 60–61.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Fong, S., & Hirose, Y. (2005). Minimalist Parsing and Japanese: on theory and experiential factors. In Y. Otsu (Ed.), *Proceedings of the Sixth Tokyo Conference on Psycholinguistics* (pp. 95–116). Tokyo: Hituzi Syobo Publishing Co.

Lewis, R. L. (1993). *An architecturally-based theory of human sentence comprehension*. Unpublished doctoral dissertation, Carnegie Mellon University.

Lonsdale, D. (2006). Learning in minimalism-based language modeling. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society* (p. 2651).

Lonsdale, D., & Rytting, C. A. (2001). Integrating WordNet with NL-Soar. In *Proceedings of NAACL-2001: WordNet and other lexical resources—Applications, extensions, and customizations.* Association for Computational Linguistics.

Newell, A. (1990). *Unified theories of cognition.* Harvard University Press.

Visualizing Egocentric Path Descriptions: A Computational Model

Don R. Lyon (don.lyon@mesa.afmc.af.mil)
L3 Communications at Air Force Research Laboratory
6030 S. Kent St., Mesa, AZ 85212-6061 USA

Glenn Gunzelmann (glenn.gunzelmann@mesa.afmc.af.mil)
Air Force Research Laboratory, Mesa Research Site
6030 S. Kent St., Mesa, AZ 85212-6061 USA

The ability to visualize spatial information from verbal descriptions is an important component of human cognition. A common example is generating a 'mental picture' of driving directions. Such directions can be given either from an external viewpoint, as if viewing a map (exocentric description, e. g. 'Left' is always West), or from the point of view of a traveler moving along the path (egocentric description, e.g. 'take a right, go 1 block, then turn left…').

Directions for driving imply a horizontal-plane, two-dimensional mental image, but one can also describe paths through 3D space. We have studied the capacity of people to visualize complex 2D and 3D paths using the Path Visualization (PV) task, which provides an objective measure of visualization accuracy. We developed an ACT-R model of visualization capacity for exocentrically-described paths (Lyon, Gunzelmann & Gluck, 2008). According to this model, the capacity to visualize an exocentric path description is limited primarily by decay and spatial interference in an exocentric-viewpoint image constructed in visuospatial working memory.

Here we extend this model to account for people's ability to visualize complex *ego*centrically-described paths. We suggest that the primary internal representation used for egocentric-path visualization is the same as in the exocentric case -- an exocentric-viewpoint mental map. This implies that egocentric descriptors would need to be converted to an exocentric reference frame before they could be added to the map. If this conversion process involves additional cognitive operations, and these operations take time, then items in spatial working memory should undergo more time-based activation decay for egocentric descriptors than for exocentric descriptors. Thus we hypothesized that accuracy for egocentrically-described paths would be lower than accuracy for exocentrically-described paths.

Model Predictions

Since our hypothesis was that egocentric-to-exocentric conversion time would be the primary cause of any accuracy difference between exocentric and egocentric conditions, we developed a model of egocentric path visualization by starting with the exocentric-case model and adding an egocentric-to-exocentric conversion process. We then conducted a rather strict test by using all of the same parameter values that were used in the model for exocentric descriptors, and adding only one parameter – the execution time of the egocentric-to-exocentric conversion process – to the model for the egocentric case. As shown below, the additional time required by this process does indeed cause the model to predict that visualizing egocentrically-described paths will be less accurate than the exocentric-description case.

Method

Each of thirteen participants completed ten 30-trial PV sessions, five with exocentric path descriptions, five with egocentric. On each trial, 15 unit-length path segment descriptions (e.g. 'Left 1') were presented for 2 sec. each. In the exocentric condition, directions were relative to a fixed reference frame, so that 'Left' would always refer to the left side of an imaginary 5 x 5 x 5 three-dimensional space within which the paths were generated. In the egocentric condition, directions were relative to the current facing of a hypothetical traveler on the path. In both conditions, the participant read each path segment description, decided whether the endpoint of that segment intersected with any previously presented part of the path, and responded *yes* or *no* with a keypress. Half of the paths could wander randomly through three dimensions; the other half were 2D paths constrained to either a coronal ('picture'), sagittal, or horizontal plane through the center of the space.

Results

As predicted, paths described exocentrically were visualized more accurately than paths described egocentrically (F(1,12)=18.5, p<0.001). There was no overall effect of path type. Model predictions fell close to human overall accuracy for both exocentric and egocentric conditions (Figure 1). The egocentric model fit was obtained using an egocentric-to-allocentric conversion time of 700 msec.

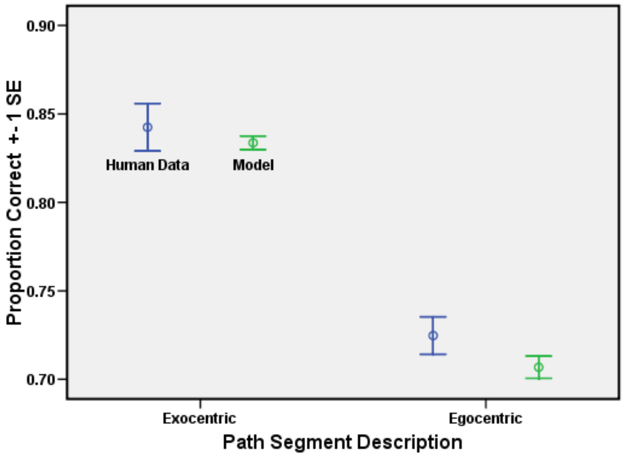


Figure 1. Visualization accuracy for exocentrically- and egocentrically-described 3D and 2D paths.

Although the model accounts well for the overall difference in accuracy between egocentric and exocentric conditions, it does not track human accuracy for different path types within the egocentric condition. In particular, the model predicts better performance for 3D paths than for 2D paths, whereas people certainly do not find 3D paths easier. A possible explanation for this discrepancy is that many paths in the 3D condition require particular kinds of ego-to-allo transformations that people find especially difficult. For example a virtual traveler on the path in the 3D condition would often be head-down, or in some other unusual body orientation, making it difficult for people to translate terms such as 'up', or 'left' into an absolute reference frame.

We tested the relative difficulty of different kinds of ego-allo transformations in an ancillary study in which movements through the same 5 x 5 x 5 virtual grid were visually depicted (from an egocentric perspective), rather than verbally described. Participants were allowed all the time they needed to accomplish each ego-to-allo translation, visualize the next segment, and produce a response. The data reveal a generally systematic increase in response time as either facing direction or body axis direction deviated from a forward-facing, upright alignment. People took an average of about 250 additional msec. per 90 deg. of facing misalignment. For body axis orientation, the time required for each 90 deg. of misalignment was roughly equal to three 90-deg. 'steps' of facing misalignment, or 750 msec.

We therefore modified the model by refining ego-to-allo translation into two components: (1) a perspective-taking process that requires additional time as body axis and facing misalignment from upright-forward increases, and (2) a segment generation process that requires a constant amount of time. The average total time for these processes was constrained by the previous model fitting to be 700 msec. By default, the generation process required one 50-msec. ACT-R cognitive cycle, leaving 650 msec. for the perspective-taking process. Because the average number of perspective misalignment 'steps' was 4, each step time was set at 650/4 (approx. 162 msec.). This change required an adjustment in retrieval threshold from -0.9 to -0.7 to maintain overall accuracy comparable to the human data.

The difference between the 250-msec step time obtained in the ancillary study and the 162-msec time used by the model in the main study is probably due to the 2-sec. deadline imposed for responses in the latter. This deadline was necessary to assure that performance was driven by factors (such as decay and interference) that influence spatial visualization itself, and not by non-spatial strategies that could conceivably be used given unlimited time.

This model resulted in a substantially better (but not ideal) fit to the data for different sub-conditions (Figure 2). A better fit might have been obtained by optimizing the division between perspective-taking and segment generation processes, but this would have required another parameter.

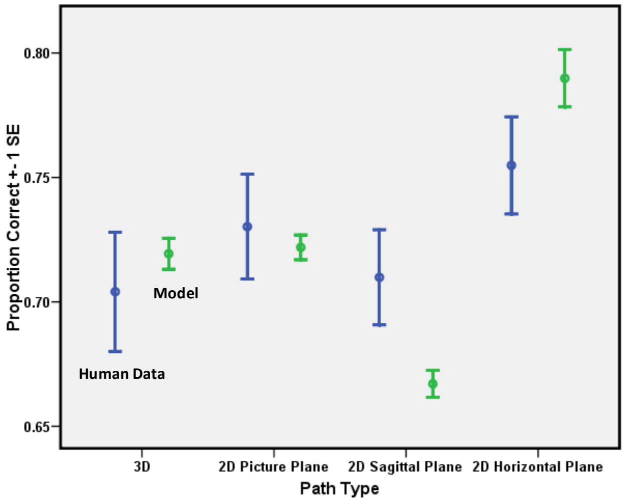


Figure 2. Human data vs. revised model for 3D paths and different kinds of 2D paths.

Conclusions

Human capacity to construct a mental image of new, complex spatial material is sharply limited. In particular, when people try to visualize a verbally described path, capacity limits are well-described by a model in which the activation of each new segment decays with time, and segments that are nearby in imaginary space interfere with each other (Lyon, Gunzelmann & Gluck, 2008).

Here we have shown that path visualization accuracy depends on the nature of the path description. If the path is described in exocentric terms, using fixed reference directions external to the path itself, accuracy is higher than if it is described in egocentric terms, from the point of view of a traveler on the map, in which the absolute direction of 'left' and 'right', etc. depend on the direction the traveler is imagined to be facing.

The success of the model in the egocentric case suggests that the basic processes that limit visualization accuracy (decay and interference) operate for both kinds of descriptions. The key difference is that egocentric descriptions require a translation process to convert them to fixed, exocentric directions. Under the conditions of this study, ego-allo translation required, on average, about 700 msec., but the time varied considerably depending upon the degree of misalignment of a virtual traveler on the path from an upright, forward-facing orientation.

Acknowledgments

We thank Ben Sperry and Rayka Mohebbi for software development, Monica Nguyen for research assistance, and the U. S. Air Force Office of Scientific Research for support (Grant #02HE01COR).

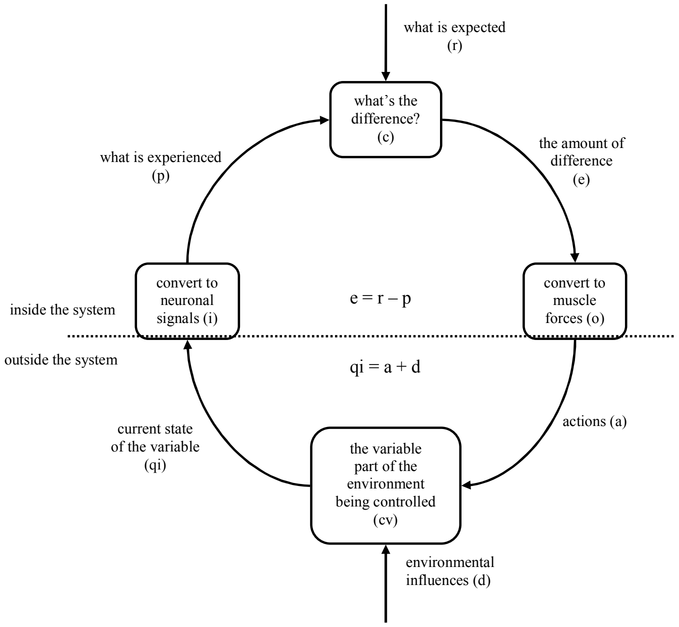
Reference

Lyon, D. R., Gunzelmann, G., & Gluck, K. A. (2008). A computational model of spatial visualization capacity. *Cognitive Psychology*, *57*, 122–152.

Perceptual Control Theory as a Framework for Modelling the Function and Dysfunction of Living Systems

Warren Mansell (warren.mansell@manchester.ac.uk)
School of Psychological Sciences, University of Manchester
Manchester, M13 9PL, UK

Perceptual Control Theory (PCT; Powers, Clark & McFarland, 1960; Powers, 1973, 2005) is a self-regulatory framework developed from control system engineering. The theory has been widely applied across the social and life sciences (see pctweb.org) yet it is not popularly recognised or understood within academic circles. The negative feedback loop, depicted below (Carey (2008) is integral to PCT.



