How does rumination impact cognition? A first mechanistic model.

van der Velde, Maarten; ESM-MERGE Investigators

Published in:
Proceedings of the 15th International Conference on Cognitive Modeling

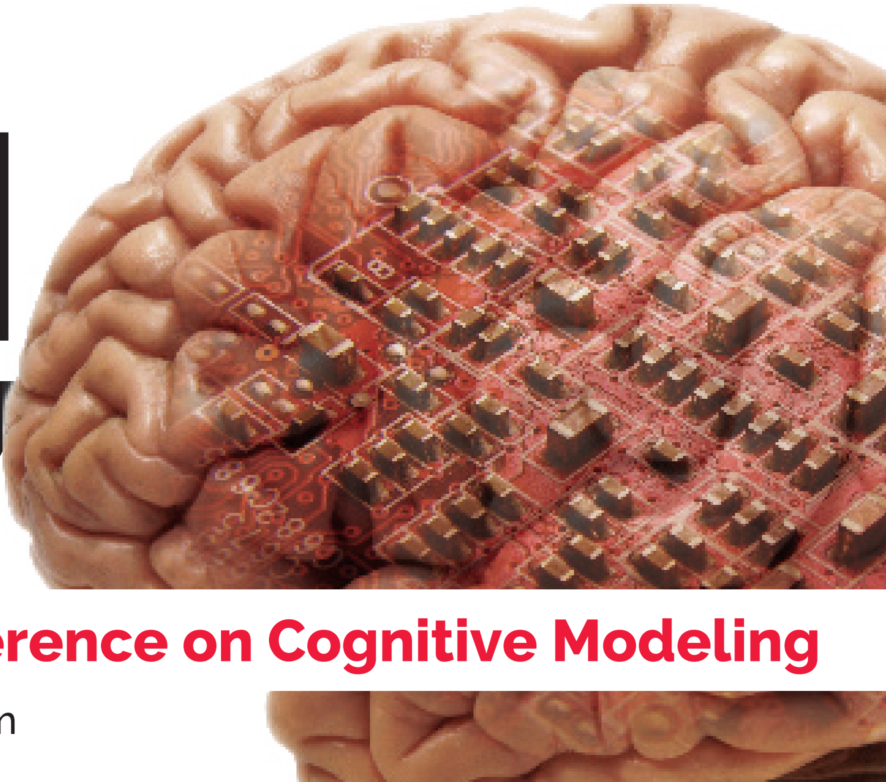
Publication date:
2017

Citation for published version (APA):
van der Velde, M., & ESM-MERGE Investigators (2017). How does rumination impact cognition? A first mechanistic model. In M. van Vugt, A. P. Banks, & W. G. Kennedy (Eds.), *Proceedings of the 15th International Conference on Cognitive Modeling* (pp. 25-30). University of Warwick, Inclusive Technology Ltd. http://Howdoesruminationimpactcognition? Afirstmechanisticmodel

WARWICK
Behavioural Science

Editors Marieke K. van Vugt, Adrian P. Banks & William G. Kennedy

Proceedings of ICCM 2017



15th International Conference on Cognitive Modeling

July 22-25, Coventry, United Kingdom

Preface

The International Conference on Cognitive Modeling (ICCM) is the premier conference for research on computational models and computation-based theories of human cognition. ICCM is a forum for presenting and discussing the complete spectrum of cognitive modeling approaches, including connectionism, symbolic modeling, dynamical systems, Bayesian modeling, and cognitive architectures. Research topics can range from low-level perception to high-level reasoning. In 2017 we for the first time jointly held our conference with the Society for Mathematical Psychology. The 15th ICCM was held at the University of Warwick in Coventry, United Kingdom, on July 22nd-25th, 2017.

All papers and abstracts in the ICCM 2017 proceedings may be cited as follows:

Author, A., & Author, B. (2017). This is the title of the paper. In M. K. van Vugt, A. P. Banks, & W. G. Kennedy (Eds.), Proceedings of the 15th International Conference on Cognitive Modeling (pp. 1-6). Coventry, United Kingdom: University of Warwick.

We would like to acknowledge the Society for Mathematical Psychology (SMP), Behavioural Sciences at the University of Warwick, the Artificial Intelligence Journal, and the US Army Research Laboratory and Army Research Office, whose generous support kept the conference fees low and allowed us to fund a large number of student awards. We also would like to acknowledge the people who brought these conferences all together for the first time (Andrew Heathcote, Amy Criss, Frank Ritter, and David Reitter), the hard work of the officers of the SMP (Brent Miller, Leslie Blaha, Richard Golden, and Scott Brown), Leslie Blaha for helping with the grant applications, Jelmer Borst for design of the poster and proceedings, University of Warwick staff for making the conference tax exempt (Steve McGaldrigan and Jonathan Pearce), the help of Warwick PhD students on many aspects of the conference (Alexandra Surdina, Jake Spicer, Mengran Wang, and Jianqiao Zhu), and the event staffing by Warwick MSc students

Marieke K. van Vugt
Adrian P. Banks
William G. Kennedy

Organizing Committee

Marieke K. van Vugt
Adrian P. Banks
William G. Kennedy

Program Committee

Erik Altmann	Michigan State University
Adrian P. Banks	University of Surrey
Thomas Barkowsky	University of Bremen
Leslie Blaha	Pacific Northwest National Laboratory
Jelmer Borst	University of Groningen
Mike Byrne	Rice University
Rich Carlson	Pennsylvania State University
Rick Cooper	Birkbeck, University of London
Chris Dancy	Bucknell University
Fabio Del Missier	University of Trieste
Francesco Gagliardi	Italian Association for Cognitive Sciences
Moojan Ghafurian	Pennsylvania State University
Kevin Gluck	Air Force Research Laboratory
Fernand Gobet	University of Liverpool
Cota Gonzalez	Carnegie Mellon University
Joseph Houpt	Wright State University
Chris Janssen	Utrecht University
Gary Jones	Nottingham Trent University
Ioan Juvina	Wright State University
Mark Keane	UCD Dublin
William G Kennedy	George Mason University
David Kieras	University of Michigan
Joseph Krems	University of Chemnitz
Johan Kwisthout	Radboud University Nijmegen
John Laird	University of Michigan
Christian Lebiere	Carnegie Mellon University
Lyle Long	Penn State University
Ralf Mayrhofer	University of Göttingen
Katja Mehlhorn	University of Groningen
Christopher Meyers	AFRL/711 Human Performance Wing
Junya Morita	Shizuoka University
Shane Mueller	Michigan Technological University
David Noelle	University of California, Merced
Enkhbold Nyamsuren	Open University
David Peebles	University of Huddersfield
Krishna Prasad	IIT Gandhinagar
David Reitter	Pennsylvania State University
Frank Ritter	Pennsylvania State University

Nele Russwinkel	Technical University Berlin
Dario Salvucci	Drexel University
Ute Schmid	University of Bamberg
Mike Schoelles	Rensselaer Polytechnic Institute
Lambert Schomaker	University of Groningen
Lael Schooler	Syracuse University
Holger Schultheis	University of Bremen
Jennifer Spenader	University of Groningen
Robert St Amant	NCSU
Chris Stevens	University of Groningen
Terence Stewart	University of Waterloo
Andrea Stocco	University of Washington
Ron Sun	Rensselaer Polytechnic Institute
Greg Trafton	US Naval Research Laboratory
Jacolien van Rij	University of Groningen
Marieke van Vugt	University of Groningen
Sharon Wood	University of Sussex
Iraide Zipitria	UPV/EHU

Table of Contents

Sunday July 23rd

9:00-10:20: Language

10:40-12:00: Emotion

14:40-16:00: Physiology

Monday July 24th

9:00-10:20: Neuroscience

10:40-12:00: Neuroscience

14:40-16:00: Reasoning

16:20-17:20: Decision making

Tuesday July 25th

9:00-10:20: Human performance & Visual cognition

10:40-12:00: Artificial systems

14:40-16:00: Language

Posters

A computational investigation of sources of variability in sentence comprehension difficulty in aphasia

Paul Mätzig (pmaetzig@uni-potsdam.de)
University of Potsdam, Human Sciences Faculty, Department Linguistics,
24–25 Karl-Liebknecht-Str., Potsdam 14476, Germany

Shravan Vasishth, (vasishth@uni-potsdam.de)
University of Potsdam, Human Sciences Faculty, Department Linguistics,
24–25 Karl-Liebknecht-Str., Potsdam 14476, Germany

Felix Engelmann (felix.engelmann@manchester.ac.uk)
The University of Manchester, School of Health Sciences
Child Study Centre, Coupland 1, Oxford Road, Manchester M13 9PL

David Caplan (dcaplan@partners.org)
Massachusetts General Hospital
175 Cambridge St, #340, Boston, Massachusetts 02114

Abstract

We present a computational evaluation of three hypotheses about sources of deficit in sentence comprehension in aphasia: slowed processing, intermittent deficiency, and resource reduction. The ACT-R based Lewis and Vasishth (2005) model is used to implement these three proposals. Slowed processing is implemented as slowed default production-rule firing time; intermittent deficiency as increased random noise in activation of chunks in memory; and resource reduction as reduced goal activation. As data, we considered subject vs. object relatives whose matrix clause contained either an NP or a reflexive, presented in a self-paced listening modality to 56 individuals with aphasia (IWA) and 46 matched controls. The participants heard the sentences and carried out a picture verification task to decide on an interpretation of the sentence. These response accuracies are used to identify the best parameters (for each participant) that correspond to the three hypotheses mentioned above. We show that controls have more tightly clustered (less variable) parameter values than IWA; specifically, compared to controls, among IWA there are more individuals with low goal activations, high noise, and slow default action times. This suggests that (i) individual patients show differential amounts of deficit along the three dimensions of slowed processing, intermittent deficient, and resource reduction, (ii) overall, there is evidence for all three sources of deficit playing a role, and (iii) IWA have a more variable range of parameter values than controls. In sum, this study contributes a proof of concept of a quantitative implementation of, and evidence for, these three accounts of comprehension deficits in aphasia.

Keywords: Sentence Comprehension; Aphasia; Computational Modeling; Cue-based Retrieval

Introduction

In healthy adults, sentence comprehension has long been argued to be influenced by individual differences; a commonly assumed source is differences in working memory capacity (Daneman & Carpenter, 1980; Just & Carpenter, 1992). Other factors such as age (Caplan & Waters, 2005) and cognitive control (Novick, Trueswell, & Thompson-Schill, 2005) have also been implicated.

An important question that has not received much attention in the computational psycholinguistics literature is: what are sources of individual differences in healthy adults versus impaired populations, such as individuals with aphasia (IWA)? It is well-known that sentence processing performance in IWA is characterised by a performance deficit that expresses itself as slower overall processing times, and lower accuracy in question-response tasks (see literature review in Patil, Hanne, Burchert, De Bleser, & Vasishth, 2016). These performance deficits are especially pronounced when IWA have to engage with sentences that have non-canonical word order and that are semantically reversible, e.g. Object-Verb-Subject versus Subject-Verb-Object sentences (Hanne, Sekerina, Vasishth, Burchert, & Bleser, 2011).

Regarding the underlying nature of this deficit in IWA, there is a consensus that some kind of disruption is occurring in the syntactic comprehension system. The exact nature of this disruption, however, is not clear. Although a broad range of proposals exist (see Patil et al., 2016), we focus on three influential proposals here:

1. *Intermittent deficiencies*: Caplan, Michaud, and Hufford (2015) suggest that occasional temporal breakdowns of parsing mechanisms capture the observed behaviour.
2. *Resource reduction*: A third hypothesis, due to Caplan (2012), is that the deficit is caused by a reduction in resources related to sentence comprehension.
3. *Slowed processing*: Burkhardt, Piñango, and Wong (2003) argue that a slowdown in parsing mechanisms can best explain the processing deficit.

Computational modelling can help evaluate these different proposals quantitatively. Specifically, the cue-based retrieval account of Lewis and Vasishth (2005), which was developed within the ACT-R framework (Anderson et al., 2004), is a computationally implemented model of unimpaired sentence comprehension that has been used to model a broad array of empirical phenomena in sentence processing relating to similarity-based interference effects (Lewis & Vasishth, 2005; Nicenboim & Vasishth, 2017; Vasishth, Bruessow, Lewis, & Drenhaus, 2008; Engelmann, Jäger, & Vasishth, 2016) and the interaction between oculomotor control and sentence comprehension (Engelmann, Vasishth, Engbert, & Kliegl, 2013).[1]

The Lewis and Vasishth (2005) model is particularly attractive for studying sentence comprehension because it relies on the general constraints on cognitive processes that have been laid out in the ACT-R framework. This makes it possible to investigate whether sentence processing could be seen as being subject to the same general cognitive constraints as any other information processing task, which does not entail that there are no language specific constraints on sentence comprehension. A further advantage of the Lewis and Vasishth (2005) model in the context of theories of processing deficits in aphasia is that several of its numerical parameters (which are part of the general ACT-R framework) can be interpreted as implementing the three proposals mentioned above.

In Patil et al. (2016), the Lewis and Vasishth (2005) architecture was used to model aphasic sentence processing on a small scale, using data from seven patients. They modelled proportions of fixations in a visual world task, response accuracies and response times for empirical data of a sentence-picture matching experiment by Hanne et al. (2011). Their goal was to test two of the three hypotheses of sentence comprehension deficits mentioned above, slowed processing and intermittent deficiency.

In the present work, we provide a proof of concept study that goes beyond Patil et al. (2016) by evaluating the evidence for the three hypotheses—slowed processing, intermittent deficiencies, and resource reduction—using a larger dataset from Caplan et al. (2015) with 56 IWA and 46 matched controls.

Before we describe the modelling carried out in the present paper and the data used for the evaluation, we first introduce the cognitive constraints assumed in the Lewis and Vasishth (2005) model that are relevant for this work, and show how the theoretical approaches to the aphasic processing deficit can be implemented using specific model parameters. Having introduced the essential elements of the model architecture, we simulate comprehension question-response accuracies for unimpaired controls and IWA, and then fit the simulated accuracy data to published data (Caplan et al., 2015) from controls and IWA. When fitting individual participants, we vary three parameters that map to the three theoretical proposals mentioned above. The goal was to determine whether the distributions of parameter values furnish any support for any of the three sources of deficits in processing. We expect that if there is a tendency in one parameter to show non-default values in IWA, for example slowed processing, then there is support for the claim that slowed processing is an underlying source of processing difficulty in IWA. Similar predictions hold for the other two constructs, intermittent deficiency and resource reduction; and for combinations of the three proposals.

Constraints on sentence comprehension in the Lewis and Vasishth (2005) model

In this section, we describe some of the constraints assumed in the Lewis and Vasishth (2005) sentence processing model. Then, we discuss the model parameters that can be mapped to the three theoretical proposals for the underlying processing deficit in IWA.

The ACT-R architecture assumes a distinction between long-term declarative memory and procedural knowledge. The latter is implemented as a set of rules, consisting of condition-action pairs known as production rules. These production rules operate on units of information known as chunks, which are elements in declarative memory that are defined in terms of feature-value specifications. For example, a noun like *book* could be stored as a feature-value matrix that states that the part-of-speech is nominal, number is singular, and animacy status is inanimate:

$$\begin{pmatrix} \text{pos} & \textit{nominal} \\ \text{number} & \textit{sing} \\ \text{animate} & \textit{no} \end{pmatrix}$$

Each chunk is associated an *activation*, a numeric value that determines the probability and latency of access from declarative memory. Accessing chunks in declarative memory happens via a cue-based retrieval mechanism. For example, if the noun *book* is to be retrieved, cues such as {part-of-speech nominal, number singular, and animate no} could be used to retrieve it. Production rules are written to trigger such a retrieval event. Retrieval only succeeds if the activation of a to-be-retrieved chunk is above a minimum threshold, which is a parameter in ACT-R.

The activation of a chunk is determined by several constraints. Let C be the set of all chunks in declarative memory. The total activation of a chunk $i \in C$ equals

$$A_i = B_i + S_i + P_i + \varepsilon, \qquad (1)$$

where B_i is the base-level or resting-state activation of the chunk i; the second summand S_i represents the spreading activation that a chunk i receives during a particular retrieval event; the third summand is a penalty for mismatches between a cue value j and the value in the corresponding slot of chunk i; and finally, ε is noise that is logistically distributed, approximating a normal distribution, with location 0 and scale ANS which is related to the variance of the distribution. It is generated at each new retrieval request. The retrieval time T_i of a chunk i depends on its activation A_i via $T_i = F\exp(-A_i)$, where F is a scaling constant which we kept constant at 0.2 here.

The scale parameter ANS of the logistic distribution from which ε is generated can be interpreted as implementing the *intermittent deficiency* hypothesis, because higher values of ANS will tend to lead to more fluctuations in activation of a

[1] The model can be downloaded in its current form from https://github.com/felixengelmann/act-r-sentence-parser-em.

chunk and therefore higher rates of retrieval failure.[2] Increasing ANS leads to a larger influence of the random element on a chunk's activation, which represents the core idea of *intermittent deficiency*: that there is not a constantly present damage to the processing system, but rather that the deficit occasionally interferes with parsing, leading to more errors.

The second summand in (1), representing the process of *spreading activation* within the ACT-R framework, can be made more explicit for the goal buffer and for retrieval cues $j \in \{1,\dots,J\}$ as

$$S_i = \sum_{j=1}^{J} W_j S_{ji}. \qquad (2)$$

Here, $W_j = \frac{\mathrm{GA}}{J}$, where GA is the *goal activation* parameter and S_{ji} is a value that increases for each matching retrieval cue. S_{ji} reflects the association between the content of the goal buffer and the chunk i. The parameter GA determines the total amount of activation that can be allocated for all cues j of the chunk in the goal buffer. It is a free parameter in ACT-R. This parameter, sometimes labelled the "W parameter", has already been used to model individual differences in working memory capacity (Daily, Lovett, & Reder, 2001). Thus, it can be seen as one way (although by no means the only way) to implement the resource reduction hypothesis. The lower the GA value, the lower the difference in activation between the retrieval target and other chunks. This leads to more retrieval failures and lower differences in retrieval latency on average.

Finally, the hypothesis of *slowed processing* can be mapped to the *default action time* DAT in ACT-R. This defines the constant amount of time it takes a selected production rule to "fire", i.e. to start the actions specified in the action part of the rule. Higher values would lead to a higher delay in firing of production rules. Due to the longer decay in this case, retrieval may be slower and more retrieval failures may occur.

Next, we evaluate whether there is evidence consistent with the claims regarding slowed processing, intermittent deficiency, and resource reduction, when implemented using the parameters described above.

Simulations

In this section we describe our modelling method and the procedure we use for fitting the model results to the empirical data from Caplan et al. (2015).

Materials

We used the data from 56 IWA and 46 matched controls published in Caplan et al. (2015). In this data-set, participants listened to recordings of sentences presented word-by-word; they paced themselves through the sentence, providing self-paced listening data. Participants processed 20 examples of 11 spoken sentence types and indicated which of two pictures corresponded to the meaning of each sentence. This yielded accuracy data for each sentence type.

We chose two of the 11 sentence types for the current simulation: simple subject relatives (*The woman who hugged the girl washed the boy*) vs. object relatives (*The woman who the girl hugged washed the boy*), and subject relatives with a reflexive (*The woman who hugged the girl washed herself*) vs. object relatives with a reflexive (*The woman who the girl hugged washed herself*). We chose relative clauses for two reasons. First, relative clauses have been very well-studied in psycholinguistics and serve as a typical example where processing difficulty is (arguably) experienced due to deviations in canonical word ordering (Just & Carpenter, 1992). Second, the Lewis and Vasishth model already has productions defined for these constructions, so the relative clause data serve as a good test of the model as it currently stands. The reflexive in the second sentence type adds an additional layer of complexity to the sentences. In the model, this is reflected by an additional retrieval process on the reflexive, where the antecedent is retrieved.

The Caplan et al. (2015) dataset only provides accuracy data for the dependency between the embedded verb and its subject. We will address this problem in future studies where new data will be collected.

Lastly, since the production rules in the model were designed for modelling unimpaired processing, using them for IWA amounts to assuming that there is no damage to the parsing system per se, but rather that the processing problems in IWA are due to some subset of the cognitive constraints discussed earlier. This also implies that the IWA's parsing system is not engaged in heuristic processing, as has sometimes been claimed in the literature; see ? (?) for discussion on that point.

Method

For the simulations, we refer to as the parameter space Π the set of all vectors $(\mathrm{GA},\mathrm{DAT},\mathrm{ANS})$ with GA, DAT, ANS $\in \mathbb{R}$. For computational convenience, we chose a discretisation of Π by defining a step-width and lower and upper boundaries for each parameter. In this discretised space Π', we chose GA $\in \{0.2, 0.3, \dots, 1.1\}$, DAT $\in \{0.05, 0.06, \dots, 0.1\}$, and ANS $\in \{0.15, 0.2, \dots, 0.45\}$.[3] Π' could be visualised as a three-dimensional grid of 420 dots, which are the elements $p' \in \Pi'$.

The default parameter values were included in Π'. This means that models that vary only one or two of the three parameters were included in the simulations. This is motivated by the results of Patil et al. (2016): there, the combined model varying both parameters (default action time (DAT) and utility noise) achieved the best fit to the data. Including all mod-

[2] As an aside, note that Patil et al. (2016) implemented intermittent deficiency using another source of noise in the model (utility noise). In future work, we will compare the relative change in quality of fit when intermittent deficiency is implemented in this way.

[3] The standard settings in the Lewis and Vasishth (2005) model are GA = 1, DAT = 0.05 (or 50 ms), and ANS = 0.15.

		GA	DAT	ANS	GA & DAT	GA & ANS	DAT & ANS	GA & DAT & ANS
SR	control	19	24	18	18	11	16	10
	IWA	38	41	42	32	33	36	27
OR	control	21	26	36	21	20	25	20
	IWA	40	48	53	38	40	48	38

Table 1: Number of participants in **simple subject / object relatives** for which non-default parameter values were predicted, in the subject vs. object relative tasks, respectively; for goal activation (GA), default action time (DAT) and noise (ANS) parameters.

		GA	DAT	ANS	GA & DAT	GA & ANS	DAT & ANS	GA & DAT & ANS
SR	control	17	36	23	11	11	5	5
	IWA	40	46	42	36	35	31	31
OR	control	28	26	37	27	19	27	18
	IWA	51	48	51	44	46	41	39

Table 2: Number of participants in **subject / object relatives with reflexives** for which non-default parameter values were predicted, in the subject vs. object relative tasks, respectively; for goal activation (GA), default action time (DAT) and noise (ANS) parameters.

els allows us to do a similar investigation.

For all participants in the Caplan et al. (2015) data-set, we calculated comprehension question response accuracies, averaged over all items of the subject / object relative clause and subject / object relative clause with reflexive conditions. For each $p' \in \Pi'$, we ran the model for 1000 iterations for the subject and object relative tasks. From the model output, we determined whether the model made the correct attachment in each iteration, i.e. whether the correct noun was selected as subject of the embedded verb, and we calculated the accuracy in a simulation for a given parameter $p' \in \Pi'$ as the proportion of iterations where the model made the correct attachment. We counted a parsing failures, where the model did not create the target dependency, as an incorrect response.

The problem of finding the best fit for each subject can be phrased as follows: for all subjects, find the parameter vector that minimises the absolute distance between the model accuracy for that parameter vector and each subject's accuracy. Because there might not always be a unique p' that solves this problem, the solution can be a set of parameter vectors. If for any one participant multiple optimal parameters were calculated, we averaged each parameter value to obtain a unique parameter vector. This transforms the parameter estimates from the discretised space Π' to the original parameter space Π.

Results

In this section we presents the results of the simulations and the fit to the data. First, we describe the general pattern of results reflected by the distribution of non-default parameter estimates per subject. Following that, we test whether tighter clustering occurs in controls.

Distribution of normal parameter values Tables 1 and 2 show the number of participants for which a non-default parameter value was predicted. By default values we mean the values $\text{GA} = 1$, $\text{DAT} = 0.05$ (or 50 ms), and $\text{ANS} = 0.15$. It is clear that, as expected, the number of subjects with non-default parameter values is always larger for IWA vs. controls, but controls show non-default values unexpectedly often. In controls, the main difference between subject and object relatives is a clear increase in elevated noise values in object relatives for both simple subject / object relatives and those with reflexives. Perhaps surprisingly, in the reflexives condition (cf. Table 2), controls display higher DAT in subject vs. object relatives.

For IWA in simple subject relatives, the single-parameter models are very similar, whereas in simple object relatives, most IWA (95%) exhibit elevated noise values, while a far smaller proportion (71%) showed reduced goal activation values. In the relatives with reflexives, IWA show the same pattern in subject and object relatives, with a high degree of non-default parameter estimates for each of the three parameters.

Overall, most IWA exhibit non-default parameter settings ANS and DAT. While in subject / object relatives with reflexives, a similar number of IWA shows elevated GA settings, we think this might be due to the similar model behaviours that non-default GA and ANS elicit. We address this point in the discussion below.

Cluster analysis In order to investigate the predicted clustering of parameter estimates, we performed a cluster analysis on the data too see to which degree controls and IWA could be discriminated. If our prediction is correct that, compared to IWA, clustering is tighter in controls, we expect that a higher proportion of the data should be correctly assigned to one of two clusters, one corresponding to controls, the other one corresponding to IWA. We chose hierarchical clustering to test this prediction.

We combined the data for subject and object relatives into

	Subject relatives		Object relatives	
predicted group	controls	IWA	controls	IWA
control	**34**	21	**42**	24
IWA	12	**35**	4	**32**
accuracy	74%	63%	91%	57%

Table 3: Discrimination ability of hierarchical clustering on the combined data for **simple subject / object relative clauses**. Numbers in bold show the number of correctly clustered data points. The bottom row shows the percentage accuracy.

	Subject relatives		Object relatives	
predicted group	controls	IWA	controls	IWA
control	**31**	17	**27**	45
IWA	15	**39**	19	**11**
accuracy	67%	70%	59%	20%

Table 4: Discrimination ability of hierarchical clustering on the combined data for **subject / object relative clauses with reflexives**. The numbers in bold are the correct classifications of controls/IWA. The bottom row shows the percentage accuracy.

one respective data set, one for simple relatives, and one for relatives with reflexives. We calculated the dendrogram and cut the tree at 2, because we are only looking for the discrimination between controls and IWA. The results of this are shown in Table 3 and 4. In simple relatives (cf. Table 3), the clustering is able to identify controls better than IWA, but the identification of IWA is better than chance (50%). In relatives with reflexives (cf. Table 4), clustering shows moderate but above chance discrimination ability in subject relatives. In object relatives with reflexives, controls are discriminated barely above chance, while there is an above chance proportion of misclassifications in IWA, demonstrating poor performance of the clustering there. Discriminative ability might improve if all 11 constructions in Caplan et al. (2015) were to be used; this will be investigated in future work.

Discussion

The simulations and cluster analysis above demonstrate overall tighter clustering in parameter estimates for controls, and more variance in IWA. This is evident from the clustering results in Tables 3 and 4. These findings are consistent with the predictions of the small-scale study in Patil et al. (2016). However, there is considerable variability even in the parameter estimates for controls, more than expected based on the results of Patil et al. (2016).

The distribution of non-default parameter estimates (cf. Tables 1 and 2) suggest that all three hypotheses are possible explanations for the patterns in our simulation results: compared to controls, estimates for IWA tend to include higher default action times and activation noise scales, and lower goal activation. These effects generally appear to be more pronounced in object relatives vs. subject relatives. This means that all the three hypotheses can be considered viable candidate explanations. Overall, more IWA than controls display non-default parameter settings. Although there is evidence that many IWA are affected by all three impairments in our implementation, there are also many patients that show only one or two non-default parameter values. Again, this is more the case in object relatives than in subject relatives.

In general, there is evidence that all three deficits are plausible to some degree. However, IWA differ in the degree of the deficits, and they have a broader range of parameter values than controls. Nevertheless, even the controls show a broad range of differences in parameter values, and even though these are not as variable as IWA, this suggests that some of the unimpaired controls can be seen as showing slowed processing, intermittent deficiencies, and resource reduction to some degree.

There are several problems with the current modelling method. First, using the ACT-R framework with its multiple free parameters has the risk of overfitting. We plan to address this problem in three ways in future research. (1) Testing more constructions from the Caplan et al. (2015) data-set might show whether the current estimates are unique to this kind of construction, or if they are generalisable. (2) We plan to create a new data-set analogous to Caplan's, using German as the test language. Once the English data-set has been analysed and the conclusions about the different candidate hypotheses have been tested on English, a crucial test of the conclusions will be cross-linguistic generalisability. (3) We plan to investigate whether an approach as in Nicenboim and Vasishth (2017), using lognormal race models and mixture models, can be applied to our research question.

Second, the use of accuracies as modelling measure has some drawbacks. Informally, in an accuracy value there is less information encoded than in, for example, reading or listening times. In future work, we will implement an approach modelling both accuracies and listening times. Also, counting each parsing failure as 'wrong' might yield overly conservative accuracy values for the model; this will be addressed by assigning a random component into the calculation. This reflects more closely a participant who guesses if he/she did not fully comprehend the sentence.

Lastly, simulating the subject vs. object relative tasks separately yields the undesirable interpretation of participants' parameters varying across sentence types. While this is not totally implausible, estimating only one set of parameters for all sentence types would reduce the necessity of making additional theoretical assumptions on the underlying mechanisms, and allows for easier comparisons between different syntactic constructions. We plan to do this in future work.

Although our method, as a proof of concept, showed that all three hypotheses are supported to some degree, it is worth investigating more thoroughly how different ACT-R mechanisms are influenced by changes in the three varied parameters in the present work. Implementing more of the constructions from Caplan et al. (2015) will, for example, enable us to explore how the different hypotheses interact with each other in our implementation. More specifically, the decision to use the ANS parameter makes the assumption that the high noise levels for IWA influence all declarative memory retrieval processes, and thus the whole memory, not only the production system. Also, as both the GA and ANS parameters lead to higher failure rates, it will be worth investigating in future work whether a more focussed source of noise, such as utility noise, may be a better way to model intermittent deficiencies.

One possible way to delve deeper into identifying the sources of individual variability in IWA could be to investigate whether sub-clusters show up within the IWA parameter estimates. For example, different IWA being grouped together by high noise values could be interpreted as these patients sharing a common source of their sentence processing deficit (in this hypothetical case, our implementation of intermittent deficiencies). We will address this question once we have simulated data for more constructions of the Caplan et al. (2015) data-set.

Acknowledgements

Paul Mätzig was funded by the Studienstiftung des deutschen Volkes. This research was partly funded by the Volkswagen Foundation grant 89 953 to Shravan Vasishth.

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.

Burkhardt, P., Piñango, M. M., & Wong, K. (2003). The role of the anterior left hemisphere in real-time sentence comprehension: Evidence from split intransitivity. *Brain and Language*, *86*(1), 9–22.

Caplan, D. (2012). Resource reduction accounts of syntactically based comprehension disorders. In C. K. Thompson & R. Bastiannse (Eds.), *Perspectives on agrammatism* (pp. 34–48). Psychology Press.

Caplan, D., Michaud, J., & Hufford, R. (2015). Mechanisms underlying syntactic comprehension deficits in vascular aphasia: New evidence from self-paced listening. *Cognitive Neuropsychology*, *32*(5), 283–313.

Caplan, D., & Waters, G. (2005). The relationship between age, processing speed, working memory capacity, and language comprehension. *Memory*, *13*(3-4), 403-413.

Daily, L. Z., Lovett, M. C., & Reder, L. M. (2001). Modeling individual differences in working memory performance: A source activation account. *Cognitive Science*, *25*(3), 315–353.

Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, *19*, 450–466.

Engelmann, F., Jäger, L. A., & Vasishth, S. (2016). *The effect of prominence and cue association in retrieval processes: A computational account.* Retrieved from `https://osf.io/b56qv/`

Engelmann, F., Vasishth, S., Engbert, R., & Kliegl, R. (2013). A framework for modeling the interaction of syntactic processing and eye movement control. *Topics in Cognitive Science*, *5*(3), 452-474.

Hanne, S., Sekerina, I., Vasishth, S., Burchert, F., & Bleser, R. D. (2011). Chance in agrammatic sentence comprehension: What does it really mean? Evidence from Eye Movements of German Agrammatic Aphasics. *Aphasiology*, *25*, 221-244.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, *99(1)*, 122–149.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*(3), 375–419.

Nicenboim, B., & Vasishth, S. (2017). Models of retrieval in sentence comprehension. In *Proceedings of the First Stan Conference, StanCon.*

Novick, J. M., Trueswell, J. C., & Thompson-Schill, S. L. (2005). Cognitive control and parsing: Reexamining the role of Broca's area in sentence comprehension. *Cognitive, Affective, & Behavioral Neuroscience*, *5*(3), 263–281.

Patil, U., Hanne, S., Burchert, F., De Bleser, R., & Vasishth, S. (2016). A computational evaluation of sentence processing deficits in aphasia. *Cognitive Science*, *40*(1), 5–50.

Vasishth, S., Bruessow, S., Lewis, R. L., & Drenhaus, H. (2008). Processing polarity: How the ungrammatical intrudes on the grammatical. *Cognitive Science*, *32*(4), 685–712.

Ambiguity Resolution in a Cognitive Model of Language Comprehension

Peter Lindes (plindes@umich.edu)
John E. Laird (laird@umich.edu)
University of Michigan, 2260 Hayward Street
Ann Arbor, MI 48109 USA

Abstract

The Lucia comprehension system attempts to model human comprehension by using the Soar cognitive architecture, Embodied Construction Grammar (ECG), and an incremental, word-by-word approach to grounded processing. Traditional approaches use techniques such as parallel paths and global optimization to resolve ambiguities. Here we describe how Lucia deals with lexical, grammatical, structural, and semantic ambiguities by using knowledge from the surrounding linguistic and environmental context. It uses a local repair mechanism to maintain a single path, and shows a garden path effect when local repair breaks down. Data on adding new linguistic knowledge shows that the ECG grammar grows faster than the knowledge for handling context, and that low-level grammar items grow faster than more general ones.

Keywords: Natural language understanding; cognitive models; Soar; construction grammar; Embodied Construction Grammar; local repair; ambiguity resolution; garden path effect.

Introduction

In previous work, we described the development of a cognitive model of language comprehension (Lindes and Laird, 2016; 2017), implemented in Soar (Laird, 2012), that incorporates the Embodied Construction Grammar (ECG) cognitive linguistic theory of grammar (Feldman, Dodge, and Bryant, 2009; Bergen and Chang, 2013). A key part of our model is that it attempts to model human comprehension processes. This is done by using parsing that is incremental and word by word, eagerly applying all available knowledge sources at each step, while maintaining a single syntactic and semantic interpretation. Our work is inspired by previous cognitive model-based theories, such as NL-Soar (Newell, 1990; Lehman et al. 1991; Lewis, 1993), and is consistent with the recent "Now–or-Never bottleneck" proposal of Christiansen and Chater (2016).

Traditional natural language processing approaches focus on syntactic analysis of isolated sentences (Hale, 2014). Techniques for resolving ambiguities include multiple parallel paths, using statistics from corpora, global optimization, and producing a ranked list of possible parses. These methods lack contextual knowledge to resolve ambiguities to produce accurate, grounded meanings in context. Their success is at the cost of relaxing constraints imposed by an incremental model of human processing.

Although our system, called Lucia, has been successful in supporting language understanding for an embodied robotic agent (Lindes and Laird, 2016), a significant question is whether incremental, word-by-word approaches can handle the many types of ambiguity that can arise in language understanding. Parsers developed for ECG (Bryant, 2008) and Fluid Construction Grammar (FCG; Steels and Hild, 2012) do not attempt to model incremental parsing, but instead treat parsing as optimization over a complete sentence, with no commitment to word-by-word processing. Thus, these other approaches do not treat the issues of dealing with ambiguity that arise in incremental parsing.

In this paper, we explore the problem of ambiguity in incremental language processing. We build on previous work by Lewis (1993), where local repair is used to recover from some types of syntactic ambiguity, but we extend this to other forms of lexical, grammatical, structural, and semantic ambiguity, taking advantage of the contextual knowledge that is available during processing. Comparison to detailed human performance data is outside the scope of our current research. In the following, we discuss the basic operation of the system, and explore how it deals with different ambiguous situations.

Basic Comprehension

Lucia is built within a Soar agent called Rosie (Mininger and Laird, 2016) that learns new tasks involving robotic object manipulation and navigation. It uses a grammar for a domain-specific subset of English written in the formal language of ECG (Bryant, 2008). A program translates the ECG grammar into Soar production rules that we call G rules. Another set of Soar rules that connect to the embodied context of the agent, are written by hand, and are called C rules. Together these rules process language input to produce meaningful messages that Rosie uses to perform actions and learn new tasks.

Grammars in the ECG language are made up of two kinds of "items:" constructions and schemas. Each schema defines the structure of a certain kind of meaning element and defines its "roles" or "slots." A construction is a pairing of a form with a meaning. There are three types of ECG constructions. Lexical constructions (L cxns) recognize input words. Phrasal constructions (P cxns) combine one or more constituents already recognized into a higher-level structure. General constructions (G cxns) do not recognize specific forms, but augment instances of other constructions that are marked as their subcases. Any construction can evoke a schema to represent its meaning and provide constraints to specify how to populate the slots of the schema.

Semantic parsing is carried out incrementally, with processing done greedily for each word, as in the incremental approach called "Chunk-and-Pass," which Christiansen and Chater (2016) claim models human comprehension. The basic operation is a word cycle in which a new word is received, a lexical access operator retrieves one or more senses of that word (L cxns), and then further processing is performed. The further processing includes operators that recognize and apply phrase level constructions (P cxns) and operators that ground the meanings built from the grammar to the perceptions and actions of the agent using C rules.

The current state of the parse is represented by a stack in working memory that contains a sequence of construction instances that have been recognized but not yet incorporated as constituents of a higher level construction. During lexical access, one or more L cxn instances are added to the current state. Then a P cxn that matches the current state, if any, creates a new instance of itself on the stack, removing its constituents from the stack and adding them as its children, to form a new "chunk." This can happen several times in a single word cycle. When a construction instance is created, its corresponding meaning structure is also built. These meaning structures trigger grounding operators that look for something to ground this meaning, either in the agent's perceptual model or its general background knowledge.

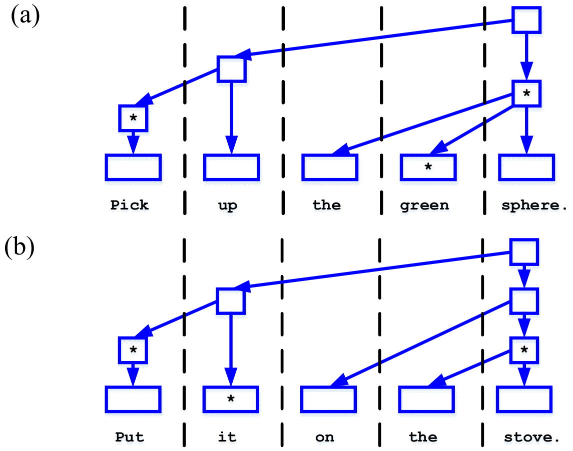


Figure 1: Examples of word-by-word comprehension.

Figure 1 shows some example parses. The word processing cycles are separated by vertical dotted lines. Each rectangle is a construction instance, with L cxns shown larger. An asterisk means a grounding operator was used. Meaning structures are not shown. Within each cycle, operators are executed from the bottom up. When the whole sentence has been processed and the result is a single construction instance, that construction is interpreted to produce a message to tell the robot what to do. If the processing does not produce a single result, the parse fails.

The Lucia comprehender has been applied to a corpus of several hundred sentences previously used with the Rosie system. The grammar and context rules have been developed sufficiently to correctly comprehend 130 of those sentences. A variety of sentential forms are comprehended, including the examples in (1).

(1) a. `The sphere is green.`
b. `Store the large green sphere on the red triangle.`
c. `Pick a green block that is larger than the green box.`
d. `Drive to the wall.`
e. `Go until there is a doorway.`
f. `If the green box is large then go forward.`
g. `What is inside the pantry?`
h. `Where is the red triangle?`
i. `Is the large triangle to the right of the green sphere?`
j. `Drive down the hall until you reach the end.`
k. `Fetch a soda.`

A variety of declarative, interrogative, and imperative sentences are handled, including ones with relative clauses and conditional clauses. In many of the 130 sentences, various kinds of lexical, syntactic, and semantic ambiguities must be handled. Below we examine some of these cases.

Handling Ambiguities

Here we analyze how Lucia handles instances of lexical, grammatical, structural, and semantic ambiguities, as well as garden path sentences. For each type of ambiguity, we give some specific examples and show how Lucia resolves them using different types of contextual knowledge within its incremental, word-by-word approach to comprehension.

Lexical Ambiguities

Lucia has several strategies for dealing with words that have different meanings depending on the context.

Resolution by Syntactic Context Many function words have meanings that vary depending on the syntactic context. For example, *up* can be a particle together with a verb as in *pick up*, or it can be a preposition. Various forms of *to be*, such as *is*, have many possible uses. When possible, Lucia uses the strategy of having a single construction for the word defined in the grammar and instantiated during lexical access, and then resolving the correct meaning from the syntactic context by what phrasal construction uses that word. This follows the principle in construction grammar theory that both words and larger constructions contribute to meaning (Goldberg, 1995). Consider some of the many uses of *is* in (2):

(2) a. `The sphere` *`is`* `green.`
b. `The red triangle` *`is`* `on the stove.`
c. `Go until there` *`is`* `a doorway.`
d. *`Is`* `the large orange block a sphere?`

Is can declare an object property (2a) or a relation (2b). With *there, is* can declare the existence of something (2c). *Is* can also introduce a question (2d). None of this information is derived during lexical access, but is added as phrasal constructions are recognized.