The negative feedback loop functions to maintain a variable (*qi*) at, or close to, some reference value (*r*) despite the environmental disturbances (*d*) that tend to vary it. The variable is converted to neuronal signals (*i*) which are substracted from *r* to form an error (*e*) which drives the output of the system (*o*) to modify the controlled variable (*cv*) through actions (a) on that variable.

PCT proposes that human information processing is achieved through the functioning of multiple negative feedback loops and their organization with respect to one another. More specifically, the loops are organised within a hierarchical organisation. Higher level goals are achieved through setting the reference values (equivalent to goals) for lower level systems that in turn set reference values for lower order systems, and so on (see Figure 2; Runkel, 2003). In other words, a higher order goal never has a direct impact on behaviour; it merely sets the perceptual standards for lower order systems.

Within PCT, the lowest systems control the *intensity* of a perceptual signal. The systems increase in the complexity of perception with higher levels. The highest levels are concerned with maintaining *principles* (e.g. loyalty) and, in turn above, *system concepts* (e.g. the self in the world).

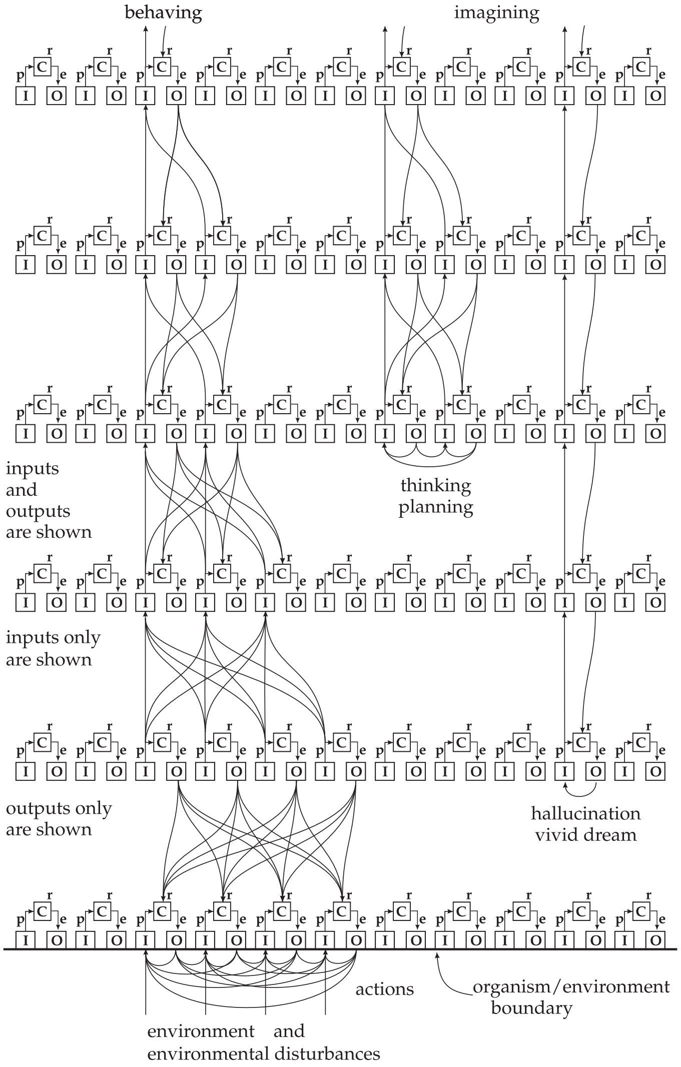
Any living system will have control hierarchies for many variables, and so the system as a whole needs to balance and regulate multiple lower level systems so that they each achieve their ends; in other words - so that they do not experience prolonged error. In order to manage this, the system leads to be able to modify its internal properties at the appropriate locations throughout the control hierarchies. For example, the *gain* of a system is a parameter that determines how much the error affects its output (output = error x gain). By changing the gain of different systems, they vary in the extent to which they exert control over specific variables. It is possible that at high levels in the system this may be felt by the individual as the reprioritisation of goals. Therefore, an optimization process called *reorganisation* is hypothesised to create variation in the properties of the control systems until error is reduced (Marken & Powers, 1989).

PCT considers chronic *conflict* between control systems as they key cause of psychological distress (see Mansell, 2005). Internal conflict occurs when two control systems attempt to control the same variable within different ranges. internal conflict is most likely to occur in these higher level systems where incompatible reference signals are set.

Testing PCT

There are many ways that PCT has been tested within computer models (see pctweb.org). They have essentially involved building a PCT model of a specific system and then evaluating the match between the model and the observations of how the real system works.

For example, Bourbon (1995) developed a PCT model of a simple tracker task that matched observed behaviour to a very high level (r > .95) over a period of five years within an individual. Marken (2001) describes a PCT model of catching fly balls in baseball that demonstrates of closer match with observed behaviour than alternative approaches. His model utilizes two parallel control systems – one that controls the vertical velocity of the image of the ball on the retina, and another that controls the lateral displacement of this image. These simulations provide a close match to data collected through mounting a portable video camera on the shoulder of a baseball fielder. In his latest book, Powers (2008) provides a wide range of different computer demonstrations of PCT.


behaving
imagining
p
C
e
r
I
O
inputs and outputs are shown
thinking planning
inputs only are shown
outputs only are shown
hallucination vivid dream
actions
organism/environment boundary
environment and environmental disturbances

Current Research Directions

The current research direction taken in my research group at the University of Manchester is to use PCT to model human psychological function and dysfunction, with particular relevance to psychopathology (Mansell, 2005). We are using PCT to model the jumping-to-conclusions bias (Dudley & Over, 2003) in probabilistic reasoning through the implementation of two control systems in conflict. Through differential weighting of the gains within these systems, the system can simulate every possible outcome demonstrated by individual data.

In a second project, a control system model will be used to simulate a model of bipolar disorder (Mansell, Morrison, Reid, Lowens, & Tai, 2007). The computer model will be designed to illustrate how mood swings can develop from two control systems in conflict, and that reorganization of the weighted outputs from a higher level system to these conflicted systems and a separate non-conflicted system can reduce mood swings, simulating successful recovery.

Further work in development includes working with a control systems engineer to model reorganization using MATLAB, using PCT model conflicting goals over access to psychological therapy, the development of sensimotor control within children with learning disabilities, and an online interactive therapy that promotes reorganization of higher level goals.

References

Bourbon, W. T. (1996). On the accuracy and reliability of predictions by Perceptual Control Theory: Five years later. *The Psychological Record, 46*, 39-47.

Carey, T. A. (2008). Perceptual Control Theory and the Method of Levels: Further contributions to a transdiagnostic perspective. *International Journal of Cognitive Therapy, 1,* 237-255.

Dudley, R.E.J. & Over, D.E. (2003). People with delusions jump to conclusions: a theoretical account of research findings on the reasoning of people with delusions. *Clinical Psychology & Psychotherapy, 10,* 263-274.

Mansell, W. (2005). Control Theory and Psychopathology. An Integrative Approach. *Psychology and Psychotherapy: Theory Research and Practice, 78,* 141-178.

Mansell, W., Morrison, A. P., Reid. G., Lowens, I., & Tai, S. (2007). The interpretation of and responses to changes in internal states: an integrative cognitive model of mood swings and bipolar disorder. *Behaviour and Cognitive Psychotherapy, 35,* 515-541.

Marken, R. S. (2001). Controlled variables: Psychology as the center fielder views it. *The American Journal of Psychology, 114, 259*-281.

Marken, R. S., & Powers, W. T. (1989). Random walk chemotaxis: Trial and error as a control process. *Behavioural Neuroscience, 103,* 1348-1355.

Powers, W. T. (1973; 2005). *Behavior: The Control of Perception.* Benchmark Publications.

Powers, W. T. (2008). *Living Control Systems III: The Fact of Control.* Benchmark Publications

Powers, W. T., Clark, R. K., & McFarland, R. L. (1960). A general feedback theory of human behavior. Part II. *Perceptual and Motor Skills, 11,* 309-323.

Runkel, P. J. (2003). *People as Living Things: The Psychology of Perceptual Control.* Hayward, CA: Living Control Systems Publishing. www.livingcontrolsystems.com. Permission for illustration granted by W. T. Powers on behalf of Mary Powers.

Capturing and Modeling Human Cognition for Context-Aware Software

Lin Mei (lmei@cs.toronto.edu)
Steve Easterbrook (sme@cs.toronto.edu)
Department of Computer Science, University of Toronto
Toronto, M5S 2E4 CANADA

Keywords: context-aware computing; context modeling.

Introduction

As computing becomes ubiquitous, it is possible for systems to sense their context of use and adapt their behavior accordingly. Using an appropriate context model that relates the users' cognitive contexts with specific activities can make ubiquitous computing systems more convenient and effective for their users. Recent work has explored the use of structural models for representing, sharing and reasoning complicated, dynamic and interrelated context information, e.g. Context Toolkit (Dey, Salber, & Abowd, 2001) , extended Object Role Model(ORM) (Henricksen, Indulska, & Rakotonirainy, 2003), and ontology-based context models (Strang, Linnhoff-Popien, & Frank, 2003; Serrano, Serrat, & Galis, 2006). However, human activities and preferences tend to be diverse and dynamically changing, depending upon material and social circumstances. Current context models usually concentrate on the computational representations of contexts which can be tracked and recorded, but ignore cognitive properties that are essential to human activities and decisions.

Theories from sociology and philosophy, especially ethnomethodology and phenomenology, suggest that user experience, such as subjective perception of system features and past experience of similar contexts, may influence current activity (Dourish, 2004). Ignoring human cognition in context analysis is therefore likely to frustrate and disorient users. In this paper, we present a cognitive context modeling framework that analyzes the diversity and dynamics of context-aware behavior by capturing and representing human cognition of context information, from objective settings, explicit user activities, to implicit user preferences, for a given task.

The Cognitive Context Modeling Framework

Context, according to Dewey (Dewey, 1960), has two components: 1) background, which is both spatial and temporal and is ubiquitous in all thinking; 2) selective interest, which conditions the subject matter of thinking. Computational tasks operate in a set of contexts, and the selection of contexts for monitoring and sensing is subject to computational, technical, and social constraints. Therefore, we classify the context information of a task into two major categories: *Objective Context(ObjCt)* and *Cognitive Context(CogCt)*. *ObjCt* refers to the contemporary settings within which a course of action emerges or the objective state of an activity, e.g., who, what, when, and where, which can be automatically sensed with a certain level of accuracy; while *CogCt* refers to a set of beliefs belonging to an individual or a community, e.g., purposes and preferences, which answers ”*why*” a piece of information should be considered as ”context” and ”*how*” it affects the result.

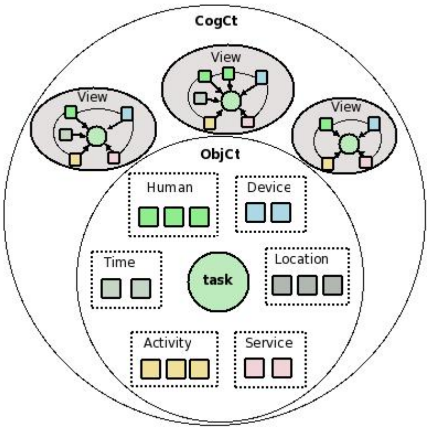


Figure 1: The Framework of Cognitive Context Model

Figure 1 shows our framework for modeling human cognition on a context-aware task. There are three key components in this framework: *task*, *objective context(ObjCt)* and *cognitive context(CogCt)*. An *task* can be interpreted as a flow of operations for transforming the object into an outcome. The process of the transformation, e.g., when and where to execute which operation, is affected by the state of a set of *ObjCt*s. The *ObjCt*s for a task are the detectable surroundings during the task process, e.g., time, location, device, etc. The cognitive selection of ”interesting” $ObjCt$s of a task is specified in the *CogCt* component by context *views* drawn from end-users or communities. Each *view* contains a set of $ObjCt$s, which are relatively ranked according to their relevance to the $task$.

Case Study: Power Saving Schedule

The cognitive context model structures the representation of task and its contexts from end-users' perspective. We conducted case study on a power saving task to illustrate the use of cognitive context model for context identification and analysis.