Multiple Senses, Immediate Resolution Content words often have multiple senses, with context needed to select from them. In these cases, the grammar defines two or more alternative lexical constructions. A phrasal construction that recognizes one of them chooses that one and deletes the others, as in (3):

(3) a. `The` *`sphere`* `is` *`red`*`.`
b. `Where is the` *`red`* `triangle?`
c. `Is this a` *`sphere`*`?`

These three sentences show different senses for both *sphere* and *red*. *Sphere* produces two senses, a noun and a class name. The noun sense is recognized by one P cxn in (3a), while *a sphere* in (3c) is recognized by a different P cxn that uses the class sense, discarding the noun. In both (3a) and (3b) *red* is recognized as a property, but in (3a) it is declared to apply to *the sphere,* while in (3b) it is used as an adjective to modify *triangle*.

That can be deictic (4a) to refer to something being pointed to, or can be used to introduce a relative clause (4b). Both senses are generated in lexical access. A P cxn that matches the context then selects one of the senses and deletes the other.

(4) a. `Put` *`that`* `in the pantry.`
b. `Pick up the green block` *`that`* `is on the stove.`

Multiple Senses, Delayed Resolution The word *square* can be a property to be applied, a noun, or an adjective:

(5) a. `This is a` *`square`*`.`
b. `Put the` *`square`* `in the` *`square`* `box.`

All three senses are generated by lexical access each time. For a property application as in (5a), that sense is chosen by a P cxn and the others discarded. In the first case in (5b), the noun is chosen similarly.

The second case in (5b) is more complicated: in processing this instance of *square,* the noun will be chosen as before. When *box* is being processed, the system recognizes that the chosen sense is wrong, and an operator called *snip* is selected, which deletes the P cxn for *the square.* Next, the previously discarded adjective sense of *square* replaces the noun sense. Now the whole phrase *the square box* can be recognized. Many nouns can be used as adjectives like this.

The case of *square* as an adjective illustrates the delayed resolution strategy. In immediate resolution, other senses are not completely forgotten; they are linked to the chosen sense and can be brought back and selected in a later context. This is one kind of repair process that makes incremental parsing possible. These strategies make it possible for the comprehender to maintain only a single path in its parse state, yet still have enough information available to make a local repair when necessary.

Resolution by Semantic Context Some lexical ambiguities must be resolved by semantic rather than syntactic context. The meaning of *bank*, for example, depends on whether the semantic context is related to rivers or finances. Lucia has access to semantic information, both in the part of the sentence that has already been processed and in the more general discourse context. At the moment, none of the sentences we have worked with have needed this kind of resolution, but this can be easily added when needed.

Grammatical Ambiguities

Lucia uses one of two strategies when multiple phrasal constructions match a given parse state. The first is simple: when two different constructions match at the same time, if one matches more constituents than the other, then the more specific one (the one with the greater span) is chosen. When processing *sphere* in Figure 1a, either the noun by itself could be recognized or the phrase *the green sphere*. The longer, more specific match is preferred to the shorter, more general one.

There are cases where two constructions with the same span match the same parse state. In order to choose a more specific option over a more general one in these cases, there are preference rules to select the more specific one.

(6) a. *`The`* `sphere is green.`
b. `This is` *`a`* `sphere.`

In (6) we have two phrases with *sphere*. Either could be recognized by a noun phrase construction, but in (6b) the phrase should be interpreted as a property that can be applied to the subject of the sentence rather than a noun phrase to ground to an object. Two preference rules, one for a definite and one for an indefinite determiner, make the distinction.

Structural Ambiguities

Often the immediate context suggests one way of integrating a word into the ongoing parse, but later on that decision turns out to be wrong, as in *the square box* where the word *square* should be an adjective and not a noun. Of particular importance are the attachment of prepositional phrases and relative or subordinate clauses. Lucia implements a strategy of local repair, similar to that used by Lewis (1993), to resolve these ambiguities, as the following examples show.

(7) a. `Pick up the green block on the stove.`
b. `Put the green sphere in the pantry.`
c. `Pick up the green block that is on the stove.`

d. `Put the green block that is on the stove in the pantry.`

e. `Move the green rectangle to the left of the large green rectangle to the pantry.`

Sentence (7a) appears to be complete after processing *block*. However, there are more words. After processing *stove,* there is a prepositional phrase that could either modify *the green block* or provide a target location for the verb. In this case, it should modify the noun phrase, since *pick up* does not expect a target location. However, that noun phrase has already been consumed by the clause construction and is no longer available on the stack as a constituent, so the system is at an impasse. What can be done?

The answer is a variant of the *snip* operator described earlier, which was introduced by Lewis (1993). This version deletes the clause construction to expose the noun phrase for *the green block* on the stack. Then that noun phrase is combined with the prepositional phrase to form a new referring expression that is grounded to that particular green block, which happens to be on the stove. Figure 2 shows two steps of this process.

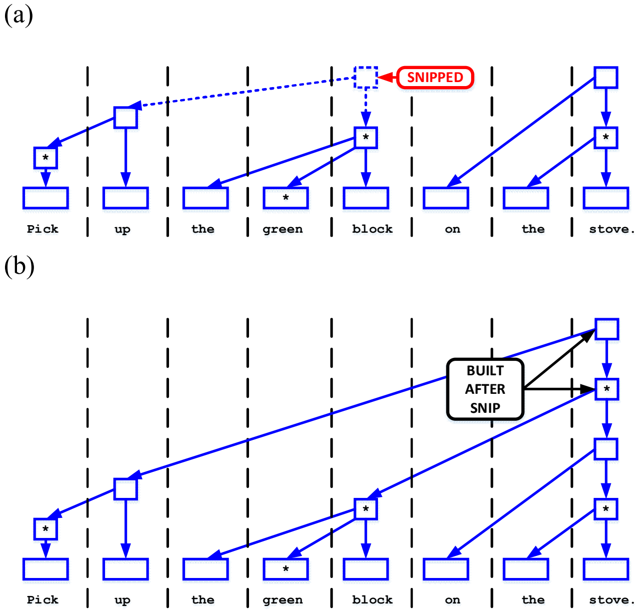


Figure 2: A local repair using *snip*

Figure 2a shows the state of the parse when we reach the impasse. At this point, a *snip* is performed to delete the clause construction shown with dotted lines, allowing the creation and grounding of the expression for *the green block on the stove*, as in Figure 2b. Finally, a new clause construction is created with this new referring expression.

Another aspect of grounded comprehension is shown by (7a). *The green block* is first grounded to a set of four green blocks that all exist in the current environment. If the sentence ended here, the comprehender would have two choices: either pick one of the four at random or report that it sees four possible meanings and ask for clarification. However, when the full expression *the green block on the stove* has been processed, grounding yields a single green block, which is currently on the stove. This shows an example of resolving ambiguous semantics through grounding.

Semantic Ambiguities

The current Lucia system resolves several problems using semantic information built into its grammar. One example is the different prepositional phrase attachments chosen for sentences (7a) and (7b). The two verbs *pick up* and *put* are not simply processed as instances of some general verb part of speech. Instead, distinct meaningful constructions for the two verbs are treated differently in the grammar, causing one to require a prepositional phrase and the other not. This is an example of how grammatical constructions, not just lexical items, carry meaning, as Goldberg (1995) insists.

Prepositions give another interesting example of this effect. Consider the two sentences in (8).

(8) a. `Go` *`to`* `the kitchen.`

b. `Go` *`down`* `the hall.`

Most generative grammar approaches produce the same exact grammatical structure for both of these sentences. Such an approach fails in an incremental semantic parse that must produce actionable meanings. The final messages that are to be sent to the robot for these two sentences are different. For (8a), the message specifies a specific waypoint as the goal of the *go* action, whereas for (8b) no specific goal is given, just an object representing *the hall* to guide the motion.

When sentence (8b) was first encountered while building Lucia's grammar, we realized that not all prepositions are the same. Consider a number of other possible prepositions that could have appeared in one of these sentences: *across*, *along*, *around*, *behind*, *in*, *into*, *out of*, *past*, *through*, *to the left of*, and so on. Some of these would work perfectly well in one of the sentences while making the other infelicitous[1]. Whether some of these make sense in certain sentences may depend on the noun that follows or the main verb of the sentence. Each of these prepositions seem to describe a trajectory in space, which may or may not have a terminating point. An interesting mental exercise is to try to imagine a diagram of the trajectory expected for each of the prepositions listed in each of the given sentences or in a similar one.

To deal with this problem, some refactoring was done in the part of the grammar dealing with prepositions. In (8a), *to* is treated as an ordinary preposition. For *down* in (8b) we created a new construction that can only be a constituent of a corresponding special subcase of a prepositional phrase. These constructions provide an alternative way of parsing

[1] Linguists use the term *infelicitous* to describe a sentence which is syntactically correct but does not make sense semantically.

depending on the particular preposition involved, which then allows building a different meaning structure.

This is another example of constructions carrying meaning, and shows key characteristics of a constructionist approach to grammar. In this approach we seek to define many specific constructions to build meaning into the grammar, rather than a minimal number of meaningless phrase labels to cover the language. This fits with psychological theories of children's language acquisition that emphasize children learning very specific constructions first and then gradually generalizing them (Tomasello, 2003).

Garden Path Sentences

"Garden path sentences" are grammatically correct, but are difficult for humans to parse correctly, at least at first. It appears that humans make a wrong decision early on in the parse, and later on, no local repair mechanism is sufficient to correct the problem. The Lucia theory produces this effect as we see with (9).

(9) `The horse raced past the barn fell.`

Lewis (1993) provides a theory of garden paths. He describes three possible causes: there is a lack of structural cues to trigger repair, the syntactic relation that needs to be altered is no longer available, or the system has not learned an alternative solution through previous deliberation.

The Lucia analysis of this sentence is consistent with this theory, as shown in Figure 3. First, *the horse raced* looks like a whole sentence using the past tense of *race* and discarding its past participle sense. Later a correct parse is found for *The horse raced past the barn*. Now when *fell* arrives, there is no way to integrate it into the sentence, because of the wrong choice that was made to use *raced* as a simple past tense verb rather than a past participle. This creates a garden path effect.

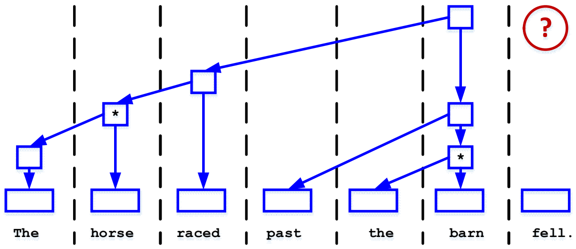


Figure 3: A garden path sentence.

Why does local repair not work here? Because when the system gets to the impasse, the change that needs to be made is at *raced,* which is two layers back on the stack and two layers deep in the hierarchy. This is not local enough for local repair to work, consistent with Lewis's second reason.

If the grammar only has the past participle sense of *raced*, Lucia produces a correct analysis. A deliberative repair process might produce the correct parse. Neither humans nor Lucia can do this as part of automatic parsing.

Taken together, the examples above show that an incremental comprehension system can resolve many lexical, grammatical, structural, and semantic ambiguities, and at the same time produce garden path effects.

Adding to Linguistic Knowledge

Currently, Lucia has no mechanism for learning new vocabulary, new phrasal constructions, or new concepts. The principle that meaningful language relies on many very specific constructions organized in a network with some generalities (Goldberg, 2006), rather than a few general rules, suggests that adding linguistic knowledge by hand will not scale up to something approaching general human language. Thus, even if our comprehension mechanisms are sufficient, the system will be limited in its application if it is unable to acquire new language. A means of acquisition is an essential goal for future work.

However, by analyzing Lucia's development, we can make some predictions about learning. In Lucia, the linguistic knowledge has grown incrementally. To process each new sentence, we coded new constructions and schemas in ECG and added new context rules when necessary. We expect that the G rules, which encode items in the grammar, would grow faster than the C rules which perform contextual processing.

Figure 4 shows how the number of Soar production rules of each type grew as the number of sentences comprehended grew from 42 to 130. Many more grammar rules than context rules were added, and the number of grammar rules grew more rapidly than the number of context rules.

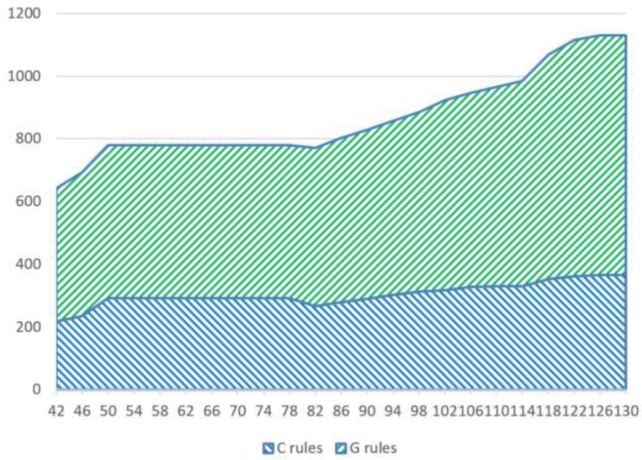


Figure 4: Growth of C & G rules as language coverage increases.

Figure 5 gives a different perspective on this growth data. Here we show the growth in ECG items, both constructions and schemas. Constructions are further broken down into lexical constructions (L cxns), phrasal constructions (P cxns), and general constructions (G cxns). We see that lexical constructions and schemas are growing faster than the more general construction types, confirming that the more specific items grow faster.

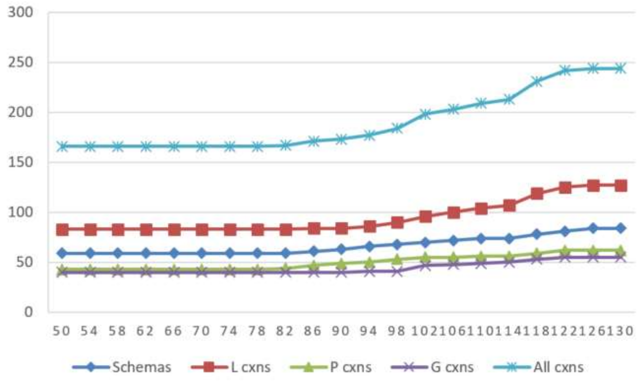


Figure 5: Growth of ECG items as language grows.

Conclusions

The results from Lucia are consistent with the claim that a comprehension system using a human-like, integrated, incremental parsing approach, within a cognitive architecture, and with construction grammar, can incrementally resolve a variety of linguistic ambiguities. They also are consistent with one type of breakdown that arises in garden path sentences in a way similar to humans.

How scalable is this approach? There are many linguistic forms that it does not handle: past and future tenses, auxiliary verbs, conjunctions, metaphor, and on and on. Nevertheless, as Figures 4 and 5 show, as new forms have been addressed, most of the new knowledge required has been expressible in the ECG grammar and has not required changes to the underlying context rules

The techniques we have described for handling ambiguity, however, depend mostly on the context operators. They provide grounded semantics, select among grammatical alternatives, and perform local repairs. This is consistent with the theory that human-like comprehension relies heavily on context to resolve ambiguities.

The current approach requires coding context rules by hand. In the future, we will attempt to enhance the ECG language to encode contextual constraints and/or use context-dependent retrievals with spreading activation in long-term declarative memory (Jones et al., 2016).

Also in future work we intend to explore comparing detailed processing data from Lucia to the large amount of available human performance data, and to datasets other than the Rosie sentence corpus we have considered here.

Acknowledgment

The work described here was supported by the AFOSR under Grant Number FA9559-15-1-0157. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the AFOSR or the U.S. Government.

References

Bergen, B. and N. Chang (2013). Embodied Construction Grammar. In Thomas Hoffman and Graeme Trousdale, eds, *The Oxford Handbook of Construction Grammar*. Oxford University Press, New York.

Bryant, J. E. (2008). *Best-Fit Constructional Analysis*. PhD dissertation in Computer Science, University of California at Berkeley.

Christiansen, M. H. and N. Chater (2016). The Now-or-Never bottleneck: A fundamental constraint of language. *Behavioral and Brain Sciences (2016)*, 1-72.

Feldman, J., E. Dodge, and J. Bryant (2009). Embodied Construction Grammar. In *The Oxford Handbook of Linguistic Analysis*, Bernd Heine and Heiko Narrog eds. Oxford Handbooks Online.

Goldberg, A. E. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. The University of Chicago Press.

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language.* Oxford University Press.

Hale, J. T. (2014). *Automaton Theories of Human Sentence Processing*. Stanford: CLSI Publications.

Jones, S. J., A. R. Wandzel, and J. E. Laird (2016) Efficient Computation of Spreading Activation Using Lazy Evaluation. *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM)*.

Laird, J. E. (2012). *The Soar Cognitive Architecture*. Cambridge, MA: The MIT Press.

Lehman, J. F., R. L. Lewis, and A. Newell (1991). Integrating Knowledge Sources in Language Comprehension. *The Soar Papers, Vol. II*. Cambridge, MA: MIT Press.

Lewis, R. L. (1993). *An Architecturally-based Theory of Human Sentence Comprehension*. PhD dissertation in Computer Science, Carnegie Mellon University.

Lindes, P. and J. E. Laird (2016). Toward Integrating Cognitive Linguistics and Cognitive Language Processing. *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016*.

Lindes, P. and J. E. Laird (2017). Cognitive Modeling Approaches to Language Comprehension Using Construction Grammar. Accepted for publication with the *AAAI Spring Symposium 2017* track on *Computational Construction Grammar and Natural Language Understanding.*

Mininger, A. and J. E. Laird (2016). Interactively Learning Strategies for Handling References to Unseen or Unknown Objects. *Advances in Cognitive Systems 4*.

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

Steels, L. and M. Hild, eds. (2012). *Language Grounding in Robots.* Springer.

Tomasello, M. (2003). *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Cambridge, MA.

Feature overwriting as a finite mixture process: Evidence from comprehension data

Shravan Vasishth (vasishth@uni-potsdam.de)
Department of Linguistics, University of Potsdam, Germany

Lena Jäger (lena.jaeger@uni-potsdam.de)
Department of Linguistics, University of Potsdam, Germany.

Bruno Nicenboim (bruno.nicenboim@uni-potsdam.de)
Department of Linguistics, University of Potsdam, Germany.

Abstract

The ungrammatical sentence *The key to the cabinets are on the table* is known to lead to an illusion of grammaticality. As discussed in the meta-analysis by Jäger et al., 2017, faster reading times are observed at the verb *are* in the agreement-attraction sentence above compared to the equally ungrammatical sentence *The key to the cabinet are on the table*. One explanation for this facilitation effect is the feature percolation account: the plural feature on *cabinets* percolates up to the head noun *key*, leading to the illusion. An alternative account is in terms of cue-based retrieval account (Lewis & Vasishth, 2005), which assumes that the non-subject noun *cabinets* is misretrieved due to a partial feature-match when a dependency completion process at the auxiliary initiates a memory access for a subject with plural marking. We present evidence for yet another explanation for the observed facilitation. Because the second sentence has two nouns with identical number, it is possible that these are, in some proportion of trials, more difficult to keep distinct, leading to slower reading times at the verb in the first sentence above; this is the feature overwriting account of Nairne, 1990. We show that the feature overwriting proposal can be implemented as a finite mixture process. We reanalysed ten published data-sets, fitting hierarchical Bayesian mixture models to these data assuming a two-mixture distribution. We show that in nine out of the ten studies, a mixture distribution corresponding to feature overwriting furnishes a superior fit over both the feature percolation and the cue-based retrieval accounts.

Keywords: Feature overwriting; feature percolation; cue-based retrieval; sentence processing; interference; reading; Bayesian hierarchical mixture models

Introduction

It is well-known that sentences such as (1a) can lead to an illusion of grammaticality. The sentence is ungrammatical because of the lack of number agreement between the subject *key* and the auxiliary *are*. Note that the second noun, *cabinets*, and the auxiliary *are* agree in number, but no syntactic agreement is possible between these two elements.

(1) a. The key to the cabinets are on the table.
b. The key to the cabinet are on the table.

Many sentence comprehension studies have shown that the illusion has the effect that the auxiliary *are* is read faster in (1a) compared to the equally ungrammatical sentence (1b) (see Jäger, Engelmann, & Vasishth, 2017 for a review). In contrast to (1a), in (1b) the second noun (*cabinet*) is singular and does not agree with the auxilary in number.

Several explanations have been proposed for the illusion of grammaticality in (1a) vs. (1b). We discuss two of these here. The feature percolation account proposes that in (1a) the plural feature on *cabinets* can, in some proportion of trials, move or percolate up to the head noun *key* (see Patson & Husband, 2016 for recent evidence for this model). The head noun now has the plural feature, leading to an illusion of grammaticality compared to (1b), where no such feature percolation occurs. Another prominent explanation, due to Wagers, Lau, and Phillips (2009), is the retrieval interference account. Here, in ungrammatical sentences like (1a), a singular verb would be predicted; but when the plural verb *are* is encountered, a cue-based retrieval process (Lewis & Vasishth, 2005) is triggered: The verb triggers an access (called a retrieval) for a noun that is plural marked and is a subject. A parallel cue-based associative memory access leads to the retrieval of a partially matching noun in memory (*cabinets*) that agrees in number but is not the subject. This partial match leads to a successful retrieval and an illusion of grammaticality.[1]

As we show next, there is evidence for both these accounts: a facilitatory effect is generally present in the published data.

The facilitatory effect in reading time in the "illusion of grammaticality" data-sets

We first establish that a facilitatory effect is found in studies comparing sentences like (1a) and (1b). In connection with the meta-analysis relating to studies on cue-based retrieval reported in Jäger et al. (2017), we had obtained the raw data from 10 studies on sentences like (1a) and (1b). These were reading-time studies reported in Dillon, Mishler, Sloggett, and Phillips (2011), Lago, Shalom, Sigman, Lau, and Phillips (2015), and Wagers et al. (2009). Except for the eyetracking experiment by Dillon and colleagues, all the other studies were self-paced reading experiments. In these data-sets, the dependent measure was reading time in milliseconds at the auxiliary or the region following it. Most of the 10 studies

[1] The cue-based retrieval account may a priori be implausible because it predicts that an incorrect dependency is built between *cabinets* and *are*; building such a dependency would imply that the sentence has the implausible meaning that the cabinets are on the table. The reader should detect such an implausible meaning and this should lead to a slowdown rather than facilitation.

found statistically significant effects in this post-critical region. What is noteworthy here is the consistently negative sign of the effect of interest; this consistency is much more informative than the statistical significance of individual studies.

We first reanalyzed these 10 data-sets in order to confirm the facilitatory effect reported.[2] We fit Bayesian hierarchical models to each data-set using Stan (Stan Development Team, 2016). We fit Bayesian models because of the ease with which statistical models can be defined flexibly to reflect the cognitive process of interest.

The model specification was as follows. Assume that (i) i indexes participants, $i = 1, \dots, I$ and j indexes items, $j = 1, \dots, J$; (ii) y_{ij} is the reading time in milliseconds for the i-th participant reading the j-th item; and (iii) the predictor x, which represents the experimental manipulation, is sum-coded (± 1). In our case, the condition (1a) is coded $+1$ and the condition (1b) is coded -1.

Then, the data y_{ij} (reading times in milliseconds) are defined to be generated by the following process:

$$y_{ij} \sim LogNormal(\beta_1 + \beta_2 x_{ij} + u_i + w_j, \sigma_e^2) \tag{1}$$

where $u_i \sim Normal(0, \sigma_u^2)$, $w_j \sim Normal(0, \sigma_w^2)$ and σ_e^2 is the error variance. The terms u_i and w_j are called varying intercepts for participants and items respectively; they represent by-subject and by-item adjustments to the fixed-effect intercept β_1. The variances σ_u^2 and σ_w^2 represent between-participant (respectively item) variance.[3] The facilitation effect is the estimate of β_2 (on the log scale).

As priors, we chose the Cauchy(0,2.5) distribution for all coefficients, and a half-Cauchy (with only positive values) for the standard deviations. This are mildly informative priors (Gelman et al., 2014) which express the belief that that the most likely value of the parameter is near 0, but allows for a wide range of non-zero values because of the fat tails of the Cauchy.

As shown in Figure 1, the effects in each study consistently show negative estimates of β_2, which indicates a facilitation in reading time at the auxiliary or a subsequent region. This is consistent with both the feature percolation and retrieval interference accounts. There is a third explanation for the observed facilitation effect in these studies, which we turn to next.

An alternative explanation for the facilitatory effect Consider the ungrammatical example sentences again. These are repeated below for convenience:

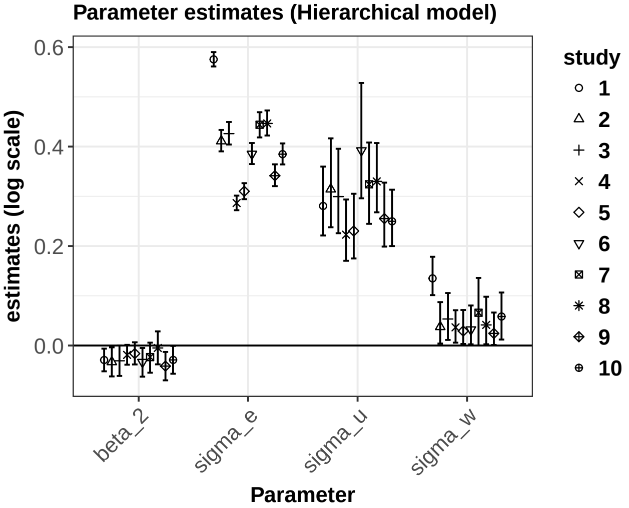


Figure 1: The parameter estimates of the hierarchical model fitted to the 10 data-sets. The condition representing (1a) is coded $+1$ and the condition representing (1b) is coded -1, so that parameter `beta_2` shows a facilitation effect if its value is negative. Shown are the estimates of the facilitatory effect (`beta_2`), and the standard deviations of (i) the error (`sigma_e`), (ii) the by-subjects varying intercepts (`sigma_u`), and (iii) the by-items varying intercepts (`sigma_w`).

(2) a. The key to the cabinets are on the table.
 b. The key to the cabinet are on the table.

In example (2b), both the nouns are marked singular, whereas in example (2a) the nouns have different number marking. As discussed in Villata and Franck (2016), the similarity in number of the two nouns in (2b) could be the underlying cause for increased processing difficulty, compared to (2a). The identical number marking in (2b) could lead to increased confusability between the two nouns, leading to longer reading times at the moment when a subject noun is to be accessed at the auxiliary verb. The feature overwriting model of Nairne (1990) formalizes this idea. To quote (p. 252): *An individual feature of a primary memory trace is assumed to be overwritten, with probability F, if that feature is matched in a subsequently occurring event. Interference occurs on a feature-by-feature basis, so that, if feature b matches feature a, the latter will be lost with probability F.*

The Nairne proposal has a natural interpretation as a finite mixture process. Specifically, feature overwriting could occur with a higher probability in example (2b) compared to (2a). This assumption implies that the reading times in both (2b) and (2a) are generated from a mixture of two distributions. In a particular trial, if no feature overwriting occurs, the reading time would come from a Lognormal distribution with some location and scale parameters; this situation would result in minimal processing difficulty in carrying out a retrieval and detecting the ungrammaticality. In other trials, when feature overwriting does occur, the reading time would have a larger

[2] The published studies had other experimental conditions that we do not discuss here. The published studies also used a trimming procedure to analyze the data, and their analysis was done on the raw millisecond scale. Thus, our analysis has some differences from the original analyses, but the conclusions are substantially unchanged.

[3] This so-called crossed participants and items varying intercepts linear mixed model can be made more complex by adding varying slopes for the factor X by participant and by item, but for space reasons we do not consider these more complex models here.

location parameter, and possibly also a larger scale parameter; this would represent the cases where additional difficulty occurred due to feature overwriting.[4]

An explicit assumption here is that feature overwriting could occur in both (2b) and (2a), but the proportion would be higher in (2b). It is also possible to assume that feature overwriting only occurs in (2b), but due to space reasons we do not consider this and other alternative models here.

Thus, in the mixture model implementation of the Nairne proposal, one distribution will have a larger location parameter (and perhaps also the scale parameter). In the modelling presented below, one goal is to estimate the mixing proportions of these distributions. In the results section, we will refer to the proportion of the slow reading time distributions in (2b) as `prob_hi`, and in (2a) `prob_lo`. The suffixes `hi` and `lo` here refer to whether we expect confusability to be high or low.

To summarize, the feature percolation, cue-based retrieval, and feature overwriting models all predict facilitation in the ungrammatical sentences (2a) compared to (2b), but the underlying generative process assumed in each model is different. Feature percolation and feature overwriting can be seen as finite mixture models of different types, and cue-based retrieval can be seen as implemented by the standard hierarchical model. Our goal here is to implement all the three proposals as statistical models and then compare their relative fit to the data in order to adjudicate between them. Before we do this, we introduce finite mixture models.

Finite mixture models

A finite mixture model assumes that the independently distributed outcome $y_i, i = 1, \ldots, N$ is drawn from one of several distributions. Each distribution's identity is controlled by a Categorical distribution. For example, assume that we have K distributions with location parameter (the mean) $\mu_k \in \mathbb{R}$ and scales (standard deviation) $\sigma_k \in (0, \infty)$, where $k = 1, \ldots, K$. Assume also that we have a vector of probabilities $< \lambda_1, \ldots, \lambda_K >= \Lambda$ that represent the mixing proportions. The parameters λ_k are non-negative values and they sum to 1.

Thus, if the K distributions are mixed in proportion Λ, where $\lambda_k \geq 0$ and $\sum_{k=1}^{K} \lambda_k = 1$, for each outcome y_i there is a latent variable $z_i \in \{1, \ldots, K\}$ with a Categorical distribution[5] parameterized by $\lambda : z_i \sim \texttt{Categorical}(\lambda)$. The variable y_i is then distributed as follows:

$$y_i \sim Normal(\mu_{z_i}, \sigma^2_{z_i}) \qquad (2)$$

Assuming that each of the K mixture distributions $f(\cdot)$ has a vector of parameters θ_k associated with it, the mixture density can be written in the following manner:

$$p(y_i \mid \theta, \Lambda) = \lambda_1 \cdot f(y_i \mid \theta_1) + \cdots + \lambda_K \cdot f(y_i \mid \theta_K) \qquad (3)$$

A random variable Y with the above density can then be written in abbreviated form as follows.

$$Y \sim \lambda_1 f(y \mid \theta_1) + \cdots + \lambda_K f(y \mid \theta_K) \qquad (4)$$

In this paper, we consider a mixture of LogNormals with $K = 2$; this is because the feature overwriting model assumes a mixture of two distributions. We choose LogNormals to model reading times because reading times must be greater than 0 and follow a LogNormal distribution. We will write the models as follows:

$$\begin{aligned} Y \sim & \lambda_1 \cdot LogNormal(\mu_1 + \delta, \sigma_1^2) + (1 - \lambda_1) \cdot LogNormal(\mu_1, \sigma_2^2) \\ & \text{where } \sigma_1^2 = \sigma_2^2 \text{ or } \sigma_1^2 \neq \sigma_2^2 \end{aligned} \qquad (5)$$

The parameter δ marks the shift in the mean in the first mixture distribution relative to the second mixture distribution. Note that the scale parameters (σ_1, σ_2) can be either identical (homogeneous variances) in both distributions, or different (heterogeneous variances). We will consider both types of models here.

The above models assume that the data are independent. When we have repeated measures data, the independence assumption is no longer valid. In order to address this issue, finite mixture models can be made hierarchical by adding varying intercepts for subjects (indexed by i) and items (indexed by j):

$$\begin{aligned} y_{ij} \sim & \lambda_1 \cdot LogNormal(\mu_1 + \delta + u_i + w_j, \sigma_1^2) + \\ & (1 - \lambda_1) \cdot LogNormal(\mu_1 + u_i + w_j, \sigma_2^2) \end{aligned} \qquad (6)$$

where $u_i \sim Normal(0, \sigma_u^2)$ and $w_j \sim Normal(0, \sigma_w^2)$. Thus, the mixture model with $K = 2$ will have the following parameters: four variance components, $\sigma_1^2, \sigma_2^2, \sigma_u^2$, and σ_w^2; two coefficients μ_1 and δ; and two probabilities λ_1 and $\lambda_2 = (1 - \lambda_1)$.

An evaluation of the Nairne feature overwriting proposal

Method

Implementing the Nairne proposal We fit the homogeneous and heterogeneous variance hierarchical mixture models to the 10 reading time data-sets that compared reading times at the auxiliary or the following region for sentences like (2a) and (2b).

The data were assumed to be generated from a two-mixture Lognormal distribution with either a homogeneous variance in both mixture distributions, or heterogeneous variances.

[4] In grammatical sentences like *The key to the cabinet/s is...*, both feature overwriting and cue-based retrieval predict a slowdown when the nouns have the same number. The literature largely shows no difference in reading time. But the two models' relative performance can still be investigated; we plan to do this in future work.

[5] The Categorical distribution can be seen here as the Bernoulli distribution in the case where K=2. In this paper, we focus only on the K=2 case.

Thus, for the high confusability condition (2b), we considered two models:

$$\underline{\text{Homogeneous variance feature overwriting model}}$$

$$\begin{aligned}
y_{ij} \sim & \text{prob\_hi} \cdot \mathit{LogNormal}(\beta + \delta + u_i + w_j, \sigma_e^2) + \\
& (1 - \text{prob\_hi}) \cdot \mathit{LogNormal}(\beta + u_i + w_j, \sigma_e^2) \\
& \text{where:} \\
& u_i \sim \mathit{Normal}(0, \sigma_u^2), w_k \sim \mathit{Normal}(0, \sigma_w^2)
\end{aligned} \tag{7}$$

$$\underline{\text{Heterogeneous variance feature overwriting model}}$$

$$\begin{aligned}
y_{ij} \sim & \text{prob\_hi} \cdot \mathit{LogNormal}(\beta + \delta + u_i + w_j, \sigma_{e'}^2) + \\
& (1 - \text{prob\_hi}) \cdot \mathit{LogNormal}(\beta + u_i + w_j, \sigma_e^2) \\
& \text{where:} \\
& u_i \sim \mathit{Normal}(0, \sigma_u^2), w_k \sim \mathit{Normal}(0, \sigma_w^2)
\end{aligned} \tag{8}$$

In both models, y_{ij} is the reading time in milliseconds from subject i and item j. The probability `prob_hi` represents the mixing probability of the distribution that generates the slow reading times corresponding to trials where feature overwriting occurred (2b). Although not shown, another mixture distribution is defined for example (2a); here, `prob_lo` represents the mixing probability of the distribution that generates the slower reading times corresponding to the trials where feature overwriting occurred.

The homogeneous variance model assumes that both mixture distributions have the same standard deviation σ_e. The heterogeneous mixture model assumes that the mixture distribution that leads to the slower reading times is assumed to have both a different mean ($\beta + \delta$) and a different standard deviation ($\sigma_{e'}$) than the other distribution. Alternative models can be fit which relax these assumptions, but due to space constraints we consider only these two models.

We had the following priors for the parameters:

$$\begin{aligned}
\text{prob\_hi} \sim & \mathit{Beta}(1,1) \\
\beta, \delta \sim & \mathit{Cauchy}(0, 2.5) \\
\sigma_e, \sigma_{e'}, \sigma_u, \sigma_w \sim & \mathit{Cauchy}(0, 2.5) \\
& \text{constraint: } \sigma_e, \sigma_{e'}, \sigma_u, \sigma_w > 0
\end{aligned} \tag{9}$$

The priors for the variance components (the standard deviations σ_e, $\sigma_{e'}$, σ_u, σ_w) and the coefficients representing the means of the Lognormal distributions (β, δ) are mildly informative priors, as in the standard hierarchical model above. These Cauchy priors assume that values of the parameters near 0 are the most likely ones, but extreme values are possible. The Beta(1,1) prior for the mixing probabilities expresses a large prior uncertainty, and express the assumption that the probability is equally likely to be any value between 0 and 1.

Baseline models As baselines, we fit a model corresponding to the retrieval interference account (the standard hierarchical model shown in equation 1 and summarized in Figure 1), and the feature percolation proposal. The latter also assumes a mixture distribution, but only for the condition corresponding to example (2a). Recall that the claim is that in ungrammatical sentences, in some proportion of trials the plural feature on the distractor *cabinets* moves up to the head noun. In (2b), no such mixture process should occur because percolation never occurs; hence a standard hierarchical LogNormal distribution can be assumed here. We therefore defined the following generative process for (2a):

$$\underline{\text{Feature percolation model}}$$

$$\begin{aligned}
y_{ij} \sim & \text{prob\_perc} \cdot \mathit{LogNormal}(\beta + \gamma + u_i + w_j, \sigma_e^2) + \\
& (1 - \text{prob\_perc}) \cdot \mathit{LogNormal}(\beta + u_i + w_j, \sigma_e^2) \\
& \text{where:} \\
& u_i \sim \mathit{Normal}(0, \sigma_u^2), w_k \sim \mathit{Normal}(0, \sigma_w^2), \gamma < 0
\end{aligned} \tag{10}$$

Note that in the specification above the parameter γ, which represents the change in the location parameter, is constrained in the model to be negative; this is because the assumption in the feature percolation proposal is that percolation leads to faster reading time.

For sentences like (2b), in which no percolation is assumed to occur, we simply assumed a LogNormal generative process:

$$y_{ij} \sim \mathit{LogNormal}(\beta + u_i + w_j, \sigma_e^2) \tag{11}$$

Model comparison Having fitted the homogeneous and heterogeneous variance models, as well as the baseline models (the cue-based retrieval and feature percolation models), we need a method for comparing the quality of fit of the mixture models relative to the standard hierarchical models. We use an approximation of the leave-one-out cross-validation (LOO), as discussed in Vehtari, Gelman, and Gabry (2016). We find this approach attractive because it focuses on the predictive performance of the model. LOO compares the expected predictive performance of alternative models by subsetting the data into a training set (for estimating parameters) by excluding one observation. The difference between the predicted and observed held-out value can then be used to quantify model quality by successively holding out each observation. The sum of the expected log pointwise predictive density, $\widehat{elpd}$, can be used as a measure of predictive accuracy, and the difference between the $\widehat{elpd}$'s of competing models can be computed, including the standard deviation of the sampling distribution of the difference in $\widehat{elpd}$. When comparing a model M1 with another model M2, if M2 has a higher $\widehat{elpd}$, then it has a better predictive performance compared to M1. The model comparisons are transitive; if a third model M3 has a higher $\widehat{elpd}$ than M2, then it has a better performance than M1 as well. Vehtari and colleagues have developed an efficient computation of LOO using Pareto-smoothed importance sampling (PSIS-LOO), This is what we use here. For details of PSIS-LOO, see Vehtari et al. (2016).

Results

Table 1 shows model comparisons between the standard hierarchical model, corresponding to the retrieval interference account, and the homogeneous variance model. The table shows that apart from study 1, the homogeneous variance feature overwriting model is clearly superior to the retrieval interference model because it has higher $\widehat{elpd}$ values. Table 1 also shows that the homogeneous variance feature overwriting model furnishes a better fit than the feature percolation model. Finally, the table shows that, except for study 1, the heterogeneous variance model is superior to the homogeneous variance model.

Since the model comparisons are transitive, we can conclude that, among the models compared, the heterogeneous variance feature overwriting model characterises the data best. We therefore focus on the parameter estimates of the heterogeneous variance model below. The estimates from the models for the 10 data-sets are shown in Figure 2. In this model, two noteworthy points are the following: (i) The variance of the high confusability distribution (`sigmap_e`; this corresponds to $\sigma_{e'}$ in the models defined earlier) is relatively large compared to the other variance components; (ii) The difference in probabilities of the two mixture distributions, `diffprob`, is generally greater than 0 across all the studies; however, the uncertainty in the estimate of the probability in study 1 is very high. These two observations suggest that there is more variability in the reading time when the feature overwriting occurs, and that there some evidence that the proportion of trials with feature overwriting is higher in the condition with two singular nouns, consistent with the Nairne proposal.

In summary, overall there is good motivation to assume that in the condition with two singular nouns (example 2b), a proportion of trials comes from a distribution with a larger mean and larger standard deviation, and this proportion is higher than in the condition with one singular and one plural noun (example 2b).

Discussion

We implemented as a statistical model the proposal that nouns with similar feature marking (here, number) may be more confusable due to feature overwriting in some proportion of trials, which in turn leads to occasional increase in difficulty in accessing the correct noun when a dependency is to be completed between the subject and the verb. By fitting Bayesian hierarchical two-mixture models, we showed that 9 out of the 10 data-sets showed evidence for this increased confusability in one condition over the other. The feature overwriting account for the ungrammatical sentences (2a, 2b) appears to be superior to both the retrieval interference and feature percolation accounts.

The three accounts make the same predictions for ungrammatical sentences—a facilitation effect. The modelling presented here allows us to quantitatively compare the relative fit of these proposals for these otherwise indistinguishable accounts. An interesting future direction is to evaluate the predictions of these models for grammatical sentences such as those considered in Franck, Colonna, and Rizzi (2015); Villata and Franck (2016). We plan to address this in future work.

Acknowledgments

Our thanks to Brian Dillon, Sol Lago, Colin Phillips, and Matt Wagers for generously sharing their data. We also benefitted a great deal from discussions with Julie Franck, Whitney Tabor, Aki Vehtari, and Michael Betancourt. For partial support of this research, we thank the Volkswagen Foundation through grant 89 953 to the first author.

References

Dillon, B., Mishler, A., Sloggett, S., & Phillips, C. (2011). A computational cognitive model of syntactic priming. *Journal of Memory and Language*, *69*(4), 85-103.

Franck, J., Colonna, S., & Rizzi, L. (2015). Task-dependency and structure-dependency in number interference effects in sentence comprehension. *Frontiers in psychology*, *6*, 349.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayesian data analysis* (Third ed.). Chapman and Hall/CRC.