Shutting down the computer when it is not in use contributes to energy saving. Figure 2 shows a cognitive context model for the power saving task, in which each plot line represents one context view, and for each view, the x-axis represents the time contexts and the y-axis represents their relevance to the power-off state. The model is built by moni-

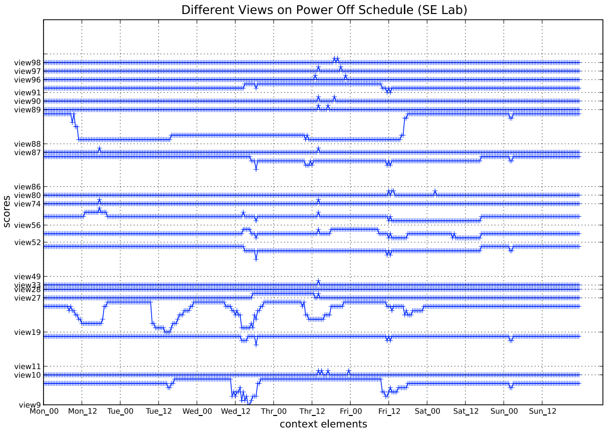


Figure 2: The diversity of context views on power saving

toring the power state of desktop computers used by staff and graduate students in our department, and the score is calculated by the power-off ratio (the total number of power-offs divided by the total number of observations).

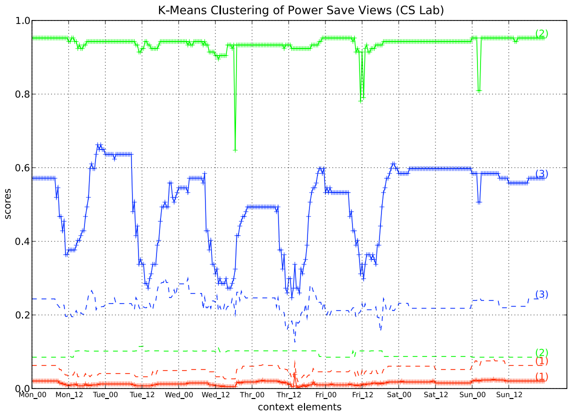


Figure 3: K-means clustering on context views (k=3)

The model exhibits the diversity of context views on the power saving task, i.e., the power state varies with time and views. The structure and data provided by this model also allow numerical analysis on the variance of human cognition and context-aware behavior. Figure 3 shows the result of K-means clustering we conducted on the context views in the model. With clustering, the context views are classified into three categories (the '+' lines): (1) almost never power-off, (2) always power-off, and (3) power state varies with time. The dashed lines in the figure represent the standard deviation values of each cluster. A low standard deviation value implies low diversity among context views in the cluster.

Since the context views in cluster(3) exhibit consciousness of power-saving activities, by assigning higher weights on these views, we generated an optimized context view for the power-saving task with AHP calculation (Saaty, 1994), as shown in Figure 4. The y-axis value indicates the relative importance/relevance of each time context to the power-off state. The optimized view integrates all the context views in cognitive context model and optimizes the relevance value of each context element, which can thus be used as an input to adaptation engine for context-aware task execution and reconfiguration.

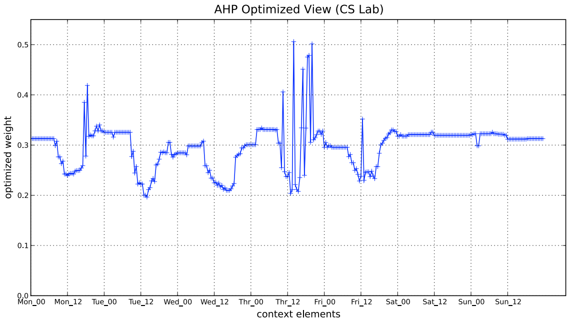


Figure 4: Optimized context view

Conclusions

This paper presents a cognitive context modeling framework for capturing and analyzing end-users' cognition of context-aware behavior. The performance of this framework is illustrated with a case study on computer power saving. It shows that with cognitive context modeling, various context views of a given task can be captured and visualized, which provides efficient support on checking the variance of human cognition and reducing bias in decision making.

References

Dewey, J. (1960). Context and thought. In R. Bernstein (Ed.), *On experience, nature, and freedom* (p. 95). Bobbs-Merrill.

Dey, A., Salber, D., & Abowd, G. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, *16*, 97-166.

Dourish, P. (2004). What we talk about when we talk about context. *Personal Ubiquitous Comput.*, *8*(1), 19–30.

Henricksen, K., Indulska, J., & Rakotonirainy, A. (2003). Generating context management infrastructure from context models. In *Mdm'03: International conference on mobile data management.*

Saaty, T. (1994). *Fundamentals of decision making and priority theory with the analytic hierarchy process.* RWS Publications.

Serrano, J. M., Serrat, J., & Galis, A. (2006). Ontology-based context information modelling for managing pervasive applications. In *Icas '06: International conference on autonomic and autonomous systems.*

Strang, T., Linnhoff-Popien, C., & Frank, K. (2003). Cool: A context ontology language to enable contextual interoperability. In *Dais'03:international conference on distributed applications and interoperable systems.*

Multiple Methods of Modeling and Detecting Perceptual and Cognitive Configurality

Tamaryn Menneer (T.Menneer@soton.ac.uk)
School of Psychology, Shackleton Building
Highfield Southampton SO17 1BJ UK

Michael J. Wenger (mjw19@psu.edu)
Department of Psychology, 216 Moore Building
University Park PA 16802 USA

Leslie Blaha (lblaha@indiana.edu)
Department of Psychological and Brain Sciences, 1101 East 10th Street
Bloomington, IN 47405 USA

Keywords: perceptual organization, face perception, configurality, holism, statistical methods

Introduction

A central question in perceptual and cognitive psychology is the nature of the processes that combine the multiple sources of environmental information in order to support the subjective, unitary percepts of objects. Characterization of these processes, at a general level, has been a goal of perceptual and cognitive sciences for more than a century. One of the more promising extant approaches is known as general recognition theory (GRT; Ashby & Townsend, 1986). GRT provides formal, mathematically-specified definitions of the ways in which perceptual dimensions (e.g., the various elements of a face) can interact during perception and identification, and generalizes the signal-detection-based distinction between perceptual and decisional effects to multidimensional stimuli.

Mean-Shift Integrality

One situation in which information about the internal perceptual and decision spaces may be difficult to obtain is when a mean-shift occurs in a set of stimuli (see, e.g., Maddox & Ashby, 1992). This particular situation is one in which a change in level of one dimension of a multidimensional stimulus results in a shift of the internal representation such that all relative distances are preserved (Figure 1). Here the equal-likelihood contours for bivariate distributions 2 and 4 have been shifted upwards yet their relationship is identical to that for distributions 1 and 3. Here, the distance (or marginal *d'*) between distributions 2 and 4 is the same as that between distributions 1 and 3, and for both pairs, the decision bound lies in the same relative location. In such a situation, the decision bound can shift in a piecewise (left panel) or continuous (right panel) manner. A continuous decision bound gives rise to correlations in observable responses to the two dimensions: as the evidence pertinent to dimension *x* increases (for example responses to *y* increasingly fall below the decision bound (see Figure 1).

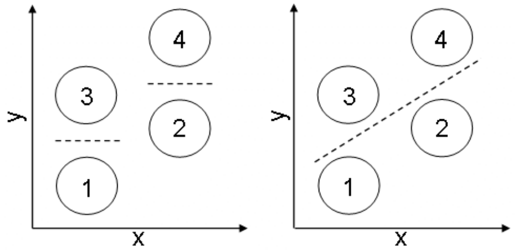


Figure 1: Illustration of mean-shift integrality. Left panel: Piecewise decision bound. Right panel: Continuous decision bound.

Mean-shift could arise if changing the level of one dimension produces a complete shift in the information for the other dimension, as has been hypothesized to be the case in a number of classic phenomena in face perception (e.g., the Thatcher illusion). This possibility has been acknowledged to be problematic for standard methods of estimating the nature of the perceptual space from standard behavioral data (Ashby & Townsend, 1986; Thomas, 2001). In addition, the ability to identify the presence of a mean shift takes on substantive importance in experimental contexts that rely on the diagonal distances within the perceptual evidence space (e.g., the difference between the two diagonal distances as an index of holistic processing. In such situations, it is easily possible to arrive at erroneous conclusions regarding either these diagonal distances or the more-standard marginal distances, absent critical additional converging evidence for the nature of the perceptual space (as provided in, e.g., Kadlec & Hicks, 1998; Wenger & Ingvalson, 2003).

Multiple Measures

The present project was intended as a first step in assessing the extent to which it is possible to reduce inferential errors in the context of mean-shift integrality by augmenting standard methods with statistical methods that to date have not been applied to this problem. The standard approach is based on a set of marginal measures of sensitivity and bias, drawn from classic signal detection theory (Ashby & Townsend, 1986; Kadlec & Townsend, 1992), and these approaches are shown to have elevated Type II error rates in identifying the conditions for mean-shift integrality.

The first of the novel methods is an approach developed by DeCarlo (2003) in which probit models are used to determine signal detection measures in multidimensional stimulus space. For each distribution, in each dimension, a probit model is implemented: $y^* = \beta x + \mu$, where y^* is the dependent variable (response) and x is the explanatory variable (correct response). Such models can be used to determine *d'* (from β), *c* (the threshold for the outcome, y^*) and bivariate correlations (revealed in the residuals, μ).

The second novel method is drawn from methods of categorical data analysis, specifically estimates of polychoric and tetrachoric correlations based on response frequencies. In particular, we assume that the entire response space is sampled from a bivariate normal distribution with a single response threshold on each of the two dimensions. Given a 2x2 response contingency table, we estimate both the tetrachoric correlation and response thresholds with maximum likelihood estimation (Olsson, 1979). A mean-shift will give rise to correlation between response frequencies because the change in decision bound causes systematic variation, particularly in response frequencies from diagonally opposite distributions. These correlations will result in a non-zero tetrachoric correlation within the bivariate distribution of the response space.

Approach and Outcomes

We evaluated these methods using simulated data sets, representing the absence and presence (in varying magnitude) of mean shifts. Using the sets of measures, alone and in all possible combinations, we estimated the relative frequency of inferential errors. The standard approach, as expected, produced regular inferential (Type II) errors in the presence of mean-shift integrality. However, when augmented with the new methods, the rate of such errors was substantially decreased.

Like the traditional approach, the probit model provides relative information about marginal *d'*s and *c*s. The outputs from each approach are largely in agreement, barring a conservative bias for bivariate correlations in the traditional method and a liberal bias for differences in *d*'s in the probit models. Because both methods provide relative estimates they cannot be used to detect mean-shift integrality directly. However, unlike the traditional approach, the probit model method includes the estimation of residuals, which allows a direct test for bivariate correlations within the underlying perceptual distributions. When a mean-shift in distributions is accompanied by a continuous shift in the decision bound, the probit models identify bivariate correlations of the same sign and similar magnitude across all distributions. They also identify any shift in decision bound relative to the distributions. Such evidence of mean-shift indicates a need for a direct test using the polychoric correlation method.

Tetrachoric correlations applied to data sampled from mean-shift distributions accompanied by a continuous decision bound shift revealed significant non-zero correlations in the response space when the mean-shift is of medium to large magnitude. These estimates are sensitive to the magnitude of the mean shift, and inferential errors based on correlation alone increase as mean-shift magnitude decreases.

Neither the probit nor the polychoric correlation approaches can identify a piecewise shift in the decision bound, because relative locations are maintained, such that response frequencies remain unaffected and do not exhibit correlations. The results suggest that, although mean-shift integrality can pose a serious inferential challenge, a multi-measure approach can reduce the potential for inferential errors. As such, the approach is consistent in spirit with the original (Ashby & Townsend, 1986) multi-measure approach to estimating GRT models.

Acknowledgments

Thanks are due to Jim Townsend, Robin Thomas, Noah Silbert, Danny Fitousi, Jennifer Bittner, Rebecca Von Der Heide, Tom Palmeri, and Isabel Gauthier for helpful stimulating discussions of these issues.

References

Ashby, F. G., & Townsend, J. T. (1986). Varieties of perceptual independence. *Psychological Review. 93,* 154–179.

DeCarlo, L. T. (2003). Source monitoring and multivariate signal detection theory, with a model for selection. *Journal of Mathematical Psychology, 47*, 292-303.

Kadlec, H., & Hicks, C. L. (1998). Invariance of perceptual spaces and perceptual separability of stimulus dimensions. *Journal of Experimental Psychology: Human Perception and Performance, 24*, 80–104.

Kadlec, H., & Townsend, J. T. (1992b). Signal detection analysis of dimensional interactions. In F. G. Ashby (Ed.), *Multidimensional models of perception and cognition* (pp. 181–228). Hillsdale, NJ: Erlbaum.

Olsson, U. (1979) Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika, 44* (4), 443-460.

Thomas, R. D. (2001). Characterizing perceptual interactions in face identification using multidimensional ignal detection theory. In M. J. Wenger & J. T. Townsend (Eds.), *Computational, geometric, and process perspectives on facial cognition: Contests and challenges* (pp. 193–228). Mahwah, NJ: Erlbaum.

Wenger, M. J., & Ingvalson, E. M. (2003). Preserving informational separability and violating decisional separability in facial perception and recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29*, 1106-1118.

A Preliminary ACT-R Compiler in Herbal

Jaehyon Paik (jaehyon.paik@psu.edu)
Department of Industrial and Manufacturing Engineering
The Pennsylvania State University, University Park, PA 16802 USA

Jong W. Kim (jongkim@psu.edu)
Frank E. Ritter (frank.ritter@psu.edu)
College of Information Science and Technology
The Pennsylvania State University, University Park, PA 16802 USA

Herbal (Haynes, Cohen, & Ritter, 2009) represents human behavior based on the Problem Space Computational Model (Newell, Yost, Laird, Rosenbloom, & Altmann, 1991). The PSCM is a theory of cognition that defines human behavior as movement through problem spaces using operators. To represent models, Herbal uses XML as basis forms of high-level language, and Herbal compiles them into low-level rule-based representations that execute within a cognitive architecture, Soar, and intelligent agent architecture, Jess. Users can create them with a GUI or directly in XML.

Herbal is implemented as an Eclipse plug-in, which provides a popular graphical development environment. It uses Eclipse's powerful functions for creating and maintaining agents, so it enables model creators to make models more easily.

We have started to create an ACT-R compiler in Herbal because of these features. Although several easy to use frameworks exists to develop ACT-R models, such as ACT-Simple (Salvucci & Lee, 2003), and G2A (St. Amant & Ritter, 2004), they cannot represent more than KLM-GOMS or GOMS models.

Matching the Herbal Components with ACT-R Components

We developed the ACT-R compiler based on the Jess compiler, because the Jess compiler compiles into declarative knowledge and procedural knowledge, and its output has a Lisp-like syntax similar to ACT-R. The current version of Herbal has 6 basic components, Agent, Problem Space, Operator, Condition, Action, and Type. These can all be mapped onto ACT-R components as shown in Table 1, which includes their Jess correspondences as well.

Table 1: Herbal components and their implementation in Jess and ACT-R.

Herbal	Jess	ACT-R
Agent	Agent	Model
Problem Space	Defmodule	Slot of Goal buffer
Operator	Defrule	Production
Condition	Condition of defrule	Condition of rule
Action	Action of defrule	Action of rule
Type	Deftemplete	Chunk-type
- Field	- Slot	- Slot

We added an additional component, called *Declarative Memory,* in the Herbal environment. With this component, users can represent hierarchical or sequential tasks in an ACT-R model easily. The *Declarative Memory* consists of 6 components: library, element name, parent name, first child name, next sibling name, and action name. Through these components, users can layout their whole task hierarchically or sequentially, and the relations among tasks are shown in a tree form in the bottom of the user interface. Based on these relationships, the productions are made by ACT-R compiler.

To explore the flexibility of this high-level compiler approach, we added a user expertise compiler flag to Herbal. It leads to compiling either a novice or an expert user model. The expert model does not retrieve declarative memory items when it executes subtasks. However, the novice model retrieves declarative memory items to move to the next task step according to the goal hierarchy in declarative memory. Figure 1 shows the difference between Expert and Novice model.

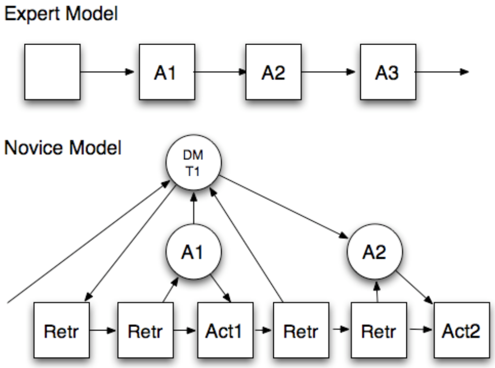


Figure 1: The Structure of the Expert and Novice model.

Experiment

We use a simple dialing number task to show a simple model. This task is decomposed into a set of hierarchical subtasks to dial each component and then the numbers in each component. It consists of four subtasks: Long Distance Code, Area Code, Exchange, and Extension. Each subtask has its own subtasks (the buttons to press), and all of these subtasks are related with other tasks and subtasks as a parent, child, or sibling.

We chose to dial 1 (814) 865–4455, so the Long Distance has a subtask, press-1, and the Area Code has subtasks, press-8, press-1, and press-4. The Exchange and Extension have similar subtasks. So, the total number of tasks is 11 leaf nodes, 4 sub nodes, and one task node, and these tasks are stored as 16 declarative memory type nodes in the Herbal environment.

Using the ACT-R compiler with the user expertise compiler flags, we generated expert and novice models. The expert model produces the next task without retrieving a declarative memory using 17 rules and 16 chunks, however, the novice model retrieves a declarative memory to produce the next task using 8 rules and 16 chunks.

We simulated 10 trials per model to get mean prediction times. The default ACT-R parameters were used (these are carried in the compiler). The expert model's predicted times, shown in Table 2, did not have any variance with the ACT-R default values, however, there exist differences among the trials in the novice model. In addition to the ACT-R cognitive modules' times, we added keystroke times (typing random letter) to each model to get total predicted time (we do not yet use ACT-R/PM for motor output).

For comparison, this task was analyzed using the KLM-GOMS theory (Card, Moran, & Newell, 1983). For the keystroke operator, we use the same time of "typing random letter", 0.50 s. The number of keystrokes is 11. Thus, the total time spent in key stroking is 5.5 s (as used above). For each mental operator, we use the default preparation time (T_M) of 1.35 s. A user mentally prepares what numbers to press, what to retrieve from memory, and what to do for the next step. In this task, a mental preparation for each subtask was counted: Long Distance Code, Area Code, Exchange, and Extension. Thus, the total time spent in mental preparation is 5.4 s. Therefore, the total execution time from the KLM is 10.9 s ($T_{execute} = T_K + T_M = 5.5 + 5.4$). Table 2 shows above result with respect to prediction time, and the number of rule firings.

Table 2: The mean, standard deviations of prediction time, and the number of rule firings in each model, and KLM model for the simple dialing number task (*N*=10).

	KLM	Novice	Expert
Mean	10.9 s	13.48 s	6.35 s
SD	0 s	0.79 s	0 s
Rule firings	-	20	16

The Herbal/ACT-R novice model is a bit slow compared to the KLM predictions, as it should be. The Herbal/ACT-R expert model is a bit fast. It is the case, that the Herbal ACT-R compiler makes different predictions across the expert and novice models, and it may be the case that subjects when they perform this task are best represented by a model between these two extremes, or by a distribution of user models, as John (1996) proposed .

Discussion and Conclusion

We started to develop an ACT-R compiler and declarative memory component in the Herbal environment. This compiler takes knowledge represented as a PSCM model in Herbal, and in addition to compiling it in Soar and in Jess, compiles it into ACT-R. This compilation process was tested and appears to show some promise for creating more sophisticated models more easily.

We added a declarative memory pane for representing hierarchical task analyses. This representation is not currently pretty, but allows users to represent tasks in a GOMS-like language. As part of this component, we included a way (a compiler flag) to generate both novice and expert models from the same knowledge set. The novice model accesses the declarative memory elements to generate behavior. The expert model is compiled so that the rules apply directly and keep the state on the goal. (This compiler flag is not yet used by the Soar or Jess compilers.)

The model of simple dialing number task was compared with a GOMS model with respect to predicted time. The GOMS model's prediction time is located between our expert and novice model. The novice model of this task fired 20 rules, and the expert model fired 16. If a task has more hierarchical levels in it, the number of rule firings between these two model types will be more different. Because this task has a hierarchical structure (3 levels), there was a noticeable difference.

References

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Haynes, S. R., Cohen, M. A., & Ritter, F. E. (2009). A design for explaining intelligent agents. *International Journal of Human-Computer Studies, 67*(1), 99-110.

John, B. E. (1996). TYPIST: A theory of performance in skilled typing. *Human-Computer Interaction, 11*(4), 321-355.

Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S., & Altmann, E. (1991). Formulating the problem space computational model. In R. F. Rashid (Ed.), *Carnegie Mellon Computer Science: A 25-Year commemorative* (pp. 255-293). Reading, MA: ACM-Press (Addison-Wesley).

Salvucci, D. D., & Lee, F. J. (2003). Simple cognitive modeling in a complex cognitive architecture. In *Human Factors in Computing Systems: CHI 2003 Conference Proceedings*, 265-272. ACM: New York, NY.

St. Amant, R., & Ritter, F. E. (2004). Automated GOMS to ACT-R model generation. In *Proceedings of the International Conference on Cognitive Modeling*, 26-31. Erlbaum: Mahwah, NJ.

Clustering and Traversals in Concept Association Networks

Arun R and **V Suresh** and **C E Veni Madhavan**
(arun_r, vsuresh, cevm)@csa.iisc.ernet.in
Department of Computer Science and Automation
Indian Institute of Science, Bangalore 560012, India

Abstract

We view association of concepts as a complex network and analyse its structure. We observe that concept association network is scale-free and has small-world properties. We also study two large scale properties of these networks — clusters and paths. First, we present an algorithm for clustering these networks which generate qualitatively better clusters than those generated by spectral clustering, a conventional mechanism used for graph partitioning. Next, we study paths generated by human traversals on these networks and contrast it with random walks and shortest distance paths. Our results are a first step towards viewing human cognitive abilities in the light of complex network analysis.

Concept Association Networks(CAN)

Concept associations can be intuitively understood as thoughts that occur in conjunction with each other. Typically, networks of such associations are built by presenting concepts to subjects and recording their output on the basis of '*what comes to your mind first*' in response to a cue. Such co-occuring cue-response concept pairs are considered to be *cognitively associated*.

Thus in a concept graph $G = (V, E)$, V the vertex set represents labelled nodes(concepts), and E the edge set represents co-occuring concepts. For our study we use word associations from USF Free Association Norm (http://w3.usf.edu/FreeAssociation) as it is more comprehensive than other databases and has also been studied earlier from a complex network prespective (Steyvers & Tenenbaum, 2005). Complex networks are graph abstractions to represent and analyse real world interacting systems like World-Wide-Web and social networks. A list of important structural properties of the concept network built based on this database is shown in table 1.

Table 1: Some salient network properties of CAN

nodes: 10618	avg degree: 12.01	max degree: 332
edges: 63788	edge density: 0.001131	
diameter: 7	$\gamma \sim 2.6$	CC: 0.1871

A power law degree distribution and high clustering coefficient(CC) are indicative of the similarity of concept association network to other widely studied complex networks such as World-Wide-Web, Social networks etc. (Albert & Barabási, 2002). Thus studying the properties of these concept interactions in the light of complex networks is justified. In this work we study two macro structres of concept networks, namely clusters –partitions of the network, and paths –traversals in the network, and relate its possible implications on cognition.

Clustering of Concepts - Algorithm

Clustering is an important aspect of generalization that helps in reducing intrinsically different things into broad groups for the sake of simplicity. Given that we learn concepts by relating to other similar concepts already known, it makes sense to cluster concept association networks into broader abstract entities. Such clusters would be useful if they can effectively represent human organisation of knowledge.