Jäger, L. A., Engelmann, F., & Vasishth, S. (2017). Similarity-based interference in sentence comprehension: Literature review and Bayesian meta-analysis. *Journal of Memory and Language*, *94*, 316-339. doi: 10.1016/j.jml.2017.01.004

Lago, S., Shalom, D. E., Sigman, M., Lau, E. F., & Phillips, C. (2015). Agreement attraction in spanish comprehension. *Journal of Memory and Language*, *82*, 133–149.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 1–45.

Nairne, J. S. (1990). A feature model of immediate memory. *Memory & Cognition*, *18*(3), 251–269.

Patson, N. D., & Husband, E. M. (2016). Misinterpretations in agreement and agreement attraction. *The Quarterly Journal of Experimental Psychology*, *69*(5), 950–971.

Stan Development Team. (2016). Stan modeling language users guide and reference manual, version 2.12 [Computer software manual]. Retrieved from `http://mc-stan.org/`

Vehtari, A., Gelman, A., & Gabry, J. (2016). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*.

Villata, S., & Franck, J. (2016). *Similarity-based interference in agreement comprehension and production: Evidence from object agreement.* (Manuscript)

Wagers, M., Lau, E., & Phillips, C. (2009). Agreement attraction in comprehension: Representations and processes. *Journal of Memory and Language*, *61*(2), 206–237.

	(a) Standard HLM vs.		(b) Percolation vs.		(c) Homogeneous variance vs.	
	Homogeneous variance		Homogeneous variance		Heterogeneous variance	
	mixture model		mixture model		mixture model	
Study	elpd_diff	SE	elpd_diff	SE	elpd_diff	SE
1	-0.29	1.67	29.55	6.97	0.57	1.09
2	56.98	13.57	76.34	14.26	15.20	6.07
3	97.62	16.10	112.40	17.43	57.12	11.11
4	71.29	14.08	84.78	14.12	19.66	8.77
5	112.74	18.17	120.45	18.56	63.28	18.12
6	66.84	12.59	85.97	13.88	43.58	12.18
7	72.45	13.76	80.93	14.72	80.92	14.41
8	88.50	14.60	90.22	14.77	40.17	11.87
9	78.35	14.21	108.10	16.04	26.21	7.76
10	90.08	14.14	105.23	15.02	33.59	11.95

Table 1: Comparison of the 10 sets of hierarchical models using PSIS-LOO. Shown are the differences in $\widehat{elpd}$ between (a) the standard hierarchical model and the homogeneous variance mixture model; (b) the feature percolation model and the homogeneous variance mixture model; and (c) the homogeneous vs. heterogeneous variance mixture model. Also shown are standard errors for each comparison. If the difference in $\widehat{elpd}$ is positive, this is evidence in favour of the second model. The pairwise model comparisons are transitive. These comparisons show that the heterogeneous variance mixture model has the best predictive performance.

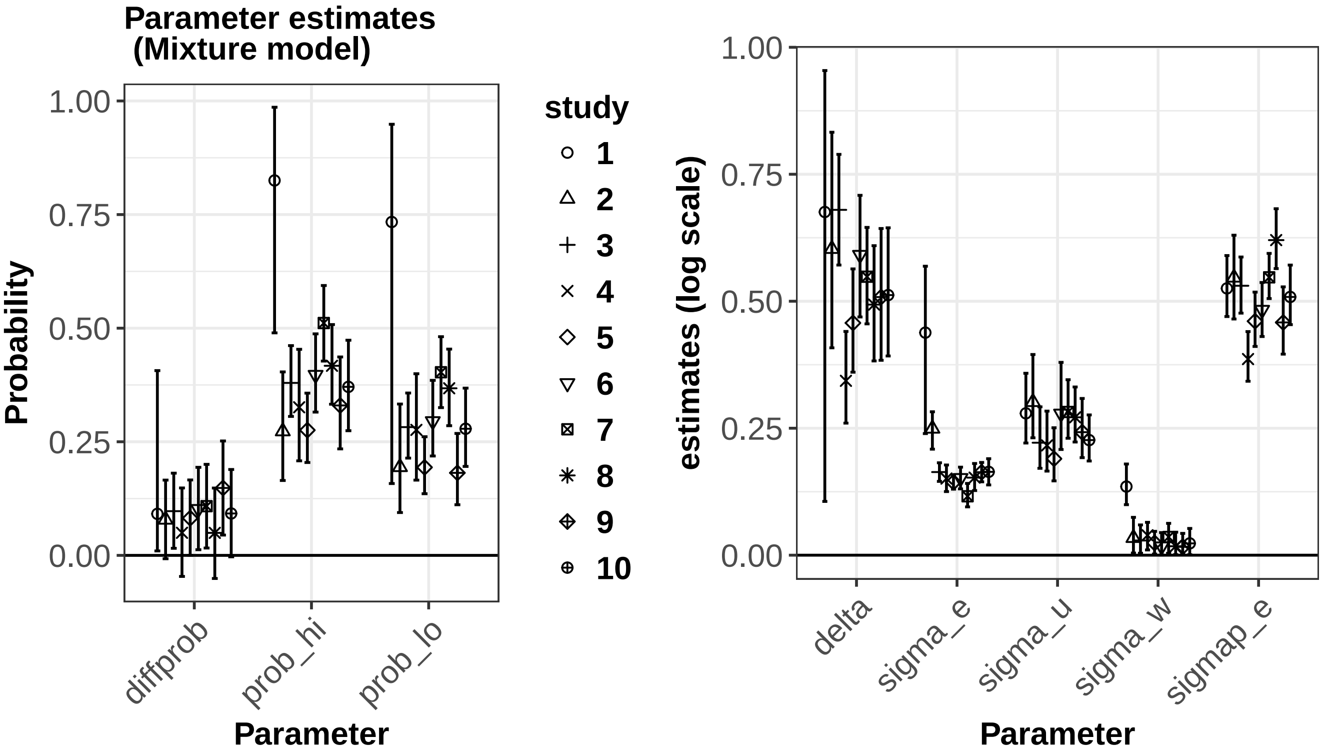


Figure 2: Parameter estimates for the heterogeneous variance hierarchical mixture models.

Implicit Memory Processing in the Formation of a Shared Communication System

Junya Morita (j-morita@inf.shizuoka.ac.jp)[1], **Takeshi Konno (konno-tks@neptune.kanazawa-it.ac.jp)**[2],
Jiro Okuda (jokuda@cc.kyoto-su.ac.jp)[3],**Kazuyuki Samejima (samejima@tamagawa.ac.jp)**[4],
Guanhong li (adam.li@jaist.ac.jp)[5], **Masayuki Fujiwara (m-fujiw@jaist.ac.jp)**[5],
Takashi Hashimoto (hash@jaist.ac.jp)[5]
[1]Faculty of Informatics, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka, Japan
[2]Information and Communication Engineering, Kanazawa Institute of Technology, 7-1 Ohgigaoka, Nonoichi, Ishikawa, Japan
[3]Faculty of Computer Science and Engineering, Kyoto Sangyo University, 458 Koyama, Kamigamo, Kita-Ku, Kyoto, Japan
[4]Graduate School of Brain Science, Tamagawa University, 6-1-1, Tamagawa-Gakuen, Machida, Tokyo, Japan
[5]School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, Japan

Abstract

This paper presents a simulation study focusing on implicit memory in the formation of a new communication system. In the models presented here, two agents aim to achieve their common goal by exchanging messages composed of two figures, whose meanings are not defined in advance. The effect of implicit memory has been studied with two different symbolic processes, implemented in ACT-R. Our results indicate that the difference caused by symbolic processes reduces when implicit memory is incorporated into the model. We have also found the effect of implicit memory on the creation of an isomorphic communication system, shared among agents. These findings suggest that implicit memory has some roles in the formation of a human communication system.

Keywords: Communication; imitation; implicit process; ACT-R

Introduction

People try to communicate with others even when they do not share a common language. They also understand others' intentions through repeated interactions. It has previously been speculated that humans have the ability to develop a new communication system, where only limited common ground is shared, in advance. Addressing the types of cognitive functions involved in such a process will contribute to understand the origins of human communication.

Some researchers have examined this question by designing communication environments in laboratories (for a review Galantucci & Garrod, 2011; Scott-Phillips & Kirby, 2010). For example, Galantucci (2005) conducted an experiment to observe the formation of communication systems, wherein, a pair of participants communicated through a medium that restricted the use of standard communication means, such as utterances and letters. He observed the process of forming a new communication system, and discussed that implicit information was conveyed through routine behavior.

Related studies have also been conducted in the field of language acquisition. Most human infants naturally acquire languages, while a few experience difficulty. From the observations of such a developmental process, some behavioral characteristics that lead to language learning have been found. For example, Tomasello (1999) argued that a type of imitation, called "role-reversal", in which the child aligns himself/herself with the adult speaker, is essential for producing communicative symbols. The cognitive modules behind this behavior have also been discussed. Baron-Cohen (1997) hypothesized the Theory of Mind Module (ToMM) used for imitations of intentional behaviors in others. Rizzolatti and Arbib (1998) also suggests the origins of language from a viewpoint of the mirror neuron system.

For the cognitive modeling community, the challenging questions are: (1) how such modules are computationally represented, and (2) how are these integrated to a cognitive architecture that holds human-level goal management, and memory systems. Concerning these questions, several researchers have constructed a model of language evolution (Reitter & Lebiere, 2011), and an agent including the ToMM, (Stevens, Weerd, Cnossen, & Taatgen, 2016) in the general cognitive architecture.

In our previous study, we also developed a model of sharing communication systems (Morita, Konno, & Hashimoto, 2012). In our model, agents were implemented in the ACT-R cognitive architecture that posses general learning mechanisms such as reinforcement learning, and instance-based learning (Lebiere, Gonzalez, & Martin, 2007). By incorporating imitative learning into these mechanisms, Morita et al. (2012) investigated the role of imitation in the process of forming a new communication system. The results of the study indicated the importance of imitation to simulate the formation process of a human communication system.

However, in our previous work, the production rules executing imitative learning were coded manually. This does not provide the answer as to how these emerge from the human memory system. To overcome this limitation, our present study examines the process that substitutes the manually coded imitation process. This study especially focuses on the role of implicit memory processes in forming a communication system. Before presenting the details of our present study, we recapitulate concepts from our previous study.

Task

This research simulates the experiment reported in Konno, Morita, and Hashimoto (2013), where the authors modified, and used a coordination game taken from Galantucci (2005). As in Galantucci's study, the game environment contained two characters, each controlled by a player, and four intercommunicating rooms. The game was composed of several

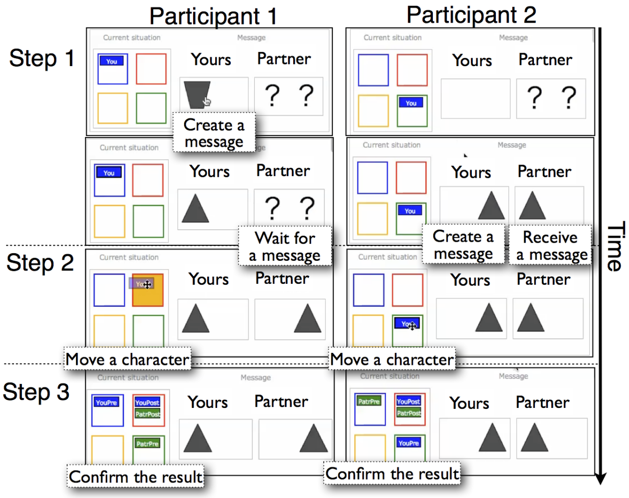


Figure 1: A single round of the coordination game consisted of three steps. In step 1, to create a message, participants selected figures by clicking the segments indicated by “Yours”. In step 2, a character (blue boxes indicated by “You”) was moved by drag-and-drop. In step 3, the result of the movement was shown to participants. Blue boxes (“You-Pre” and “You-Post”) and green boxes (“Pat-Pre” and “Pat-Post”) represent the movements made by the participant, and the partner, respectively.

repeated rounds. At the beginning of each round, characters were randomly placed in two different rooms. Players were unaware of each others’ locations, and aimed to bring their characters to the same room. The characters could not move to rooms that were located diagonally. Owing to this constraint, players needed to communicate before moving their characters.

Figure 1 presents the flow of each round, consisting of three steps: step 1 for exchanging messages; step 2 for moving characters; and step 3 for confirming the result of their movement. Among these steps, step 1 is the most crucial for the success of this task. In this step, the two players construct their own messages, composed of two figures such as “▲□”, using six available figures: ■, ▰, ▼, ◆, ▲, and □. The meanings and usages of the figures were not shared with the participants in advance. Each player could send only one message per round, but they could take turns in exchanging messages. A message sent by the first sender instantly appeared on the screen of the other player. The second sender could compose her/his message after observing the message of her/his partner (see participant 2 in Figure 1). Through such turn-taking, the first sender could transmit her/his current room location, and the second sender could transmit the destination, while taking into account the current room location of her/his partner. Participants were not assigned their roles by the experimenter; instead, they were required to self-assign their roles.

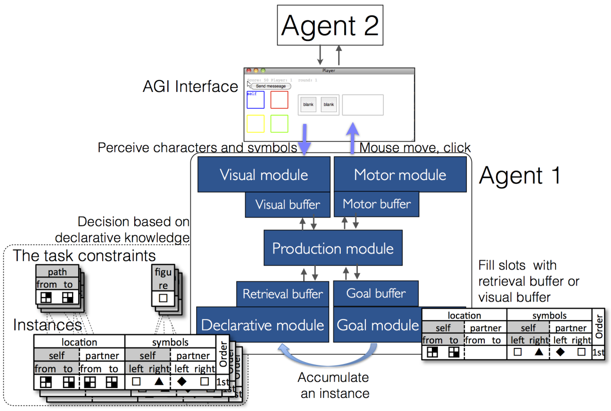


Figure 2: Schema of the model.

In the experiment reported in Konno et al. (2013), participants (21 pairs) attempted to develop a communication system within the stipulated one-hour time limit. When characters moved to the same room, players received two points, otherwise, they lost one point; although, the scores did not drop below zero. The session was terminated when the score reached 50 points. As a result, 66 percent of the participants (14 pairs) successfully reached the points in 48.42 averaged rounds. The models presented in the following section are intended to simulate the behavior of such successful pairs.

Model

Architecture

The task presented in the previous section requires symbolic learning for constructing a new symbol system. In addition, according to Galantucci (2005)’s report, implicit learning, which is not present in symbolic systems, possibly plays a role in this task. Morita et. al (2012) constructed a model using ACT-R (Anderson, 2007), which integrates symbolic and sub-symbolic learning mechanisms. This section illustrates how our previous study constructed a model for sharing the communication system.

ACT-R is composed of several independent modules. The modules used in our study are presented in Figure 2. Except for the production module, each module has a buffer to temporarily store information, called a chunk (a set of slot-value pairs). The production module integrates the other modules using production rules, which consist of a condition-action pair that is used in sequence with other productions to perform a task. The conditions and actions in the production rules are specified, along with the buffer contents of each module.

In the model presented in Morita et al. (2012), two independent agents interact through a simulated task environment developed in the ACT-R graphical user interface (AGI). AGI provides screens that hold visual information as chunks. In this study, the locations of the characters, and messages associated with each agent are displayed on the screen. An agent’s visual module searches for a character and stores its location

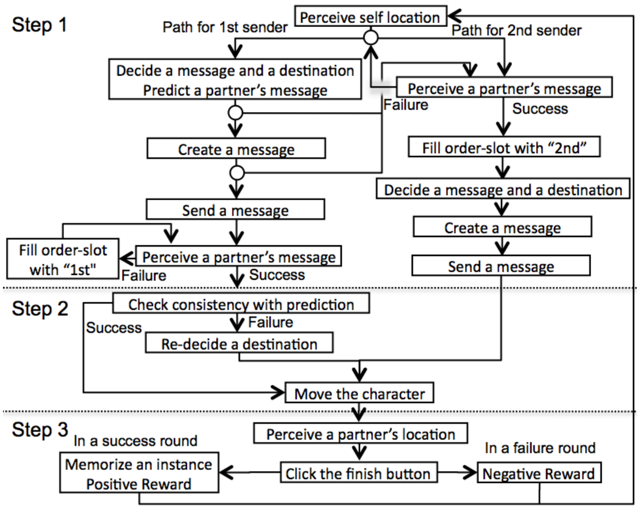


Figure 3: Process of the model. Circles indicate decisions based on conflict resolution.

(room) in a visual buffer. The visual buffer also stores the symbols that compose a message, attending to the screen locations where the figures appear. The simulated task environment also provides a virtual mouse to change the figures and move the characters on the screen.

Visual information stored in the visual buffer is transferred to the goal buffer through the production module. The goal buffer holds the goal of the current task, and other task-related information. Specifically, our model has nine slots for the goal buffer: four slots for storing room locations (initial (from)-destination (to) × self-partner), four slots for storing symbols (left-right × self-partner), and a slot for encoding the order for the exchanged messages.

The declarative module stores past states of the goal buffer, as instances. It also stores chunks representing task constraints such as path information indicating a room that the characters can move to (e.g., *from*_⊞_*to*_⊞_*isa*_*path*), or figures the agent can use to construct a message (e.g., □_*isa*_*figure*). An agent uses these chunks (i.e., declarative knowledge) to choose its destination and construct a message.

The productions of the model construct the process presented in Figure 3. This process is divided into three steps, just as in the original experiment (Figure 1). There are two paths in this process. The left path is for the first sender, and the right path is for the second sender. The choice of path is made by conflict resolution, which is a comparison of two conflicting productions, with noise added utilities. In each phase of the path of the first sender, there is a conflict (indicated by circles) between keeping the path of the first sender, and changing to the path of the second sender. If in any of these the agent selects the path of the second sender, the agent tries to perceive the message of her/his partner from the screen. When the agent obtains the message from her/his partner, s/he realizes that s/he is the second sender (fills the order slot with "2nd"). Otherwise, s/he resolves a conflict by waiting for the message of her/his partner and changing to the path of the first sender. This conflict loop continues until one of the agents sends a message.

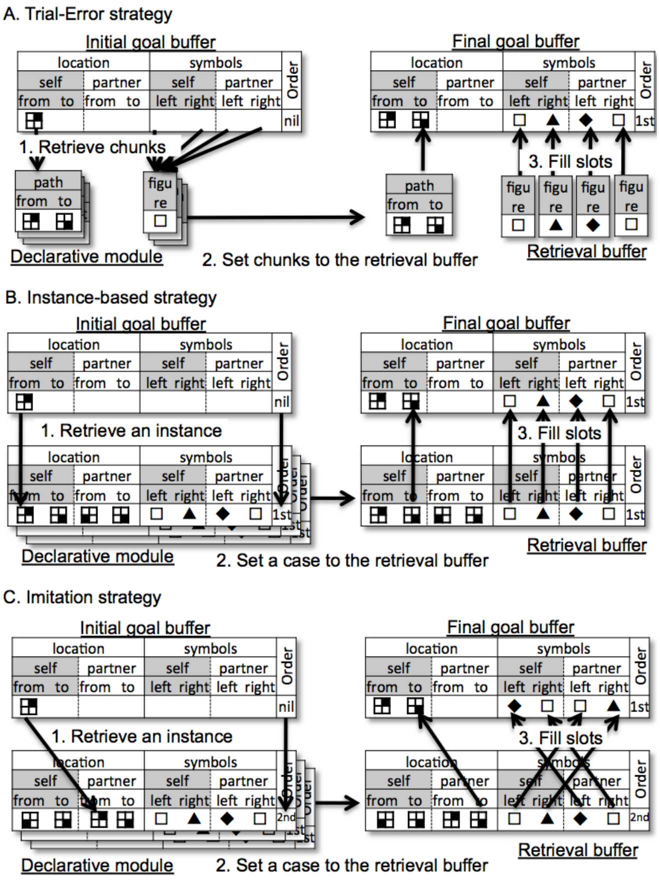


Figure 4: Three types of decision strategies.

Explicit decision process

In step 1, regardless of the contents of the order slot, both agents make decisions about their destinations, and their messages. Concurrently, the first sender predicts the message that s/he will receive from her/his partner. The predicted message is checked against the message received in step 2. When the received message is inconsistent with the predicted message, the agent makes a new decision about her/his destination.

In summary, there are three situations where agents make decisions: the first sender in step 1, the first sender in step 2, and the second sender in step 2. In these situations, agents apply one of the three decision strategies shown in Figure 4. Every decision strategy begins by retrieving chunks from the declarative module, by using the current goal buffer as a cue. In the trial-error strategy, chunks concerning task constraints (chunks representing a path and symbols) are retrieved, and are used to fill in the blank goal slots. In the instance-based strategy, the agent retrieves an instance that is consistent with the current goal buffer. The retrieved instance is used to fill slots concerning the destination, and symbols. The imitation strategy also uses an instance, but the roles of an agent, and her/his partner are reversed when retrieving and filling slots.