In this regard, we present a clustering algorithm and explain its usefulness in the context of cognition. We consider high degree hub nodes as the starting points. To begin with n hub nodes are labelled as belonging to its own cluster$\mathcal{C}_i$ ($i = 1$to n where $n = 10$ for this study). For each unlabelled node u in the graph its neighbourhood is explored to find the node with the highest degree v (say). If v is labelled, we assign the same label to u. If not, we perform the neighborhood exploration process on v. The recursion stops either when a hub node is hit or when no node with higher degree is present in the neighbourhood. In the former case, the node is assigned the hub node's cluster and in the latter, we assign it to a *default* cluster. Nodes assigned to the default cluster are finally assigned to the hub that is at the shortest distance in terms of path length. A stylistic version of the algorithm is given below.

```
$neigh(u)$:  Set of nodes formed by the immediate
neighbors of node $u$.  $deg(u)$:  degree of node $u$
$node_{degmax}(S)$:  node with max degree in the set of
nodes $S$.  $label(u)$:  label of node $u \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$
Init:  Identify $n$ hub nodes and label them $\mathcal{C}_1, \ldots, \mathcal{C}_n$
for each unlabelled node $u$
S1:  let $v \leftarrow node_{degmax}(neigh(u))$
  if $deg(v) \geq deg(u)$
    if $label(v) = \mathcal{C}_i$, then $label(u) \leftarrow \mathcal{C}_i$
    continue
    if $label(v) \neq \mathcal{C}_i, \ \forall \ i \in \{1, \ldots, n\}$
      then $u \leftarrow v$, GOTO S1
  else $label(u) \leftarrow \mathcal{C}_0$
```

Comparison and Discussions

A comparison between spectral clustering (Ng, Jordan, & Weiss, 2001) and and our algorithm is shown in Figure 1 as log-log plots of cluster degree distribution. It is clear from

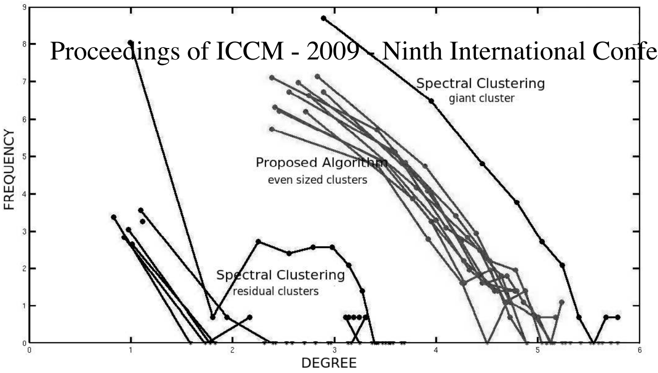


Figure 1: degree distribution of clusters: proposed Vs spectral

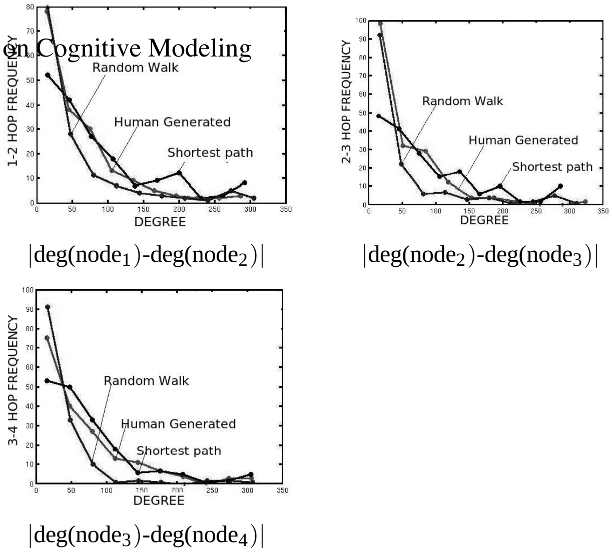
the figure that spectral clustering does not preserve scale-free characteristics within clusters. Moreover, the sizes of clusters are uneven. On the other hand, our algorithm splits the original graph into roughly equal sized clusters and each one has scale-free distribution with the same power-law exponent that applies to the whole graph. Thus the clusters from our approach are self-similar to the whole network. In effect our algorithm imparts a hierchical view to the whole network. This is in accordance with the the general hierarchical organisation of human knowledge.

Spectral clustering is a series of random walks to estimate cluster boundaries — walks are contained with in strongly connected components and rarely tend to take connecting bridges. One starts with various 'seeds' to begin the random walk and see the nodes that are reached eventually and thereby identify clusters. We believe that the difference in cluster properties between the two algorithms is because random walks are an unnatural means to navigate the cognitive space. This is further explored in the next section.

Concept Traversals - Observations

There are two extremes to (source,target) traversals: Shortest path from the source to the destination and Random walk starting from the source and proceeding till the target is reached. To quantify the properties of human generated paths[1], we compare them with both these extremes. It is intuitively clear that human generated paths must lie in the middle of these two strategic extremes. We identify a non-trivial property –the difference in degree of adjacent nodes in the paths– to offer a formal explanation for this intuition.

Figure 2 shows the difference of successive degrees of first two edges for shortest, random and human paths. Shortest paths show a steeper degree difference whereas for the random walk, the degree differences are smaller than those from human paths. The rationale for this is as follows.



$|\text{deg}(\text{node}_1)\text{-deg}(\text{node}_2)|$

$|\text{deg}(\text{node}_2)\text{-deg}(\text{node}_3)|$

$|\text{deg}(\text{node}_3)\text{-deg}(\text{node}_4)|$

Figure 2: distribution of absolute degree difference

Given the structure of the concept graph –high CC implying dense neighbourhoods– degrees of successive nodes are expected to be similar. On such graphs random walks typically spends longer durations 'dwelling' in concept neighbourhoods rather than reaching the destination. Whereas shortest paths make incoherent conceptual jumps to reach the destination. In comparison, human traversals are a mix of smooth transitions and conceptual leaps and lie in the middle of these two extremes.

Conclusions

We proposed a clustering algorithm that exploits the structural properties of concept association network to produce self similar clusters that are arguably better than those produced by conventional clustering approaches. Then we compared concept traversals for human, random and shortest paths and quantified their differences in terms of the degree difference of adjacent nodes present in such paths. We observed that this network property can explain the intuitive idea that human paths are inbetween random and shortest paths –cogent yet amenable to conceptual leaps.

References

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*, 47.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems 14* (pp. 849–856). MIT Press.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic network: Statistical analyses and model of semantic growth. *Cognitive Science*, *29*(1), 41-78.

[1] For our analysis, we asked 60 participants to perform concept traversals from source to targets for 183 concept pairs like (POWER,MONTH), (FAMILY,AREA) etc. Our observations are based on these subject generated paths.

A Comparison of the performance of humans and computational models in the classification of facial expression

A.Shenoy[1] (a.1.shenoy@herts.ac.uk)
Sue Anthony[2] (s.1.anthony@herts.ac.uk),
R.J.Frank[1] (r.j.frank@herts.ac.uk),
Neil Davey [1] (n.davey@herts.ac.uk)
[1] Department of Computer Science, [2] Department of Psychology,
University of Hertfordshire, College Lane, Hatfield
AL10 9AB

Keywords: Facial Expressions, Image Analysis, Classification, Reaction Time.

Abstract

Recognizing expressions are a key part of human social interaction, and processing of facial expression information is largely automatic for humans, but it is a non-trivial task for a computational system. In the first part of the experiment, we develop computational models capable of differentiating between two human facial expressions. We perform pre-processing by Gabor filters and dimensionality reduction using the methods: Principal Component Analysis, and Curvilinear Component Analysis. Subsequently the faces are classified using a Support Vector Machines. We also asked human subjects to classify these images and then we compared the performance of the humans and the computational models. The main result is that for the Gabor pre-processed model, the probability that an individual face was classified in the given class by the computational model is inversely proportional to the reaction time for the human subjects.

Introduction

In this work we compare the performance of human subjects classifying facial expressions, with the performance of a variety of computational models. We use a set of 176 face images, half of which express anger and the other half have a neutral expression. The images are from the BINGHAMTON BU-3DFE database (Yin, Wei et al. 2006) and some examples are shown in Figure 1.

Pre-Processing Methods and Classification

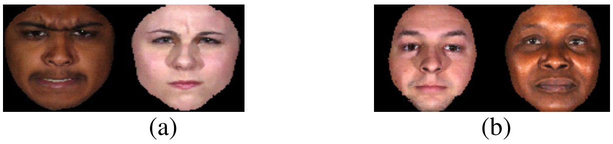
This section describes how the computational model classifies angry faces and neutral faces. High dimensional data such as face images are often reduced to a more manageable low dimensional data set. We perform dimensionality reduction using both Principal Component Analysis (PCA) and Curvilinear Component Analysis (CCA). PCA is a linear projection technique but it may be more appropriate to use a non linear Curvilinear Component Analysis (CCA) (Demartines and Hérault 1997). Gabor filters are also often used for extracting features of images, and they are thought to mimic some aspects of human visual processing (Daugman 1985). Classification is performed using a Support Vector Machines (SVM). An SVM performs classification by finding the maximum margin hyper-plane in a feature space. The relative distance of an instance from this hyper-plane can be interpreted as its probability of belonging to the appropriate class. We have used the LIBSVM-2.86 tool (Chang and Lin 2001).

Experiment

Two sets of experiments were performed. Part A - Computational models. Part B - Classification performed by human subjects.

Part A- Computational Models

The data was divided into four subsets, and training/testing took place with a leave one out strategy, so that results are averages over four independent runs. Once a training set had been selected the two parameters of the SVM were optimized by cross-validation. Six variations of data processing are tested as detailed in Table 1.



(a) (b)

Figure 1: Example face images. a) Angry b) Neutral

Computational Model Results

For PCA, the first 97 components of the raw dataset and 22 components in the Gabor pre-processed dataset account for 95% of the total variance. For CCA, we reduce the data to its Intrinsic Dimension. The intrinsic dimension of the raw faces was approximated as 5 and that of the Gabor pre-processed images was 6.

The results in Table 2 indicate the overall classification accuracy is not very good; however, classifying angry faces is a difficult task for computation models (Susskind 2007) and can be seen from the results. Nevertheless, the SVM performs well with an average of 84.09% accuracy with raw face images

.

Table 1: Types of Computational Models

Name model	Type of Input	Dimensionality Reduction
Model 1	Raw faces	None
Model 2	Raw faces	PCA
Model 3	Raw faces	CCA
Model 4	Gabor pre-processed	None
Model 5	Gabor pre-processed	PCA
Model 6	Gabor pre-processed	CCA

Table 2: SVM classification Results

Accuracy	TEST SET 4	TEST SET 3	TEST SET 2	TEST SET 1	Average
Model 1	79.54% (35/44)	93.18% (41/44)	79.54% (35/44)	84.09% (37/44)	84.09%
Model 2 (PCA97)	68.18% (30/44)	77.27% (34/44)	70.45% (31/44)	65.91% (29/44)	70.45%
Model 3 (CCA5)	68.18% (30/44)	59.09% (26/44)	63.64% (28/44)	63.64% (28/44)	63.64%
Model 4	68.18% (30/44)	79.55% (35/44)	72.73% (32/44)	81.82% (36/44)	75.57%
Model 5 (PCA22)	61.36% (27/44)	79.55% (35/44)	75% (33/44)	72.73% (32/44)	72.16%
Model 6 (CCA6)	63.64% (28/44)	70.45% (31/44)	68.18% (30/44)	63.64% (28/44)	66.48%

Part B - Human subjects

The 184 raw images were used in this experiment. Twenty individuals took part in the study.

Method

A total of 16 images were used in the pre-view block and the remaining 168 images were divided into 6 balanced blocks of 28 images each. We used a tool called as TESTBED (Taylor 2003) which is a response test generator program to record the classification and the Response Time (RT) of individuals.

Human Subject Results

Humans correctly classified the target expression with a mean of 82.86% (SD = 0.174) and the average RT was 1.132 seconds (SD = 0.714). The average RT ranges between a maximum value of 1.792sec and a minimum value of 0.714sec.

Discussion

We use the Bi-Variate Correlation to find any correlation between the average RT for human subjects and the class membership probability for the computational models. The results are considered to be significant at the level of 0.05, or below. The results of comparison are shown in correlation matrix of Table 3.

Table 3: The Bi-Variate Correlation Results

Model	Correlation value	Significance value
Model 1	-0.005	0.391
Model 2	+0.002	0.645
Model 3	-0.022	0.126
Model 4	-0.045	0.016
Model 5	-0.028	0.065
Model 6	-0.003	0.597

Interestingly all but one of the correlations are negative, but only for Model 4 (Gabor filtered images with no dimensionality reduction) was this correlation significant, with the probability of the null hypothesis being 0.016. The correlation is negative with value -0.045. This negative correlation indicates large average RT (which presumably indicates that the subjects found it hard to classify), correlates with smaller class membership probability for the model. The results are interesting and encouraging (suggestive of Gabor filtering is similar to human face processing) and our next step is extending these experiments to other expressions.

References

Chang, C.-C. and C.-J. Lin (2001). "LIBSVM: a library for support vector machines."

Daugman, J. G. (1985). "Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two dimensional visual cortical filters." *Journal of Optical.Society of.America .A* **2**(7).

Demartines, P. and D. J. Hérault (1997). "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets " *IEEE Transactions on Neural Networks* **8**(1): 148-154.

Susskind, J. M., G. Littlewort, M.S. Bartlett, J. Movellan,A.K. Anderson((2007). "Human and computer recognition of facial expressions of emotion." *Neuropsychologia* **45**(1).

Taylor, N. (2003). Developing with Authorware- Test bed. *ATSiP Conference at the University of Hertfordshire*

Yin, L., X. Wei, et al. (2006). A 3D Facial Expression Database For Facial Behavior Research. *7th International Conference on Automatic Face and Gesture Recognition* (FGR06).

Proceedings of ICCM - 2009 - Ninth International Conference on Cognitive Modeling

On Modelling Typical General Human Behavior in Games

Rustam Tagiew (tagiew@informatik.tu-freiberg.de)
Institute for Computer Science, Freiberg, Germany

Abstract

The question addressed in this work is 'What do people exactly typically do, if they interact strategically in games they have not much experience with?'. It is certain that human behavior in strategic interactions and games deviates from predictions of game theory. But, it is also certain that this behavior must have some kind of explanation. Eventually, people do not behave in a fully unpredictable way. This work considers general strategic interactions with untrained subjects. It does not consider human performance in well-known games like chess or poker. A very basic scenario is used to investigate human behavior. This scenario is a repeated zero sum game with imperfect information. An experiment with subjects is conducted and the data is analyzed using a set of different machine learning algorithms. As the result, a way of using machine learning is given. Finally, designing a formalism for representing human behavior is discussed.

Keywords: Game Theory, Data Mining, Artificial Intelligence, Domain-Specific Languages

Introduction

Typical human behavior in games is not optimal and deviates from game theoretic predictions (F.Camerer, 2003). Conceivable reasons are the bounded computational resources and the (seeming) absence of rationality. One can say without any doubt that if a human player is trained in a concrete game, he performs close to optimal. But, a chess master does not also play poker perfectly and vice versa. On the other side, a game theorist can find a way to compute an equilibrium for a game, but it does not make a successfull player out of him. For most of games, we are not trained. That is why it is more important to investigate our behavior in general game playing than game playing in concrete game.

This work is about the common human deviations from predicted equilibria in games, for which they are not trained. Modeling typical human behavior in general games needs a represention formalism which is not specific to a concrete game. An example-driven development of such a formalism is the challenge addressed in this paper. The example introduced in this work are repeated two-player zero-sum games with no pure strategy equilibria (Tagiew, 2009). Each player has a couple of actions called strategies. The solution of such games is to use mixed strategy equilibrium (MSE). An MSE is defined through a distribution over strategies, according to which the strategies are to be chosen.

The related works (Gal & Pfeffer, 2007) and (Marchiori & Warglien, 2008) use following approach. First, they construct a model, which is based on theoretical considerations. Second, they adjust the parameters of this model to the experimental data. This makes the human behavior explainable using the concepts from the model. On repeated zero sum games with more than two strategies, the correctness does not exceed 45% for all evaluated models.

Results

The seven evalutated games are related to paper-scissors-stone and have at least one MSE. The games denoted through IDs 31 till 61 have the following MSE solutions - 31$\Rightarrow$ $\{(\frac{1}{3},\frac{1}{3},\frac{1}{3})\}$, 41$\Rightarrow$ $\{(0,\frac{1}{2},0,\frac{1}{2}),(\frac{1}{2},0,\frac{1}{2},0)\}$, 51$\Rightarrow$ $\{(\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{3},\frac{1}{3})\}$, 52$\Rightarrow$ $\{(\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{2}{7},\frac{2}{7})\}$, 53$\Rightarrow$ $\{(\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{2}{7},\frac{2}{7})\}$, 54$\Rightarrow$ $\{(0,\frac{1}{2},0,\frac{1}{2},0),(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0)\}$ and 61$\Rightarrow$ $\{(0,\frac{1}{2},0,\frac{1}{2},0),(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0)\}$.

This work follows the so called *black box* approach. The black box in this case is the human player. The input are the game rules and previous decisions of players. The output is the current decision. Finding a hypothesis which matches the behavior of the black box is a typical problem called supervised learning (Mitchell, 1997). There is already a big amount of algorithms for supervised learning. Each algorithm has its own hypothesis space. For a Bayesian learner i.e., the hypothesis space is the set of all possible Bayesian networks. There are many different types of hypothesis spaces - rules, decision trees, Bayesian models, functions and so on. A concrete hypothesis is a relationship between input and output described by using the formal means of the corresponding hypothesis space.

Which hypothesis space is most appropriate to contain valid hypotheses about human behavior? That is a machine learning version of the question about a formalism for human behavior. The most appropriate hypothesis space contains the most correct hypothesis for every concrete example of human behavior. A correct hypothesis does not only perform well on the given data (training set), but it performs also well on new data (test set). Further, it can be assumed that the algorithms which choose a hypothesis perform alike well for all hypothesis spaces. This assumption is a useful simplification of the problem for a preliminary demonstration. Using it, one can consider the algorithm with the best performance on the given data as the algorithm with the most appropriate hypothesis space. The standard method for measurement of performance of a machine learning algorithm or also a classifier is cross validation.

The data of the experiment is transformed to sets of tuples for every game. Every tuple has the length $3+3+1=7$ (3 last pairs of turns and current turn). The size of a set is 540 tuples for games 31 till 53 and 340 for game 61. Implementations of classifiers provided by WEKA (Witten & Frank, 2005) are used for the cross validation on the sets of tuples. The task is to find a relationship between the last three players's decisions (6 items) and the current decision. There are 45 classifiers available, which can handle multi-valued nominal classes. Strategies in games are nominal, because there is no order between them. A cross validation of all 45

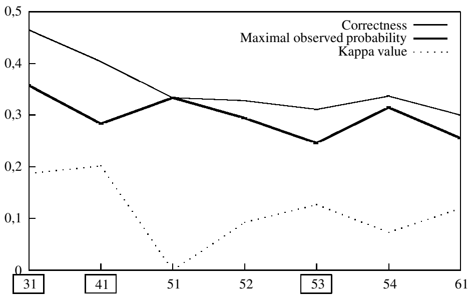


Figure 1: Average correctness in cross validation.

classifiers on all 7 sets of tuples is performed. The number of subsets for crossvalidation is 10.

There is no classifier which performs best on all games. Further even the highest average correctness is very low. Fig.1 shows the results. The gap between the highest observed probability of a strategy and the highest average correctness is different depending on the game. The Kappa value is a measure for the deviation of a classifier from random. In game 51, all classifiers completely fail to find a hypothesis in subsets better than 'always certain strategy'. The best classifiers for games with a significant gap (game ID in a box) between average correctness in cross validation and maximal probability of a gesture predictions are sequential minimal optimization (SMO) (Platt, 1998) for 31, multinomial logistic regression (L) (Cessie & Houwelingen, 1992) for 41 and Bayesian networks for 53.

Which classifier is the most robust? One can choose two criteria - highest minimum performance or highest average performance. In game playing conditions, if the correctness of prediction is 5 percentage points higher, one gets a 5% higher payoff. To find the classifier with the most robust usability in game playing conditions, the difference between average correctness and probability of equal distribution ($\frac{1}{|\text{Strategies}|}$) is calculated for each classifier and game. SMO has the highest minimum difference and a simple variant of L (SL) has the highest average difference. On the other side, L has the the highest average Kappa value and voting feature intervals classification (VFI) has the highest minimum Kappa value. Fig.2 shows the average correctness of these classifiers on the datasets. Three of these four classifiers have functions as hypothesis space. The problem of functions is that most of them can not be verbalised. Consequently, the first question from the abstract can be answered using natural language. On the other side, the success of function based classifiers means that we can not explain our behavior in our natural language. However, the correctness achieved for game 31 is about 46% and it is slightly higher than in the related work. It is doubtful, wether one can define an algorithm which predicts exactly general human strategic behavoir at all.

The single rule classifier (OneR), which is also included in the histogram on fig.2, produces a hypothesis which contains only one single rule. Using this classifier, one can find out that 43.15% of the data in game 31 matches the rule 'choose paper after choosing rock, scissors after rock and rock after paper'. This rule is a very simple answer to the first question in the abstract in this paper. Such rules of thumb are not exact enough for explaining general human behavior. The difficulty of finding a relationship between input and output is the fact that the same input can cause different outputs. Even using the instance based approach K* which is validated on training data, one achieves only 80.37% correctness in game 31. Strategography and strategophony are possible future directions in understanding general human strategic behavior - if we can not verbalise our strategic behavior, can we represent it as images or music?

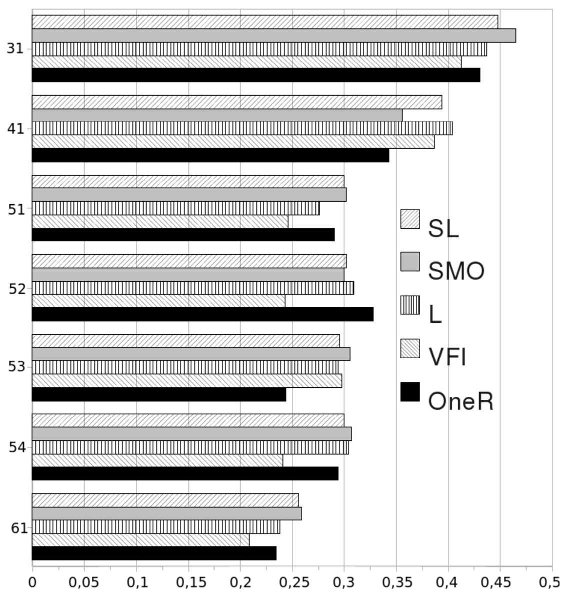


Figure 2: Cross validation

References

Cessie, S., & Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied Statistics*, *41*.

F.Camerer, C. (2003). *Behavioral game theory*. New Jersey: Princeton University Press.

Gal, Y., & Pfeffer, A. (2007). Modeling reciprocal behavior in human bilateral negotiation. In *Aaai.* AAAI Press.

Marchiori, D., & Warglien, M. (2008). Predicting human interactive learning by regret-driven neural networks. *Science*, *319*.

Mitchell, T. M. (1997). *Machine learning*. Mcgraw-Hill Higher Education.

Platt, J. (1998). Machines using sequential minimal optimization. In *Advances in kernel methods.* MIT Press.

Tagiew, R. (2009). Towards a framework for management of strategic interaction. In *Icaart.* INSTICC.

Witten, I. H., & Frank, E. (2005). *Data mining*. Morgan Kaufmann.

The Neural Basis for the Perceptual Symbol System and the Potential of Building a Cognitive Architecture Based on It

Weiyu Wang (gladiola.wang@family.ust.hk)
Department of Biology, Hong Kong Uni. of Sci. & Tech,
Clear Water Bay, Hong Kong

Abstract

The cortex is modeled as a net of minicolumns. A cell assembly in layer IV of a minicolumn is defined as a "cognit". Higher level cognits integrate lower level cognits. The minicolumn is able to form cognits and extract determinant cognit time series as perceptual symbols. A frame is a set of determinant cognit time series, and simulation is the partially retrieval of these determinant cognit time series. This model can fulfill the requirements for the perceptual symbol system proposed by Barsalou in 1999. I am trying to build neural network simulating this net. It is potentially a Cognitive architecture bearing connectionism.

Keywords: perceptual symbol system, minicolumn, cortex, connectionism, neural network

Introduction

On the standard view, perception and cognition are two distinct processes and the concepts in cognition are represented by amodal symbols. This viewpoint is challenged by the theory of perceptual symbol systems (Barsalou, 1999). They argued that there is no evidence for the existence of an amodal symbol system and symbols are intrinsically perceptual.

According to the work of Barsalou (1999), the kernel ideas of perceptual symbol system are:

1. A *perceptual symbol* is defined as a record of the neural activation during perception, while it is componential, multimodal, dynamic, and schematic.
2. A concept is equal to a simulator, which is composed of "an underlying *frame* that integrates perceptual symbols across category instances, and the potentially infinite set of *simulations* that can be constructed from the frame". A frame is object-centered and composed of multiple subregions, with four basic properties: predicates, attribute-value bindings, constraints, and recursion. The simulations must ensure categorization and categorical inferences, cognitive penetration, and stable conceptualization.