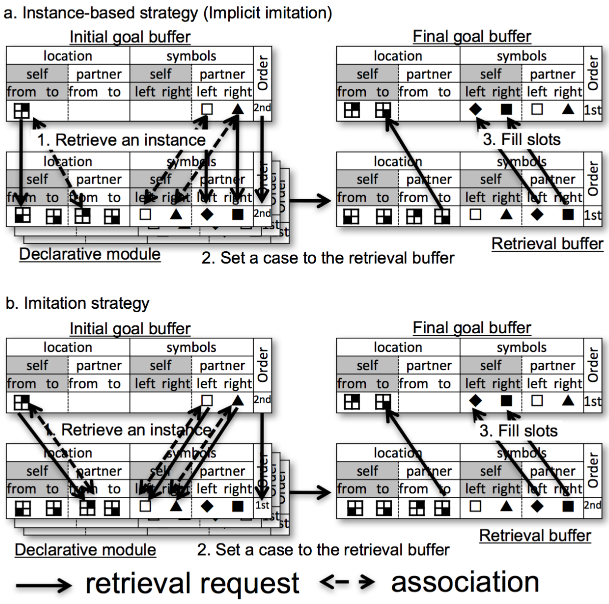


Figure 5: The implicit process of the use of instance.

The implicit decision process

The decision strategies presented above follow a purely symbolic process. Each production rule explicitly holds the mapping of slots from the goal buffer to memorized chunks. Such a process needs different rules, which correspond specifically to each decision situation. Figure 4 only shows an example of the first sender in step 1, where the *location-self-from slot* is used as a retrieval cue. In addition to this slot, a partner's message, (the *symbols-partner-left slot*, and the *symbols-partner-right slot*) can be used as retrieval cues by the second sender in step 1. In the case of the first sender in step 2, where the goal buffer contains the message sent by the agent, more complex retrieval cues are available.

In order to maximize information sent through the exchanged message, it would be better to use instances wherein the slots are either perfectly, or partially matched to the current goal buffer. In the model constructed by Morita et al. (2012), there are rules concerning each combination of buffer slots and matching states for the two decision strategies in all the situations. However, this approach will face difficulty when the model is applied to open communication tasks, where the number of signals, or the number of turns are not decided in advance. Apparently, the model needs to have abstract mechanisms that permit the acquisition of such symbolic processes.

We have not yet solved this problem perfectly. However, in this paper, we propose another process possibly involved in forming a shared communication system, and show the behavior of this for future model development. The proposed mechanism tries to represent unintentional processes in imitation. People sometimes copy others' ideas even when it is not their intention to do so. Those phenomena have been studied in the context of source monitoring error (Johnson, Hashtroudi, & Lindsay, 1993) or deficits in self-other diffrenciations (Baron-Cohen, 1985).

We consider that spreading activation and partial matching are useful to realize such unintentional imitation. These are part of ACT-R sub-symbolic computation, which controls the activation of chunks. The spreading activation represents contextual effects caused by chunks, held by the goal buffer. The same chunks stored in the declarative module receive activation from the goal buffer. The ACT-R memory process usually retrieves chunks having the highest activation within the constraint of the retrieval cues, made by the production rule. When the partial matching process is enabled, it is possible for a chunk that is not a perfect match to the retrieval cues to be the one that is retrieved (Bothell, n.d.).

The combination of the two mechanisms characterizes the process presented in Figure 5, which presents an application of the implicit process to the two strategies by using an example of the second sender in step 1. The solid one-directed arrows connecting the goal buffer with the declarative module indicate the symbolic process noted in the production rule (retrieval cues / fill slots). The dotted two-directed arrows indicate the association connected by the spreading activation.

Importantly, in this figure, the instance-based strategy and the imitation strategy reach the same conclusion. Although the retrieval cues made by the instance-based strategy do not match the instance in the declarative module, the values stored in the slots other than the requested ones accidentally match to the state of the goal buffer. Consequently, this instance receives the high activation, and is retrieved from the declarative module. The retrieved instance is applied with a filling rule used in the imitation strategy.

The benefit of such an implicit imitative process involves reducing the complexity of symbolic processes. However, it is unknown whether the ACT-R sub-symbolic computation actually generates such imitative effects. To explore the role in the formation of human communication systems, it is needed to examine the behavior of this mechanism in a controlled simulation experiment.

Simulation

Simulation conditions

We first set up the following two models controlling the decision strategies presented in Figure 4.

- Instance model: In this model, the agent first tries the instance-based strategy. If the instance-based strategy fails, the agent chooses her/his destination and message based on the trial-error strategy.
- Imitation model: This model extends the instance model by adding the imitation strategy. The agent first tries to choose her/his destination and message using the instance-based strategy. If the agent fails to retrieve an instance, the imitation strategy is applied. When all other decision strategies fail, the agent uses the trial-error strategy.

In our previous study, the imitation model indicated better performance and better fitting to the human data. The imitation strategy gives the model the benefit of using instances in

Table 1: The performance indices. The numbers in parentheses indicate standard deviation.

	Data	ExIns	ExImi	ImpIns	ImpImi
Success rates	0.66	1.00	1.00	0.97	0.98
Round	48.42	70.74	60.21	115.48	117.33
	(13.36)	(17.63)	(13.14)	(35.51)	(37.13)

different ways. Therefore, the success rates of the imitation model are higher than the instance model in the early round.

With respect to the implicit process, we also set up the following two conditions.

- Explicit process: This model does not have the spreading activation, and partial matching mechanisms. As sub-symbolic parameters, only the activation noises, and the expected gains are set ($blc = 2, ans = 0.5, egs = 1$) to make the behaviors of the two agents differ. Except for this parameter setting, this model is same with the model presented in Morita et al. (2012)
- Implicit process: This model includes the implicit process presented in Figure 5. In addition to the sub-symbolic parameters noted in the explicit process, the matching penalty, and the maximized associative strength are set ($mp = 2$, $mas = 10$, $blc = 2$). This model also has several supplemental production rules to deal with memory errors, caused by partial matching.

Combining the symbolic, and sub-symbolic conditions, we prepared four models: ExpImi (the imitation model with the explicit process), ImpImi (the imitation model with the implicit process), ExpIns (the instance model with the explicit process), and ImpIns (the instance model with the implicit process). By comparing these, we try to identify the role of implicit processes in forming a shared communication system.

In this simulation, each model runs 100 times. In each run, the model continues the trial session for 3,600 sec[1], or until the scores reach 50 points. Following the trial session, the model is engaged in three test sessions similar to the experiment presented in section 2.

Results

Performance Table 1 shows the proportion of runs/pairs whose scores reached 50 points, which is a termination condition for the session. It also presents the numbers of rounds required to reach the termination condition. Some runs utilizing the implicit model failed to form a communication system; whereas, all runs utilizing the explicit model succeeded in completing the session. Even though there were pairs that did not reach the termination condition, the number of rounds required to complete the session in the experiment (data) was smaller than that in all other models. Compared to the implicit models, the explicit models finished the session in fewer rounds. The effect of the decision strategy is only observed in the explicit models, where the imitation model finished faster than when using the instance-only strategy.

Table 2: Fitting of the model performance to the human data.

	ExIns	ExImi	ImpIns	ImpImi
RMSE	0.11	0.10	0.20	0.20
R^2	0.71	0.78	0.74	0.74

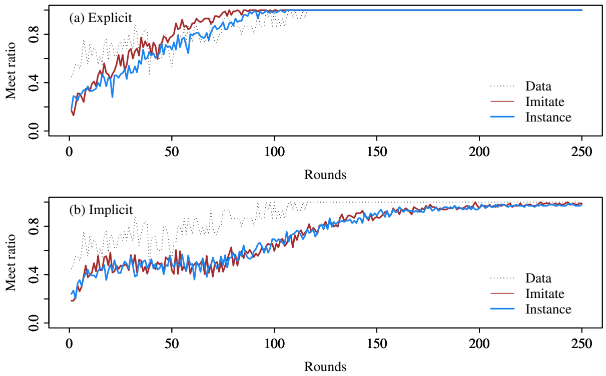


Figure 6: The ratio of sucess at each round.

The detailed processes are presented in Figure 6, which indicates the proportion of runs/pairs who met in the same room for each round. Table 2 also shows fitting indices calculated from the figure. Although the explicit models have a smaller absolute distance to the human data (RMSE) than the implicit models, there are no remarkable differences of an overall trend (R^2) between the four models.

Messages People usually try to share the same communication system even when their first languages are different. To model such characteristics of human communication, we examine the similarity of the constructed message system, as indicated by the following index.

$$Sim = \vec{M}_{player1} \cdot \vec{M}_{player2} \quad (1)$$

where $\vec{M}$ indicates a vector whose element corresponds to the use frequency of the 36 combination of figures. A dot product of the two vectors represents the degree of symbol sharing among agents.

Figure 7 indicates the moving scores of similarity with the window size of 20 rounds. Table 3 summarizes the fitting to human data, which is calculated from Figure 7. Among the four models, ExImi shows the best fit to human Data, consistent with the finding in Morita et al. (2012). It is noteworthy that models with the implicit process replicate the temporal trend of the similarity score, even without the explicit imitation strategy. The difference between the instance and the imitation models is also quite small in the implicit process.

Table 3: Fitting of the similarity score.

	ExIns	ExImi	ImpIns	ImpImi
RMSE	0.38	0.16	0.29	0.29
R^2	0.06	0.72	0.57	0.64

[1] We used the simulation time estimated by ACT-R.

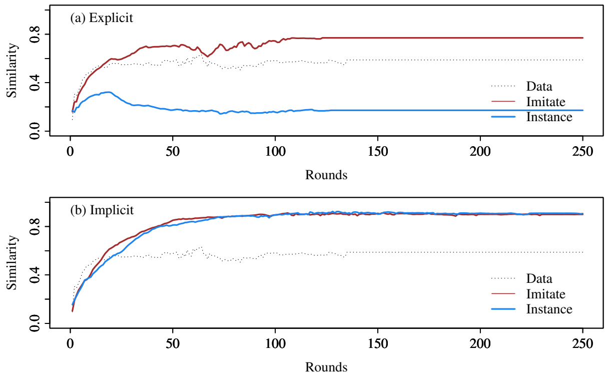


Figure 7: Similarity of messages at each round

Discussion and Conclusions

This study constructed a model that forms a new communication system through interactive coordination. To date, many models for language evolution have been developed (for a review Steels, 2011). In addition, there exists a research that uses ACT-R to simulate experiments of forming a communication system (Reitter & Lebiere, 2011). However, such studies have not dealt with a situation with spontaneous turn-taking, or role-setting operations. Most of the previous models assign roles to agents, including being a director, or matcher, using simulation parameters.

Setting such an interactive situation, this paper examined the effect of implicit processes in forming a shared communication system. The results indicate a clear influence of the process on both the performance, and the similarity of messages. Importantly, adding the implicit process into the model, the difference caused by the explicit process almost disappeared. Although these findings alone are not enough to draw a concrete conclusion, this study shows that an isomorphic symbol system can be made without hand-coded imitations.

However, compared to human data, the implicit process results in a slower forming process, as presented in Table 1. Several explanations can be considered for this difference. The first explanation is about heuristics, utilized by human participants. Some participants in the experiment used ▲to indicate the upper-rooms based on the shape similarity to the upper arrows. If such a pre-existing common ground is used in the model, the performance will undoubtedly increase. The other possible explanation relates to individual differences. As suggested by the failure pairs in the experiment, there are large variations in the formation process of the symbol communication system, in the collected human data. The literatures in the field of developmental psychology also indicate that children on the autism spectrum exhibit a unique language acquisition process (Baron-Cohen, 1985, 1997). Considering these factors, we can hypothesize that cognitive functions involved in forming a communication system are not determined uniquely, and the variations of the ACT-R model presented here might represent such individual differences. To examine this hypothesis, our future study will analyze the detailed behavior characteristics involved in this task. Especially, we will improve the similarity score used in this study to include characteristics of the syntax (combination rules of symbols), and symbol-meanings mappings.

Acknowledgment

This work was supported by JSPS KAKENHI 26240037, and 16K16113.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Baron-Cohen, S. (1985). *Social Cognition and Pretend Play in Autism.* Doctoral thesis, University of London.

Baron-Cohen, S. (1997). *Mindblindness: An Essay on Autism and Theory of Mind.* The MIT Press.

Bothell, D. (n.d.). Act-r 6.0 reference manual working draft [Computer software manual].

Galantucci, B. (2005). An experimental study of the emergence of human communication systems. *Cognitive Science*, *29*(5), 737–767.

Galantucci, B., & Garrod, S. (2011). Experimental Semiotics: A Review. *Frontiers in Human Neuroscience*.

Johnson, M. K., Hashtroudi, S., & Lindsay, D. (1993). Source Monitoring. *Psychological Bulletin*, *114*(1), 3–28.

Konno, T., Morita, J., & Hashimoto, T. (2013). Symbol communication systems integrate implicit information in coordination tasks. In *Advances in cognitive neurodynamics (iii).* Springer.

Lebiere, C., Gonzalez, C., & Martin, M. (2007). Instance-based decision making model of repeated binary choice. *Proceedings of the 8th International Conference on Cognitive Modeling*.

Morita, J., Konno, T., & Hashimoto, T. (2012). The Role of Imitation in Generating a Shared Communication System. *Proceedings of the 34th Annual Conference of the Cognitive Science Society*.

Reitter, D., & Lebiere, C. (2011). How groups develop a specialized domain vocabulary: A cognitive multi-agent model. *Cognitive Systems Research*, *12*, 175–185.

Rizzolatti, G., & Arbib, M. A. (1998). Language within our grasp. *Trends in Neuroscience*, *21*, 188–194.

Scott-Phillips, T. C., & Kirby, S. (2010, January). Language evolution in the laboratory. *Trends in Cognitive Sciences*, *14*(9), 411–417.

Steels, L. (2011). Modeling the cultural evolution of language. *Physics of Life Reviews*, *8*(4), 339–356.

Stevens, C., Weerd, H. d., Cnossen, F., & Taatgen, N. (2016). A metacognitive agent for training negotiation skills. In *Proceedings of the 14th International Conference on Cognitive Modeling* (pp. 27–32).

Tomasello, M. (1999). *The cultural origins of human cognition.* Harvard University Press.

How does rumination impact cognition? A first mechanistic model.

Marieke K. van Vugt (m.k.van.vugt@rug.nl) & Maarten van der Velde
Department of Artificial intelligence, Nijenborgh 9
9747 AG, Groningen, The Netherlands

ESM-MERGE Investigators[*]
Maastricht University, The Netherlands.

Abstract

Rumination is a process of uncontrolled, narrowly-foused negative thinking that is often self-referential, and that is a hallmark of depression. Despite its importance, little is known about its cognitive mechanisms. Rumination can be thought of as a specific, constrained form of mind-wandering. Here, we introduce a cognitive model of rumination that we developed on the basis of our existing model of mind-wandering. The rumination model implements the hypothesis that rumination is caused by maladaptive habits of thought. These habits of thought are modelled by adjusting the number of memory chunks and their associative structure, which changes the sequence of memories that are retrieved during mind-wandering, such that during rumination the same set of negative memories is retrieved repeatedly. The implementation of habits of thought was guided by empirical data from an experience sampling study in healthy and depressed participants. On the basis of this empirically-derived memory structure, our model naturally predicts the declines in cognitive task performance that are typically observed in depressed patients. This study demonstrates how we can use cognitive models to better understand the cognitive mechanisms underlying rumination and depression.

Keywords: mind-wandering; rumination; associative memory; depression; sustained attention

Introduction

Rumination is the process of narrowly-focused uncontrolled repetitive negative thinking—mostly self-referential—that lies at the core of depression (Marchetti, Koster, Klinger, & Alloy, 2016; Nolen-Hoeksema & Morrow, 1991; Treynor, Gonzalez, & Nolen-Hoeksema, 2003). Despite the serious clinical consequences of this process, there is to date no coherent computational cognitive theory that describes it. While there are several verbal theories (Marchetti et al., 2016), those can only explain their own limited set of experiments and do not make quantitative predictions.

To develop a theory of rumination, we built on recent research and modeling of mind-wandering, because rumination can be thought of as a highly constrained form of mind-wandering (Christoff, Irving, Fox, Spreng, & Andrews-Hanna, 2016). Mind-wandering is a process of task-unrelated thinking that takes up approximately 50% of our time (Killingsworth & Gilbert, 2010; Smallwood & Schooler, 2015), and can sometimes help and sometimes hinder performance. For example, in very undemanding contexts, mind-wandering can serve useful functions for creativity (Baird et al., 2012) and planning (Baird, Smallwood, & Schooler, 2011). On the other hand, it disrupts performance when it takes away cognitive resources that are needed to perform the task, and this occurs in particular when mind-wandering is unintentional and uncontrolled (Seli, Risko, Smilek, & Schacter, 2016), as is the case with rumination. This could explain why people that suffer from rumination typically also report having difficulties concentrating.

So far, the theories of rumination can be broadly divided into three classes. One class of theories suggests that rumination arises from an increased bias towards negatively-valenced information (Dalgleish & Watts, 1990). When attention is focused more on negative information, this reduces ability to focus on other things (Whitmer & Gotlib, 2013). Another class of theories instead focuses on inhibition, and suggests that the primary deficit underlying rumination is an inability to disengage from information, in particular when this information is negative and self-focused (Whitmer & Banich, 2007). The third theory of rumination—which we refer to as "habits of thought"—focuses not on control processes such as attention and inhibition, but rather on the content of thoughts during mind-wandering. Patterns of memory associations that are frequently rehearsed can become something like an attractor (Cramer et al., 2016), and therefore will be replayed any moment there is time for mind-wandering. To start to distinguish between these different theories of rumination, it is helpful to specify them in more detail by implementing them in a cognitive architecture, and to simulate their predictions for performance on a simple sustained attention task. Here we will start by implementing the habits of thought theory, which is of interest because it exploits the fact that the ACT-R cognitive architecture is in essence a memory theory.

To implement our theory of rumination, we will make use of our own computational model of mind-wandering (Taatgen, van Vugt, Daamen, Katidioti, & Borst, submitted; van Vugt, Taatgen, Bastian, & Sackur, 2015). This model frames mind-wandering in terms of resource competition, in which task goals compete with mind-wandering goals, and mind-wandering occurs when that goal wins the competition. Mind-wandering is modelled as a process of memory retrieval. Consequently, the mind-wandering model is uniquely suited for implementing the third theory of rumination, which says that rumination is driven by the existence of thought habits that are maladaptive. We hypothesize that these thought patterns are what causes people to get caught

[*] ESM-MERGE Investigators (alphabetical order): D. Collip, Ph. Delespaul, , N. Geschwind, M. Janssens, M. Lardinois, J. Lataster, T. Lataster, C. Menne-Lothmann, I. Myin-Germeys, M. van Nierop, M. Oorschot, C. Simons, J. van Os, M. Wichers.

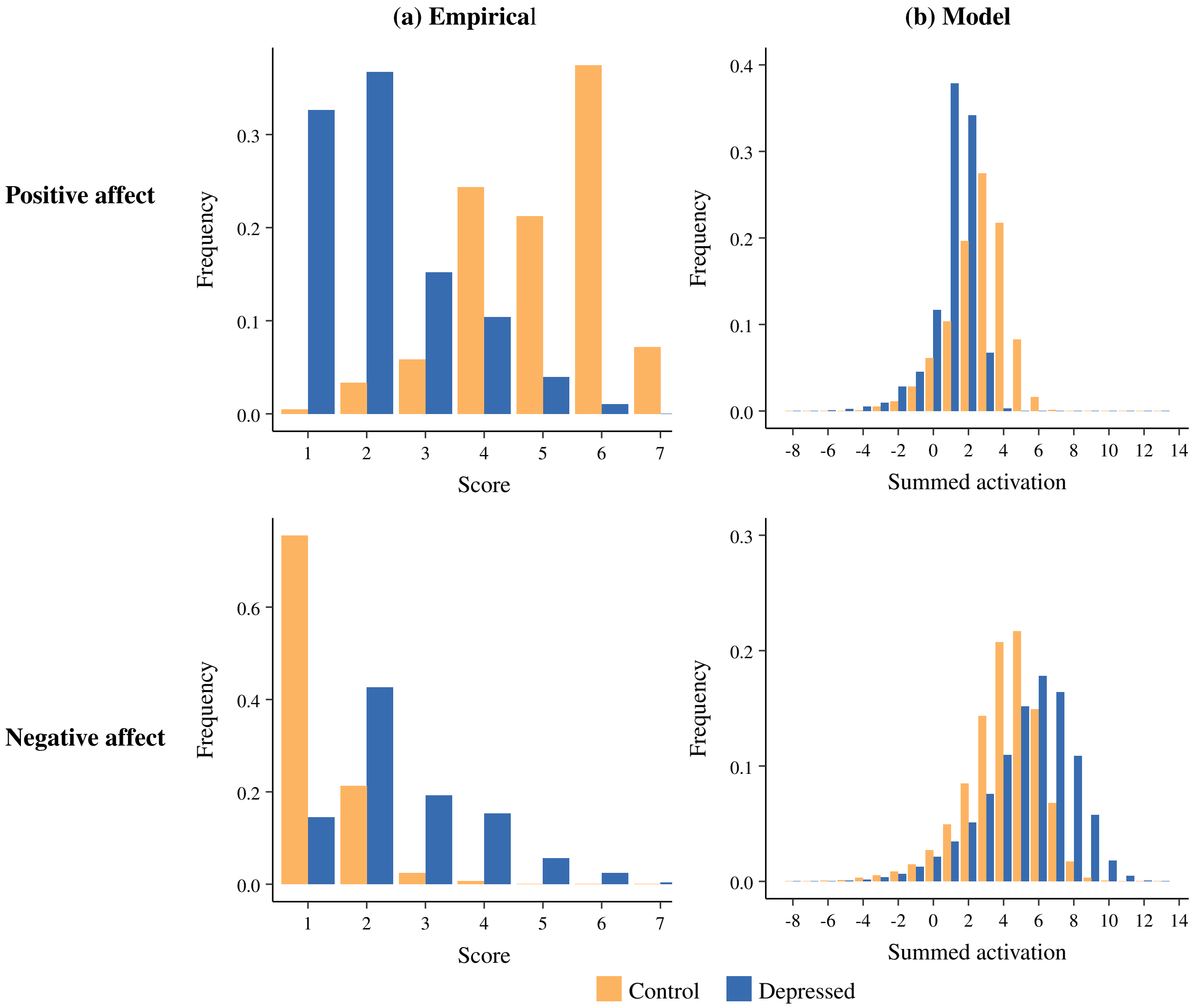


Figure 1: Reported positive and negative affect. (a) shows the frequency with which participants reported experiencing particular degrees of positive and negative affect in the experience sampling data, while (b) shows the summed activation of positive and negative chunks produced by the model (our closest proxy for the continuous affect ratings in the empirical data).

in a funnel of repetitive negative thinking, and disconnect from the current task, which leads to the perceived problems in concentration. This predicts that a model of rumination with exactly the same production rules but a different memory chunk structure should perform worse on a sustained attention task than a "healthy model." Later studies should implement the other two theories of rumination, and examine how their predictions may differ.

Methods

Mind-wandering model

We implemented our mind-wandering model (which forms the basis for the rumination model) in the adaptive control of thought-rational (ACT-R) architecture (Anderson, 2007; Anderson, Fincham, Qin, & Stocco, 2008). The model rests on two basic assumptions: firstly, there is a continuous competition between a mind-wandering and a task process, and consequently, mind-wandering is likely to kick in when there is a spare moment in the task, and secondly, mind-wandering is primarily a process of memory retrieval (van Vugt et al., 2015; Taatgen et al., submitted); implemented as retrieving chunks from declarative memory. As is usual in ACT-R's memory retrieval, the most active chunk is the one that will be retrieved. Each chunk's activation is determined by three factors: the amount of recent use (more recent and more frequent use imply a larger chunk activation), the spreading activation from other chunks, and random activation noise. Since each chunk has a slot containing its emotional valence, the spreading activation ensures that chunks with the same emotional valence are more likely to follow each other than chunks with different emotional valence, in line with previous empirical results (van Vugt, Shahar, & Britton, 2012). The mind-wandering memory retrieval process continues until a memory chunk that is retrieved reminds the model of its main task.

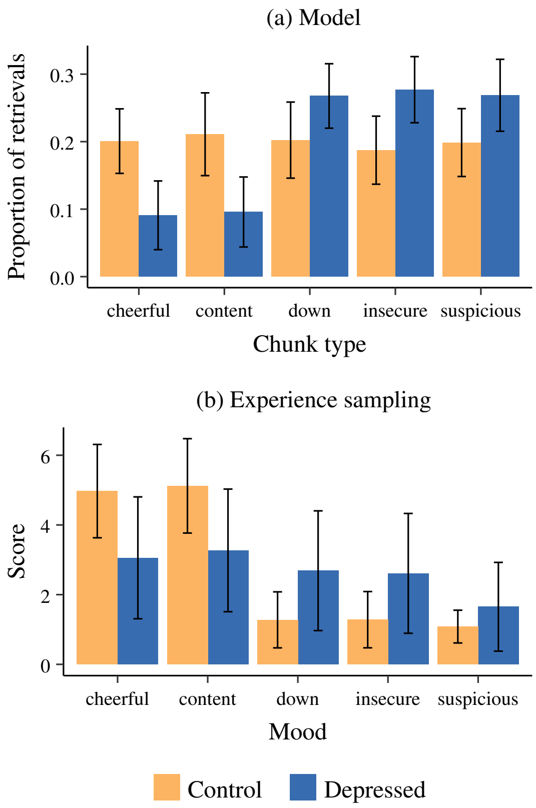


Figure 2: The probability for memories being retrieved depends on their emotional valence. (a) The control model retrieves every mood more or less equally often, while the depressed model preferentially retrieves negatively-valenced items. (b) Empirical data shows that depressed participants experienced each mood with comparable intensity, while non-depressed controls displayed a bias towards positive moods. Error bars reflect standard error of the mean.

At that point, the main goal switches from mind-wandering to being on-task. During the period of mind-wandering, the retrieval module is busy retrieving memories, which means that responses to incoming stimuli will be done in automatic mode by giving the default response, and will not involve memory retrievals. In addition, since ongoing memory retrievals (which occur during mind-wandering) first have to be finished before a response is made, the mind-wandering process results in an increase in the variability of response times during mind-wandering, in line with behavioral findings (Bastian & Sackur, 2013; van Vugt & Broers, 2016).

The mind-wandering model was given a sustained attention to response task—SART (Cheyne, Carriere, & Smilek, 2009; Smallwood et al., 2004) to make testable predictions for behavioral. In this task, participants see a stream of digits, presented at a pace of one per three seconds, and they press a button whenever a digit is presented, except when it is the number three. The number three, the nogo stimulus, is presented on roughly 10% of the trials. This means that when participants do not pay attention, they will revert to an automatic mode of responding, and fail to inhibit responses to the rare nogo stimuli.

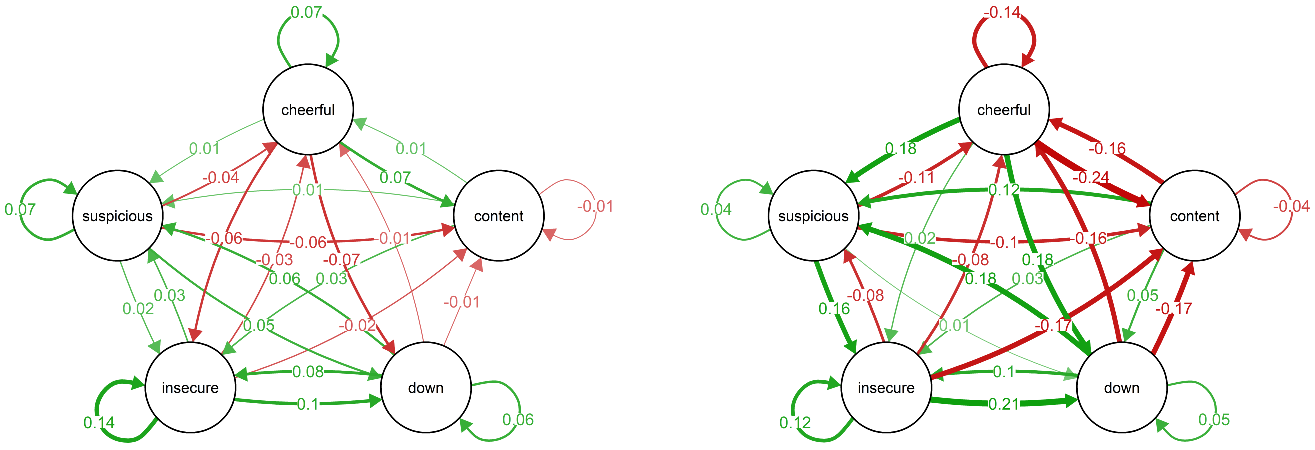
Adaptations for modeling rumination

Our rumination model implemented the "habits of thought" theory of rumination. The main idea underlying this theory is that rumination consists of retrieval of a set of well-rehearsed thought patterns that are predominantly negative and self-referential. We tried out different methods for generating strong loops of self-referential negative thinking, and found that the most effective way was to increase the number of chunks with negative valence, such that these negative-valence chunks are more likely to be retrieved. This increase in the number of negative-valence chunks also increases the amount of spreading activation between them. Specifically, the non-depressed model has 55 chunks in total, 11 per mood (cheerful, content, down, insecure, suspicious—these moods were derived from the empirical data described below). The depressed model also has 55 chunks, but those consist of 5 chunks of each of the positive moods (cheerful and content), and 15 chunks of each of the negative moods (down, insecure and suspicious). For both models, the association strengths (S_{ji}'s) were 0.1 between moods of the same valence, and 0.01 between moods of different valence. These association strengths were chosen such that the spreading activations were roughly balanced with the base level activations, and slightly adjusted to better fit the empirical data. Our rumination models differ from our previous mind-wandering model in that there are two chunks that remind the user of the main task—one with positive and one with negative valence—instead of just one with a positive valence as was the case in the previous model.

To assess model performance, we simulated data for 100 participants suffering from rumination, and 100 participants with the usual model structure (i.e., without rumination). We chose for 100 participants because this is in the same ballpark as the empirical data. We then measured how many chunks of each mood the model recalled during mind-wandering episodes, together with their transition probabilities. These measures were compared to the experience sampling data described below to adjust the model. Once the models' memory structures were adjusted to exhibit thought contents similar to what was observed in the experience sampling data, we looked at the model's task performance, and examined whether rumination impaired performance on a simple go/nogo task (as would be expected).

Experience sampling data on depression

We configured the set of memory chunks and their associative structure on the basis of an experience sampling study (Wigman et al., 2015). In such a study, participants are prompted several times a day to respond to a brief questionnaire about their thoughts and experience. This study found that depressed patients had an increase in the number of negative-valence thoughts, more difficulty concentrating, and most importantly, a network of negative thoughts (specif-



(a)

(b)

Figure 3: Transitions between different moods. (a) Difference between control and depressed networks in empirical data from Wigman et al. (2015) on the basis of regression coefficients. (b) Modeled network difference between depressed and control participants on the basis of transition probabilities. Green: control > depressed. Red: depressed > control.

ically, suspicious, down, and insecure) that was much more separate from the network of positive thoughts (content and cheerful) than in the control subjects. The experience sampling data we used in this study was collapsed across all participants in the depression and control groups. It contained data from 129 depressed patients and 212 non-depressed controls, who were sampled ten times per day for a period of 5-6 days.

Results

Average thought frequencies

Rumination is associated with increased negative memory and a prevalence of negatively valenced thought. To examine whether our model could reproduce those findings, we first compared the activation of positive-valence and negative-valence chunks, as well as the frequency of retrieval of the different subcategories. A challenge in this comparison is that the empirical data consists of the average rating of positive and negative emotions on a 7-point Likert scale, which has no direct correlate in the model. Since the judgment is supposed to reflect a participant's general mood, we used the summed activation of all positive/negative chunks as a proxy for positive and negative affect, respectively.

We were able to reproduce an increase in the summed memory activation of negative chunks, and a decrease in the summed memory activation of positive chunks (Figure 1(b)). We then examined how frequently positive and negative memory chunks were retrieved by healthy and depressed models. Figure 2(a) shows that while the healthy model retrieves positive and negative valence equally frequently, the depressed model tends to retrieve negative chunks more frequently (which then leads to a feedback loop, because these negative chunks then become more active, which makes it likely that they will be retrieved even more often). The empirical data (Figure 2(b)) are somewhat similar, although here it appears as if healthy participants relatively suppress negative memory chunks. Note that this is at odds with a substantial body of literature that reports a negativity bias for depressed patients (Whitmer & Gotlib, 2013) instead of a positive facilitation in healthy controls (but see Levens and Gotlib (2010)).

Transitions between moods

A unique feature of the data presented in Wigman et al. (2015) was that not just frequencies of different types of thought were presented, but also the network of the transitions between different moods. In the empirical work by Wigman et al, these transitions were measured by fitting a multilevel linear mixed effect model to the data. Each score at time $t-1$ was used to predict the score at time point t, and this resulted in a fixed-effect coefficient for each connection between moods. The difference in magnitude of these coefficients between depressed and control participants is shown in Figure 3(a). The largest difference between healthy and depressed participants that our model needs to capture is an increase in the number of transitions between negative-valence chunks for the depressed patients, together with a decrease in the number of transitions between positive and negative valence chunks. As before, we cannot produce exactly the same measure in our model, which retrieves one memory chunk at a time. The closest approximation to the regression coefficients in the empirical data are transition probabilities between retrieved memory chunks with different moods. Figure 3(b) shows that when we measure the transitions for the depressed and control networks, we reproduce the somewhat stronger between-negative connectivity and the somewhat weaker positive-to-negative connectivity for the depressed model. Nevertheless, the modelled effects are not as strong as in the empirical data.

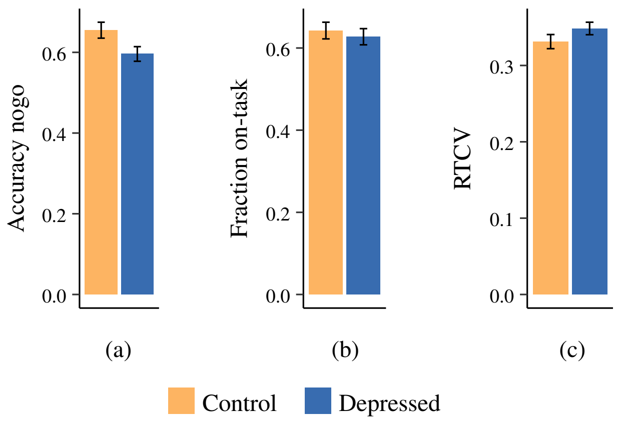


Figure 4: Comparison of performance of the control (orange) and rumination (blue) model on a sustained attention to response (go/nogo) task. The depressed model shows lower accuracy (a) but no difference in the fraction of mind-wandering (b),or coefficient of variation of response time to correct responses. Error bars reflect standard error of the mean.

Novel predictions: task performance

After having developed a rumination model by adapting the memory structure (i.e., thought patterns) on which it operates, we can examine how it performs on a cognitive task. In the data reported by Wigman et al. (2015), depressed participants reported having significantly more difficulty in concentrating than healthy controls ($t(4098.8)$=-44.1, $p < 2.2 * 10^{-16}$)[†]. Consequently, we predicted that the rumination model would exhibit an impairment on a sustained attention task that is typically used to measure mind-wandering, and that it would be distracted more frequently. Figure 4(a) shows that performance on a sustained attention to response task was worse for the depressed relative to the control model ($t(196.5)$=2.2, $p = 0.03$). A potential reason for this decline in performance is an increase in the amount of off-task thinking (Figure 4(b), although this change in off-task thinking was not statistically significant, $t(197.8)$=0.53, $p = 0.60$). There is also no significant difference in the coefficient of variation of response time (Figure 4(c); $t(195.1)$=1.39, $p = 0.17$), which is considered to be a sensitive index of off-task thinking.

Discussion

In summary, we have developed a novel approach to modeling psychopathology by means of cognitive architectures. We structured the model's memory on the basis of experience sampling data. We then used our existing mind-wandering model to make predictions for how performance on a sustained attention task would be impacted by rumination. We found that merely by modifying the structure and contents of the model's memory, we were able to produce retrieval frequencies and sequences similar to what was observed in the experience sampling study. In addition, our model predicted impairments on a sustained attention task, in line with subjective reports of participants about difficulty with concentration.

While the model's performance was qualitatively in line with the observations from Wigman et al. (2015), we were not able to fit the exact patterns. This failure to fit may point at a structural limitation of our individual model, or of the general ACT-R cognitive architecture. It turned out to be very difficult to "create" cycles of rumination because ACT-R only adapts chunk activation, and not the associations between chunks, which may be the true habits of thought.

Another potential reason for this failure is our highly simplified representation of moods. Previous studies have represented mood in terms of physiology (Dancy, 2013) or in terms of expectations and desirability of the state of the world (Marsella & Gratch, 2009).

Our study makes an important contribution to the nascent field of computational psychiatry (Adams, Huys, & Roiser, 2016). So far, computational psychiatry involved mostly simple reinforcement learning models of psychiatric problems (but see Kottlors, Brand, and Ragni (2012)), while we demonstrated the utility of cognitive architectures. The advantage of using cognitive architectures compared to simpler theories, is that it is possible to simulate performance on many different tasks. Moreover, it becomes possible to examine changes in cognitive strategies (the "software of cognition") in the same context as changes in mental habits (the "hardware of cognition"), as we have demonstrated in this paper.

In summary, we have demonstrated how we can implement a cognitive theory of rumination, and make testable predictions about performance on a mind-wandering task. This leads to new avenues in better understanding what the exact mechanisms are that underlie rumination, and depression in general.

Acknowledgments

This research was supported in part by a Mind & Life visiting scholarship to MvV. Elske Bos and Evelien Snippe helped guide us to the relevant experience sampling data.

References

Adams, R. A., Huys, Q. J. M., & Roiser, J. P. (2016). Computational Psychiatry: towards a mathematically informed understanding of mental illness. *Journal of Neurology, Neurosurgery & Psychiatry*, *87*(1), 53–63.

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 in Cognitive Sciences*, *12*(4), 136–143.

Baird, B., Smallwood, J., Mrazek, M. D., Kam, J. W. Y., Frank, M. J., & Schooler, J. W. (2012). Inspired by distraction. mind wandering facilitates creative incubation. *Psychological Science*, *23*(10), 1117-1122.

Baird, B., Smallwood, J., & Schooler, J. W. (2011). Back to the future: Autobiographical planning and the functionality

[†] T-tests used Welch's correction for degrees of freedom

of mind-wandering. *Consciousness and Cognition*, *20*(4), 1604–1611.

Bastian, M., & Sackur, J. (2013). Mind wandering at the fingertips: automatic parsing of subjective states based on response time variability. *Frontiers in Psychology*, *4*, 573.

Cheyne, D., Carriere, J. S. A., & Smilek, D. (2009). Absent minds and absent agents: Attention-lapse induced alienation of agency. *Consciousness and Cognition*, *18*, 481-493.

Christoff, K., Irving, Z. C., Fox, K. C. R., Spreng, R. N., & Andrews-Hanna, J. R. (2016). Mind-wandering as spontaneous thought: a dynamic framework. *Nature Reviews Neuroscience*, *17*(11), 718–731.

Cramer, A. O. J., van Borkulo, C. D., Giltay, E. J., van der Maas, H. L. J., Kendler, K. S., Scheffer, M., & Borsboom, D. (2016). Major Depression as a Complex Dynamic System. *PLOS ONE*, *11*(12), e0167490.

Dalgleish, T., & Watts, F. N. (1990). Biases of attention and memory in disorders of anxiety and depression. *Clinical Psychology Review*, *10*(5), 589–604.

Dancy, C. (2013). ACT-Rψ: A Cognitive Architecture with Physiology and Affect. *Biologically Inspired Cognitive Architectures*, *6*(1), 40-45.

Killingsworth, M. A., & Gilbert, D. T. (2010). A wandering mind is an unhappy mind. *Science*, *330*(6006), 932.

Kottlors, J., Brand, D., & Ragni, M. (2012). Modeling Behavior of Attention-Deficit-Disorder Patients in a N-Back Task. In *Proceedings of the 11th International Conference on Cognitive Modeling.* Berlin: Universitaetsverlag der TU Berlin.

Levens, S. M., & Gotlib, I. H. (2010). Updating positive and negative stimuli in working memory in depression. *Journal of Experimental Psychology: General*, *139*(4), 654.

Marchetti, I., Koster, E. H., Klinger, E., & Alloy, L. B. (2016). Spontaneous thought and vulnerability to mood disorders the dark side of the wandering mind. *Clinical Psychological Science*, *4*(5), 835-857.

Marsella, S. C., & Gratch, J. (2009). EMA: A process model of appraisal dynamics. *Cognitive Systems Research*, *10*, 70-90.

Nolen-Hoeksema, S., & Morrow, J. (1991). A prospective study of depression and posttraumatic stress symptoms after a natural disaster: The 1989 loma prieta earthquake. *Journal of Personality and Social Psychology*, *61*(1), 115–121.

Seli, P., Risko, E. F., Smilek, D., & Schacter, D. L. (2016). Mind-Wandering With and Without Intention. *Trends in Cognitive Sciences*, *20*(8), 605–617.

Smallwood, J., Davies, J. B., Heim, D., Finnigan, F., Sudberry, M., O'Connor, R., & Obonsawin, M. (2004). Subjective experience and the attentional lapse: Task engagement and disengagement during sustained attention. *Consciousness and Cognition*, *13*, 657-690.

Smallwood, J., & Schooler, J. W. (2015). The Science of Mind Wandering: Empirically Navigating the Stream of Consciousness. *Annual Review of Psychology*, *66*(1), 487–518.

Taatgen, N. A., van Vugt, M. K., Daamen, J., Katidioti, I., & Borst, J. P. (submitted). The resource-availability theory of distraction and mind-wandering.