This article proposes a hypothesis of how cortex may fulfill these requirements and actualize the perceptual symbol system, and discuss the potential of building a cognitive architecture based on it.

Foundation from Neuroanatomy

Horizontally the cortex is composed of six layers. Layer IV contains different types of stellate and pyramidal cells, and is the main target of thalamocortical and intrahemispheric corticocortical afferents (input). Layer III contains predominantly pyramidal cells and is the principal source of corticocortical efferents (output) (Creutzfeldt, 1995). Vertically neocortex is columnar organized with elementary module minicolumn (Mountcastle, 1997).

Minicolumn Model

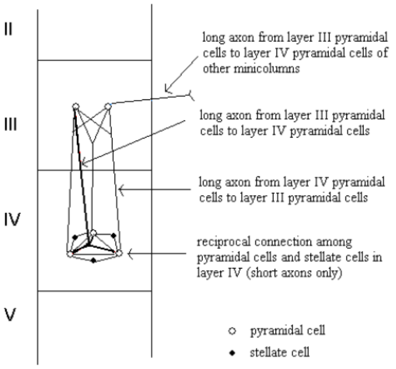

II
III
IV
V
long axon from layer III pyramidal cells to layer IV pyramidal cells of other minicolumns
long axon from layer III pyramidal cells to layer IV pyramidal cells
long axon from layer IV pyramidal cells to layer III pyramidal cells
reciprocal connection among pyramidal cells and stellate cells in layer IV (short axons only)
○ pyramidal cell
◆ stellate cell

Figure 1: A Modeled Minicolumn

The simplified model for a minicolumn is shown in Fig 1. It is viewed as a recursive neural network. Layer IV composes of both excitatory and inhibitory neurons and is the principle area for external input. Layer III composes of excitatory neurons only and is the principle area for external output. The transmission delay among neurons between layer III and IV is considered. A cell assembly formed in layer IV through Hebbian learning (Hebb, 1949) represents a most basic piece of knowledge, called a "cognit" here.

Cortex Model

The whole cortex is viewed as a net with each node a minicolumn. The nodal connection is from excitatory neurons in layer III of one minicolumn to excitatory neurons in layer IV of another minicolumn.

The cortex is organized with "vague hierarchy". The minicolumns receiving thalamus inputs which transmit information directly encoded by sensory organs are regarded as level 1. The minicolumns receiving inputs from level 1 minicolumns are regarded as level 2. So on and so forth, minicolumns receiving inputs from level n minicolumns are regarded as level n+1. Notice this hierarchy is not perfect as many minicolums can receive inputs from multiple minicolumns of different levels.

Information Processing and Learning

Any perception and cognition starts with sequential inputs from sensory organs. These inputs are viewed as discrete time series of states. Information processing has two aspects: information extraction is to represent each state by cognits (cognit formation) in level 1 minicolumns and associate cognit A to cognit B if pattern (A, B) repeatedly occurs (sequence prediction); information integration is to link higher level cognits to combination of lower level cognits spacially and temporally. Thus a higher level cognit may refer to a spaciotemporal combination of information and have very abstract meaning.

Learning occurring within a minicolumn (intra-minicolumnar) actualizes information extraction (Wang, 2008). Learning occurring among minicolumns (inter-minicolumnar) may actualize information integration.

Fulfillment of Perceptual Symbol System

A perceptual symbol is a time series of cognits in which any cognit predicts its follower (a determinant segment of the entire time series of cognits).

Componential: Any cognit (other than level 1) is a spaciotemporal combination of cognits.

Multimodal: One cognit can integrates information from cognits of minicolumns belonging to different modalities.

Dynamic: The learning modifies the system's behaviors.

Schematic: An individual is parsed into cognit series each of which represents a specific aspect of it. When perceiving a new individual, same cognit series can be retrieved (controlled by selective attention) if it shares a common specific aspect of the old one, enabling partial information retrieving.

A frame is simply a set of determinant cognit time series. This set stands for an (concrete or abstract) object, and the minicolumns containing these time series are subregions.

Predicates: this set itself is a predicate, as it defines the properties of the object.

Attribute-value Bindings: A specific specialization evokes specific determinant cognit time series in corresponding minicolumns.

Constraints: The many cognit time series identifying a specific specialization are integrated gradually through the hierarchy of the minicolumns, until a single cognit time series in one minicolumn represents this specialization. Feedback interminicolumnar connections enable the ability to retrieve all the attribute values of this specialization and prevent mismatches.

Recursion: multiple cognit time series of a frame can be integrated into a single cognit time series in a higher level minicolumn, which can in turn form an element (attribute) of a new frame.

A simulation is the partially retrieval of these determinant cognit time series. It ensures:

Categorization and Categorical Inferences: The specializations are put into a category if they share common attribute values. Thus one specialization can be put into multiple categories viewing from different aspects. Categorical inference is achieved by finding the frame from one attribute and retrieving this frame's other attributes.

Cognitive Penetration: When selective attention is paid to retrieving the attribute values of a specialization in higher level frames, the original representation in corresponding lower level frames can be suppressed and modified.

Stable Conceptualization: Concepts formed in this way have stable commonality among different individuals as the representation in level 1 minicolumns are same (similar) defined by genes. The formation of higher level representation is essentially a data mining process, concepts not derivable from level 1 representation cannot form, but only a small portion of information is mined out by an individual. Thus knowledge of individuals may differ, but there is no difficulty for them to communicate and understand others.

Potential for Building a Cognitive Architecture

The idea is to build a neural network with minicolumns as basic functional units. Each subnetwork for a minicolumn actualizes information extraction. The connection among subnetworks actualizes information integration.

The neural network for a minicolumn is already built. Its learning strategy needs to consider the transmission delay, threshold dynamic, and the overlapping problem (Wang, 2008). The strategy for inter-minicolumnar learning should be inspired from the molecular guidance for axon growth, synaptic elimination, etc. during neural development (Dickson, 2002; Lo, Poo, 1991; Purves, Lichtman, 1980).

References

Barsalou, L. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, 22, 577-660.

Creutzfeldt, O. (1995). *Cortex Cerebri: Performance, Structural and Functional Organization of the Cortex*. USA: Oxford University Press.

Dickson, B. (2002). Molecular Mechanisms of Axon Guidance. *Science* 298, 1959-1965.

Hebb, O. (1949). *The Organization of Behavior*. New York: Wiley.

Lo, Y., & Poo, M. (1991).Activity-Dependent Synaptic Competition in Vitro: Heterosynaptic Suppression of Developing Synapses. *Science*, 254, 1019-1022.

Mountcastle, V. (1997). The Columnar Organization of the Neocortex. *Brain*, 120, 701-722.

Purves, D., & Lichtman, J. (1980). Elimination of Synapses in the Developing Nervous System. *Science*, 210, 153-157.

Wang, W. (2008). A Hypothesis on How the Neocortex Extracts Information for Prediction in Sequence Learning. *Proceedings of the 5th international symposium on Neural Networks: Advances in Neural Networks* (pp. 21-29). Berlin, Heidelberg: Springer-Verlag.

A Holographic Model Of Frequency And Interference: Rethinking The Problem Size Effect

Robert L. West (robert_west@ carleton.ca),
Matthew F. Rutledge-Taylor (mrtaylo2@connect.carleton.ca), Aryn A. Pyke (apyke@connect.carleton.ca)
Institute of Cognitive Science, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6

Abstract

In this paper we used a holographic memory system to model Zbrodoff's (1995) findings on the problem size effect, a well-known effect in the area of Math Cognition. The data showed the effects of manipulating both frequency and interference.

Keywords: fan effect; frequency; holographic memory; interference; arithmetic

The Dynamically Structured Holographic Memory system (DSHM) uses holographic representations as a way of modeling human memory. It is based on Jones and Mewhort's BEAGLE lexicon model. The details of DSHM and the similarities to BEAGLE are discussed in Rutledge-Taylor & West (2007). One function that DSHM models well is memory interference. Rutledge-Taylor & West (2008) showed that the *fan effect* (Anderson, 1974) falls naturally out of the DSHM architecture.

The *fan effect* is a term used to describe a memory phenomenon in which the time needed to verify a fact is related to the number of other facts in memory that include concepts in common with the target fact (Anderson, 1974). The fan refers to how many facts share memory elements with the target. For example, if a person's declarative memory contained three propositions: "the hippie is in the park", "the lawyer is in the store", and "the lawyer is in the bank", then the fan of the terms 'hippie', 'park', 'store', and 'bank' are one, while the fan of the term 'lawyer' is two. As first demonstrated by Anderson (1974), larger fans cause slower reaction times in human subjects. This result is consistent with the theory that similar facts cause interference in the retrieval process.

The DSHM model has been used to model the fan effect (Rutledge-Taylor & West, 2008). However, the fan effect addresses only the effect of inter-fact 'interference' on the efficiency of fact retrieval. But, there is another factor that also strongly impacts retrieval speed/efficiency: the person's frequency of exposure to that fact. For example, if a participant reads "the lawyer is in the store" once and "the lawyer is in the bank" four times, the fans of 'store' and 'bank' are each still one. However, one would expect that the association between 'lawyer' and 'bank' to be stronger than the association between 'lawyer' and 'store'. Thus, both fan effects and frequency effects impact the efficiency of fact retrieval. To test the interaction of frequency and fan in DSHM we modeled the data of Zbrodoff (1995), who manipulated both of these in the context of learning alphabet arithmetic facts (e.g., A + 3 = D, which indicates that the number three letters past A is D). Zbrodoff repeatedly represented these facts and measured true/false response reaction times across trials to study learning.

In Experiment 4 all of the problems were presented with equal frequency. To model this, each problem, including the answer and whether the answer was true or false, was represented as a random vector and entered into the DSHM, so that one entry equaled one presentation to a subject. There were two ways the model could decide if a question was true. One was to submit a question vector with the problem plus the answer and a *blank* for whether it was true or false. The model would then return whether or not it believed the question was true or false. The second way was to submit the question with the answer as a *blank* and whether or not it was true filled in with *true*. In this case the model would return what it believed to be the correct answer (note, the model can make errors but this data is not presented here).

The second method fit the data better than the first, suggesting that people were recalling the answers to see if the questions were true or false. In this case the model makes the same predictions for true and false questions. Consistent with this, the human data was very similar for the true and false questions. To get accurate reaction times from the model the inverse of the activation levels were scaled up by a factor of 400. Note that this represents a claim that the activation levels of the model translate directly into reaction times. Figure 1 presents the results.

Experiment 3 was the same as Experiment 4 except that frequency was manipulated so that the questions with the smaller numerical addends were presented more frequently. The model used here was exactly the same as the one used to model Experiment 4. No parameters were altered! Figure 2 shows the human data and the simulation results. Overall, the model does a good job of accounting for the results. The only exception occurs in the later blocks (not shown on the graphs) where the model continues to have the addends 2 and 4 close together with the addend 3 higher. In contrast, in the human data, the addend 4 moves back up closer to the addend 3. This result is difficult to interpret. It could be that the model does not predict well for long term learning, although it did accurately predict long-term learning for Experiment 4. Another possibility is that subjects were using a rehearsal strategy between sessions. If subjects were recalling the questions and checking them by calculation, or rehearsing them, it could produce this effect since the addend-4 questions would be harder to recall due to the low frequency of presentation (for random recall without a cue, interference should not play a role).

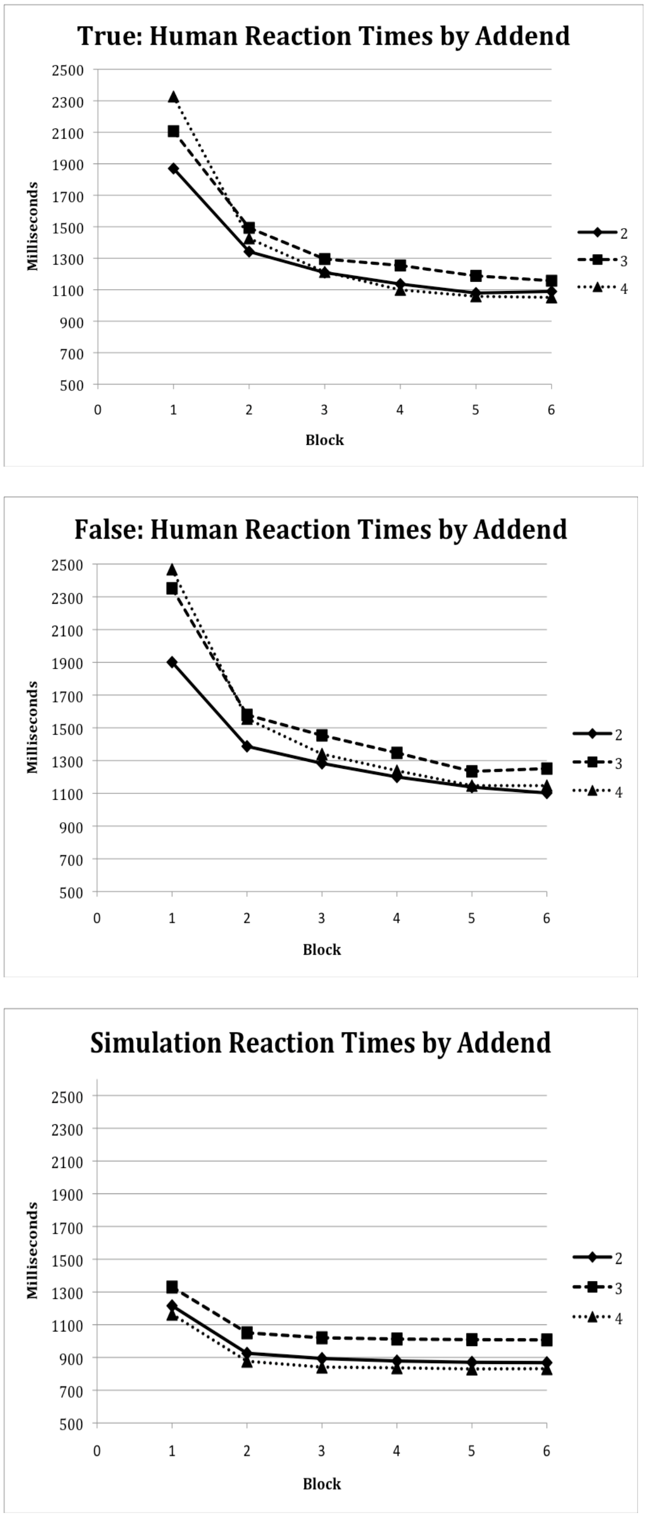


Figure 1: Data and Simulation for Experiment 4

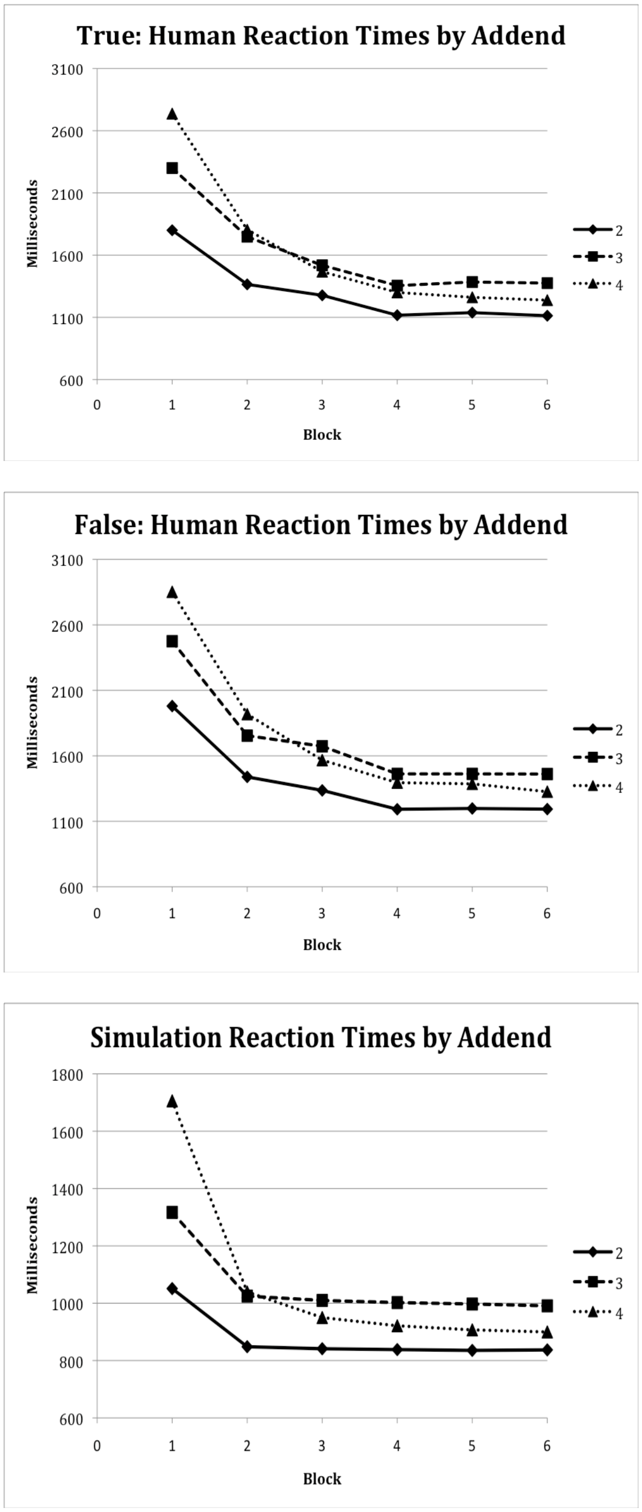


Figure 2: Data and Simulation for Experiment 3

References

Anderson, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology, 6,* 451-474.

Rutledge-Taylor, & West, R. L. (2008). Modeling The Fan Effect Using DSHM. *Cognitive Science Procedings.*

Zbrodoff, N. J. (1995). Why is 9 + 7 harder than 2 + 3? Strength and interference as explanations of the problem-size effect. *Memory and Cognition*, 23(6), 689–700.

A Comparison of Decision-Making Models for Determining File Importance

K. C. Wong (k-wong@govst.edu)
Division of Science
Governors State University
University Park, IL 60466 USA

Keywords: cognition; user modeling

Introduction

File replication is the most popular approach used to promote system reliability and file availability in a network-based environment (Purdin, et al. 1987; Son, 1987; Rodrigues, et al., 2002). However, all of the distributed file systems equipped with the functionality of file replication require their users to determine how important their files are in order to assist systems in making decisions regarding how many replicas should be made and distributed in the networks (Blair, et al., 1983). As such, system users are inevitably burdened with this potential responsibility. The problem can be partially alleviated if systems can take more responsibility for their users on determining file importance. To achieve this goal, however, we need to better understand how system users cognitively make decisions regarding determining file importance. In this paper, we quantitatively compare the performance of three decision-making models popularly used in juror decision-making (Pennington & Hastie, 1981) to examine how satisfactorily they model the process of determining file importance. The three models are the linear weighting model, the Bayesian model, and the Poisson model.

The Three Decision-Making Models

The Linear Weighting Model

The linear weighting model postulates that file importance can be determined by linearly combining those weighted pieces of information (referred to as predictors in this paper) during the session of determining file importance. The set of weights associated with the predictors identified can be determined in such a way that predicted file importance is optimally correlated with observed file importance using multiple regression analysis (Rawlings, 1988).

The Bayesian Model

The Bayesian model postulates that file importance can be determined by a series of simple inferences, in which importance is revised according to the direct impact of the predictors identified independently. In other words, the determination of file importance using the model is concerned with determining the posterior odds for importance (R_n), which is defined in terms of determining the ratio of the probability of importance given all the predictors identified, to the probability of unimportance given all the predictors identified. Once R_n is determined, it is compared with the decision criterion (dc) adopted by the system user to judge if the file under consideration is important (if $R_n \geq dc$) or not (if $R_n < dc$).

The Poisson Model

The Poisson model postulates that determining file importance is a Poisson process. In the process, it assumes that there exists an apparent weight of predictors (w) important to the file under consideration. The apparent weight accumulates constantly with time during the session of determining file importance until either a critical predictor is identified or the end of the session is encountered. The apparent weight accumulated (w_a) is then compared with the decision criterion (dc) adopted by the system user to judge if the file under consideration is important (if $w_a \geq dc$) or not (if $w_a < dc$).

Data Collection And The Experiment

Five predictors were systematically identified in this study for model comparison: the number of characters keyed, the computer cost spent, file length, file dependency, and the frequency of file access. Correlation coefficients between observed and predicted file importance were used to quantitatively evaluate the performance of the three models. A computer program, written in C++, was designed and implemented on a laptop to collect data for observed file importance and the five predictors. The data collected were classified into five importance ratings (from important to unimportant) and mapped proportionally to an importance rating scale (from 1 to 5, respectively). There were 41 subjects (randomly selected in an academic environment) participating in the experiment. These subjects accessed a total of 169 files. Since the subjects were asked to randomly pick up their files created by them, the sample may contain various types of file contents.

Model Comparison

Correlation Coefficients

The correlation coefficients computed for each of the models suggest that the linear weighting model and the Bayesian model with dc = 1 perform much more satisfactorily than the Poisson model using the empirical data collected in the study. The poor performance of the Poisson model may be resulted from the following three possible sources of errors: (1) the data collected may not be representative; (2) the assumptions made in this study may not hold for the model; (3) the model itself is inferior. More studies are needed to clarify the issue.

Nature of File Importance Determination

The linear weighting model is characterized by the nature of determining file importance slightly different from the Bayesian model and the Poisson model. The former model determines how important the file under consideration is (a rated outcome), while the latter models determine whether or not the file under consideration is important (a binary outcome). Moreover, the linear weighting model associates file importance ratings directly with predictor ratings in determining file importance. On the other hand, the Bayesian model and the Poisson model convert predictor ratings into predictor appearance probability, which may not be directly related to file importance ratings. As such, the linear weighting model provides more information about how each of the predictors is correlated with each other, and how each of the predictors is weighted by the subjects.

Implementation Efficiency

There is no noticeable performance difference in model implementation and file importance determination using the three models. All of the three models need an order of O(n×m) accesses to various data items for model implementation and an order of O(m) accesses to determine predicted file importance, where n = the number of files created by a subject and m = the number of predictors each file has.

Decision-Making Processes

The three models studied have quite different decision-making processes, reflecting how system users cognitively make decisions regarding determining file importance. The linear weighting model postulates that determining file importance is a process consisting primarily of two phases: predictor collection and predictor evaluation. In the predictor collection phase, all possible predictors are collected. The predictors collected are then assigned weights in the predictor evaluation phase and combined linearly to determine file importance.

The Bayesian model postulates that in the process of determining file importance, once a predictor is identified, it will be evaluated to examine how likely the predictor is the one identified, given that the file under consideration is important and unimportant, respectively. The likelihood ratios thus computed constitute a series of inferences, in which posterior odds for importance is revised according to the direct impact of the predictors identified independently. At the end of the process, the revised posterior odds is compared with the decision criterion adopted by the subject to determine whether or not the file under consideration is important.

The Poisson model postulates that there exists an apparent weight of predictors important to the file under consideration. The apparent weight accumulates constantly with time in the process of determining file importance until either a critical predictor is identified or the end of the process is encountered. In the process, once a predictor is identified, it is judged by the subjects to examine if it is a critical predictor. The apparent weight accumulated up to the time when the critical predictor appears or the process ends is compared with the decision criterion adopted by the subject to determine whether or not the file under consideration is important.

References

Blair, G. S. Blair, Mariani, J. A., Nicol, J. R. and Shepherd, W. D. A Knowledge Based Operating System, *The Computer Journal*, vol. 30, no. 3, pp. 193-200, 1983.

Pennington, N. Pennington & Hastie, R. Juror Decision Making Models: The Generalization Gap, *Psychological Bulletin*, vol. 89, no. 2, pp. 246-287, 1981.

Prudin, Titus D. M., Schlichting, Richard D., & Andrews, Gregory R. A File Replication Facility for Berkeley Unix, *Software – Practice and Experience*, vol. 17(12), pp. 923-940, 1987.

Rawlings, J. O. Applied Regression Analysis: A Research Tool, Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA, 1988.

Rodrigues, Rodrigo, Liskov, Barbara, & Shrira, Liuba. The Design of Robust Peer-To-Peer System, *Tenth ACM SIGOPS European Workshop*, September, 2002.

Son, Sang Hyuk. Using Replication to Improve Reliability in Distributed Information Systems, *Information and Software Technology*, vol. 29, no. 8, pp. 440-449, 1987.