Treynor, W., Gonzalez, R., & Nolen-Hoeksema, S. (2003). Rumination reconsidered: A psychometric analysis. *Cognitive Therapy and Research*, *27*(3), 247–259.

van Vugt, M. K., & Broers, N. (2016). Self-reported stickiness of mind-wandering affects task performance. *Frontiers in Psychology*, *7*, 732.

van Vugt, M. K., Shahar, B., & Britton, W. (2012). The effects of mindfulness-based cognitive therapy on affective memory recall dynamics in depression: a mechanistic model of rumination. *Frontiers in Human Neuroscience*, *6*, 257.

van Vugt, M. K., Taatgen, N. A., Bastian, M., & Sackur, J. (2015). Modeling mind-wandering: a tool to better understand distraction. In *Proceedings of the International Conference in Cognitive Modeling.* Groningen.

Whitmer, A. J., & Banich, M. T. (2007). Inhibition Versus Switching Deficits in Different Forms of Rumination. *Psychological Science*, *18*(6), 546–553.

Whitmer, A. J., & Gotlib, I. H. (2013). An attentional scope model of rumination. *Psychological Bulletin*, *139*(5), 1036.

Wigman, J. T. W., van Os, J., Borsboom, D., Wardenaar, K. J., Epskamp, S., Klippel, A. K., … Wichers, M. (2015). Exploring the underlying structure of mental disorders: cross-diagnostic differences and similarities from a network perspective using both a top-down and a bottom-up approach. *Psychological Medicine*, *45*, 2375–2387.

A computational cognitive-affective model of decision-making

Christopher L. Dancy (christopher.dancy@bucknell.edu)
Department of Computer Science, 335 Dana Bldg
Lewisburg, PA 17837

David M. Schwartz (dms061@bucknell.edu)
Department of Computer Science, 335 Dana Bldg
Lewisburg, PA 17837

Abstract

How do affective processes interact with cognitive processes to modulate our behavior? Understanding the processes that influence the interactions between affective stimuli and human decision-making behavior is important for predicting typical behavior under a variety of circumstances, from purchasing behavior to deciding when to enact certain rules of engagement in battle scenarios. Though some computational process models have been proposed in the past, they typically focus on higher-level phenomena and are less focused on the particular architectural mechanisms related to the behavior explored. This, in turn, can make it very difficult to combine the proposed model with existing related work (i.e., the models can't be tractably combined).

We used a modified version of the Iowa Gambling Task to explore the effects of subliminal affective (visual) stimuli on decision-making behavior. We developed a model that runs within the ACT-R/Φ architecture that completes the same task completed by participants. In addition to the affective and cognitive memory components particularly important to the discussion, the model also uses perceptual and motor components within the architecture to complete the task. The architecture has representations of *primitive affect* that interact with cognitive memory components mainly through an *affective-associations* module (meant to capture behavior typically ascribed to several amygdalar substructures). The model and affective architectural mechanisms provide a process-oriented explanation for the ways affect may interact with higher-level cognition to mediate human behavior during daily-life.

Keywords: Cognitive Architecture; HumMod; ACT-R; Affect: IGT; Decision-Making; Emotion

Introduction

How do affective processes interact with cognitive processes to modulate our behavior? Though this question is important, we've only seen a relatively recent surge in computational process models that have explored this question (e.g., Marinier III et al., 2009; Marsella et al., 2010). Indeed, even Newell did not have *emotion* (and motivation) as a topic that was most important to address when developing a unified theory of cognition. As more evidence of the importance of emotional/affective processes has accumulated through experimentation and simulation, it has become clear that affect and emotion play a fairly central role in mediating behavior (e.g., Bechara et al., 1997; LeDoux, 2012; Panksepp & Biven, 2012).

We conceive of emotion as an interaction between *affective* and *cognitive* processes. When we make these distinctions we do so with the idea that the two categories describe both qualitative and quantitative differences in computational processes that, nonetheless, interact within a whole computational behavioral system (e.g., see Figure 1 that describes differences in *levels,* Panksepp et al., 2011). We see affective processes as those modulate subsymbolic representations within the cognitive system, which results in certain behavior that may be deemed as *emotional*.

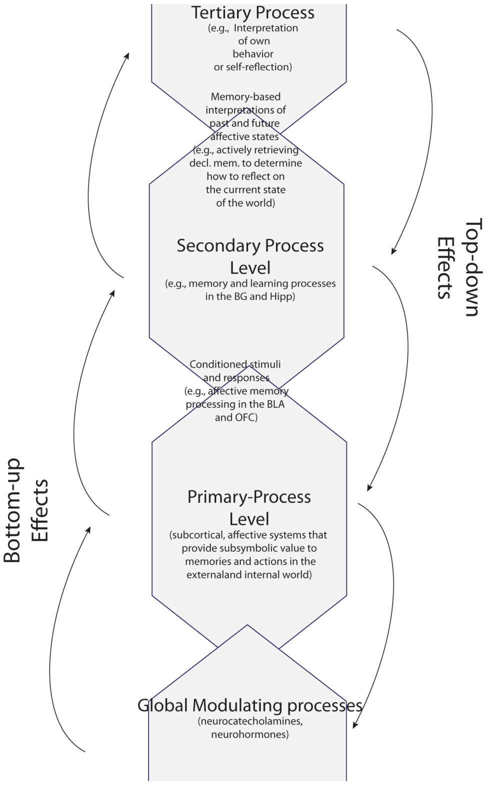

Tertiary Process
(e.g., Interpretation of own behavior or self-reflection)
Memory-based interpretations of past and future affective states (e.g., actively retrieving decl. mem. to determine how to reflect on the currrent state of the world)
Secondary Process Level
(e.g., memory and learning processes in the BG and Hipp)
Conditioned stimuli and responses (e.g., affective memory processing in the BLA and OFC)
Primary-Process Level
(subcortical, affective systems that provide subsymbolic value to memories and actions in the externaland internal world)
Global Modulating processes
(neurocatecholamines, neurohormones)
Top-down Effects
Bottom-up Effects

Figure 1. Levels of behavioral processes from Panksepp et al. (2011)

While some affective processes may have less quantitative effect on symbolic and subsymbolic representations depending on context, the implication here is that no such human interaction is truly without some bias due to affective processes. Ultimately other portions of a conditional context may have more effect on resulting action (e.g., a current *goal/intentional*), however these affective processes still may have small effects on the representations/actions that occur in a computational cognitive system. Put in a more high-level example, just because one may ultimately elect to buy the more economically functional vehicle, doesn't mean that the affective drive to select the new sports car did not factor into the decision.

Below, we give a description of a decision-making study and some results from this study. The study used a modified version of the Iowa Gambling Task (IGT) that involved subliminally presented visual stimuli to explore the particularly non-conscious and subsymbolic effects of affective processes. We also detail and discuss an affective-cognitive model of this task that uses components of the ACT-R/Φ architecture to represent interactions between affective and cognitive processes.

Description of IGT Study and Results

97 undergraduate students were recruited as participants for this study (52 males and 45 females). The average ages of males and females were similar at 20.7 and 19.8 (respectively). Electrodermal Activity (EDA) data were collected for the final 66 (37 males and 29 females) participants (data not reported here). All participants were given college course extra credit for participation.

A filter process that removed participants who completed less than 20% of their trials due to time restrictions (max 3.5s per trial) resulted in the removal of 4 participants' data from further analysis; data from 93 total participants were analyzed. The *negative*, *neutral*, and *positive* (image) groups each had 31 participants. We ceased participant enrollment in the study after we crossed a 31 per-group threshold for task-related behavioral analysis and all volunteers had the opportunity to participate.

Participants used a version of the IGT that included a fixed reward and punishment schedule for each deck that was the same as the schedule used for the original IGT by Bechara et al. (2000). A modified computerized version of the IGT was used that runs in Matlab and uses the Psychtoolbox Matlab extensions (Brainard, 1997). Psychtoolbox extensions were used due to their high timing accuracy, community support, and cross-platform availability and the specific software used has had IGT-specific timing tests done to confirm timing accuracy (Dancy & Ritter, 2016).

The visual stimuli presented during the IGT were obtained from the International Affective Picture System (IAPS; Lang et al., 1997). Table 1 lists the images used in image sets used by the different groups. Male and female pictures were matched so that, for each group, they had similar valence/arousal/dominance ratings and had a similar content subject; for example, some snake pictures had different ratings between sexes within the IAPS manual, so those images with lower valence/higher arousal ratings among the same category were chosen. Given that picture ratings in all categories differed between sexes, this method allowed more consistency in mean measured quantitative ratings among participant sexes.

Table 1. The IAPS images (and the accompanying average valence, arousal, and dominance rating) used in the experiment.

Picture-Set	Picture Numbers
$\text{Negative}_{\text{Male}}$	1050, 1202, 1220, 1304, 1525
$\text{Negative}_{\text{Female}}$	1050, 1120, 1201, 1202, 1525
$\text{Neutral}_{\text{Male}}$	1670, 7006, 7010, 7080, 7175
$\text{Neutral}_{\text{Female}}$	1670, 7004, 7010, 7012, 7175
$\text{Positive}_{\text{Male}}$	4180, 4210, 4232, 4664, 8501
$\text{Positive}_{\text{Female}}$	4505, 4525, 4660, 8001, 8501

Before participating in the study, all participants read and signed a consent form approved by the Office of Research Protections (ORP) at Penn State. Participants were assigned to one of three possible groups (with different accompanying treatments): a *negative group* with a negative image treatment, a *neutral group* with a neutral image treatment, or a *positive group* with positive image treatment. Images (consistent with participant sex and group) were presented to participants for 17ms after deck selection if they selected from one of the *bad decks* (those that give a negative net amount of money) and plain gray images were presented for the same amount of time if a card selection was made from one of the other two decks. For a full explanation of the typical IGT procedure, see (Bechara et al., 2000).

Results

As with previous IGT-based studies we split deck selection analysis into five blocks, 20 deck selections per block. *Score* was calculated by subtracting the total number of card selections from decks A and B (the *bad decks*) from the total number of card selections from decks C and D.

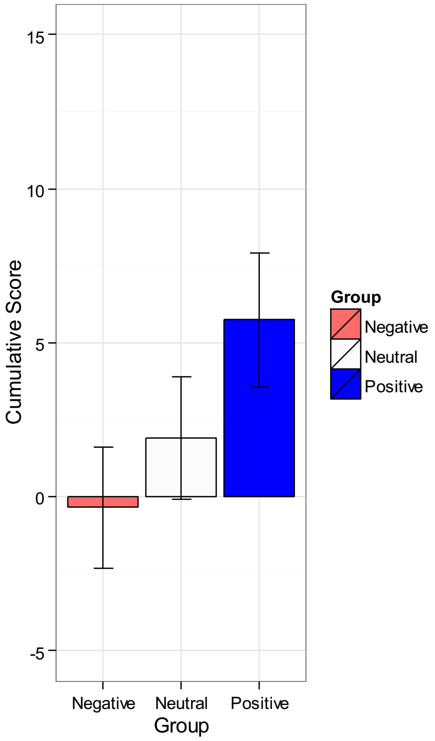


Figure 2. Cumulative score (±SEM) for all participants after the final block

Participants in the positive image group showed the highest score (Figure 2) when averaged across blocks, but all groups had a positive score by the final block (Table 2). Scores increased for all groups from blocks 1-3 and blocks 4-5, but decreased from blocks 3-4.

Table 2. Mean score of participants in all of the blocks by group. Standard errors are in presented in the parenthesis

Group	B1	B2	B3	B4	B5
Negative	-4.2	0.2	1.1	-0.2	2.7
	(1.4)	(1.3)	(1.5)	(1.3)	(1.4)
Neutral	-3.5	-0.3	2.4	1.5	1.8
	(1.1)	(0.9)	(0.8)	(1.0)	(1.1)
Positive	-3.3	-0.5	3.0	2.6	4.0
	(1.3)	(1.2)	(1.2)	(1.1)	(1.0)

A $3X5$ (group by block) mixed factor ANOVA of participant score revealed a highly significant effect of block ($F(4,\ 360)\ =\ 13.22, p\ <\ .0001$) on score, however it did not reveal a significant group ($F(2,\ 90)\ =\ 0.81,\ p\ =\ .4$) or a group:block interaction ($F(8,\ 360)\ =\ 0.40,\ p\ =\ .9$) effect.

When sex is also taken into account, males and females show an opposite score distribution across groups (Figure 3).

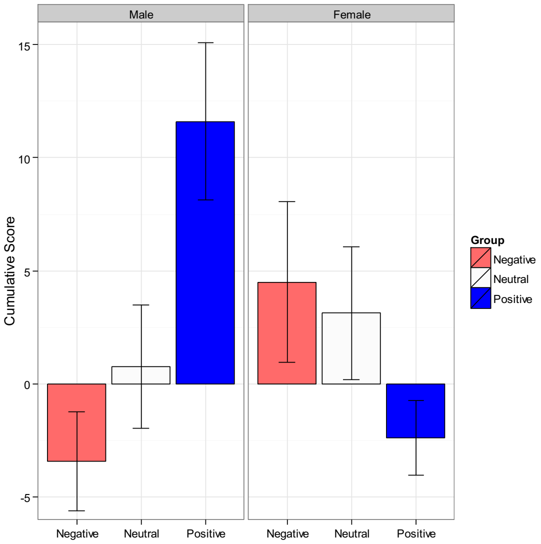


Figure 3. Cumulative score for male (left) and female (right) participants after the final block.

Among male participants, those in the positive group showed the highest cumulative score and those in the negative group showed the lowest score (the only negative among male participants). Conversely, among female participants, those in the positive group received the lowest score (the only negative among female participants), while those in the negative group received the highest scores.

The decision-making model

To simulate this task and potentially understand more about the processes that mediates behavior during this task (and others that show some effects of subliminal affective stimuli), we developed a cognitive-affective model that runs within the ACT-R/Φ architecture. This model uses simulated eyes and hands to perceive the task (e.g., see the decks, cards, rewards, and affective images) and provide feedback (e.g., press a key to select a card from a deck). To make decisions, the model uses both procedural and declarative memory (Figure 4).

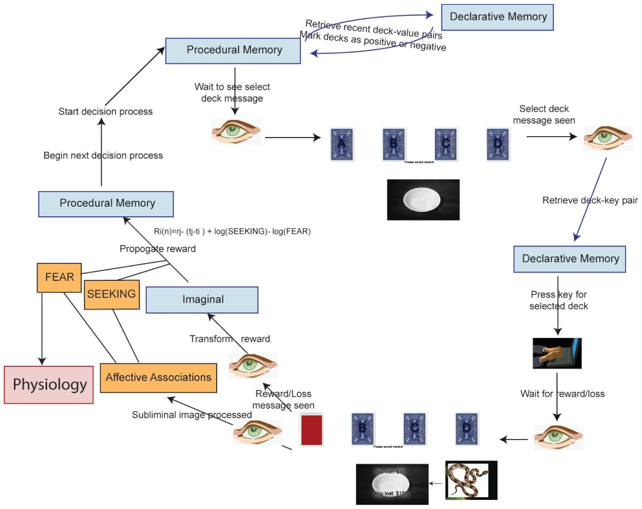


Figure 4. A high-level diagram of the ACT-R/Φ model

After the model has made a deck selection and a reward/loss is shown, it uses those values to reinforce the utility of those production rules recently fired:

$$U_i(n) = U_i(n-1) + \alpha\big(R_i(n) - U_i(n-1)\big) \qquad [1]$$

$$R_i(n) = r_j - (t_j - t_i) + \log(Value_{SEEKING}) - \log\,(Value_{FEAR}) \qquad [2]$$

$$val(t) = W_{rew} * rew(t)^{\gamma} \text{ - } W_{aversion} * loss(t)^{\gamma} \qquad [3]$$

Here, $U_i(n-1)$ represents the current utility value, α is a learning rate, and $R_i(n)$ is a reward that is determined by equation 2. In equation 2 r_j represents the reward received and $t_j - t_i$ is the temporal discount that is given to a reward so that the length of time between reward onset and a rule firing determines how much reward is applied to the utility value. $Value_{SEEKING}$ and $Value_{FEAR}$ in equation 2 are reward offsets that take into account the current affective state of the model (see Dancy, 2013) for some description of the modules/systems in ACT-R/Φ that control these values. Though the SEEKING system can be affected by several things in a realistic environment (e.g., the model would see an increase in SEEKING activation/value if it were thirsty), the limited scope of this model means that the SEEKING and FEAR values are practically determined by the emotional images flashed after selecting a card from a *bad deck*; this is controlled by the affective-associations module in ACT-R/ Φ (which is shown in Figure 4. The affective value for the images flashed is derived from equations 4 and 5, which use the values for arousal, valence, and dominance from the images listed in Table 1 (specific values are available in the IAPS manual, Lang et al., 1997).

$$FEAR_{value} = \frac{(arousal + \varepsilon) * (10 - (valence + \varepsilon) - (dominance + \varepsilon))}{90} \qquad [4]$$

$$SEEKING_{value} = \frac{(arousal + \varepsilon) * ((valence + \varepsilon) + (dominance + \varepsilon) - 10)}{90} \qquad [5]$$

The actual reward (i.e., r_j in equation 2) is determined by a function that transforms the gain and loss that results from selecting a card from a deck on a given trial, using the imaginal/imaginal-action buffers. This function implements equation 6 below, which is a slightly modified version of an equation discussed by Ahn et al. (2008) and proposed by Napoli and Fum (2010) to be used in ACT-R.

$$val(t) = W_{rew} * rew(t)^{\gamma} \text{ - } W_{aversion} * loss(t)^{\gamma} \qquad [6]$$

These equations all reinforce production rules, which can cause some production rules to be less likely to fire over time (those that consistently have a negative reward, and consequently a lower utility will decrease in likelihood of firing overtime). This selection rule is encapsulated in equation 7.

$$P(i) = \frac{e^{U_i/\sqrt{2s}}}{\sum_j e^{U_j/\sqrt{2s}}} \qquad [7]$$

$P(i)$ is the probability of selecting rule *i* which is determined by comparative weight of the rule *i* as well as any *procedural noise* (represented as *:egs* in canonical ACT-R).

The model also uses declarative memory to learn and make decisions. It encodes deck-value pairs and these pairs ultimately control which decks are selected. At the beginning of making a deck selection, the model queries declarative memory for deck-value pairs for each of the decks. The declarative memory elements with the highest activation (governed by equation 8) are selected.

$$A_i = B_i + S_i + P_i + \varepsilon_i \qquad [8]$$

$$B_i = \ln\Big(\textstyle\sum_{j=1}^{n} t_j^{-d}\Big) + \beta_i \qquad [9]$$

Thus, the model uses the majority of the major components of the original ACT-R architecture (including perceptual and motor systems).

Model Comparison to Study Results

The IGT model was run a total of 360 times, 120 for each of the negative, neutral, and positive image groups. Half of the model runs within each group were male, while the other half were female. Thus, this resulted in 60 unique runs of the model. Because the model was originally developed as a prediction of the processes occurring and those behavioral data that result from such processes, sex-based differences were not a focus. In the model, the distinction between male and female only comes into play with the affectively valued visual stimuli (which nonetheless have very similar values.)

For all groups the, the model predicted a similar scoring trend (positive) from blocks 1 to 3 (Figure 5). Overall the negative model seemed to deviate the most from the actual observed score by participants across decks.

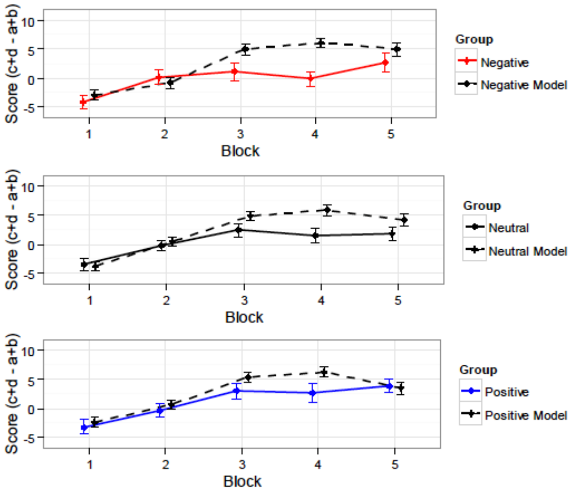


Figure 5. Comparison of score performance between participants and the model for negative, neutral, and positive groups

The neutral model predicted the scores for the five blocks best followed by the positive and negative models (Table 3).

Table 3. Comparison between model predictions for the different groups.

Model/Group	r^2	$RMSD$
$\text{IGT}_{\text{Negative}}$/Negative	.56	3.48
$\text{IGT}_{\text{Neutral}}$/Neutral	.94	2.49
$\text{IGT}_{\text{positive}}$/Positive	.81	2.05
All	0.72	2.74

Discussion and Conclusion

The model fit best to those data from the participants in the neutral group best, though the model did also fit reasonably well to those data from the positive group. It would seem that there is a key point of change that the model does not exhibit (i.e., in block 4). The model continues on the positive trend, as exhibited in previous blocks, while participants show a dip in performance during this block. Because the model does not switch deck selection in the same way participants do (and thus, continues on a *greedy* path), the model tended to exhaust decks at a certain point, causing the dip in performance seen in the final block.

The model appears to have underestimated the effects of the subliminally presented affective stimuli. While, the affective stimuli did have certain subtle (subsymbolic) effects through the affective-associations module, those participant data showed a much more overt effect on performance. What's more, these behavioral effects seemed to have some dependency on participant sex, for which the model had very little account.

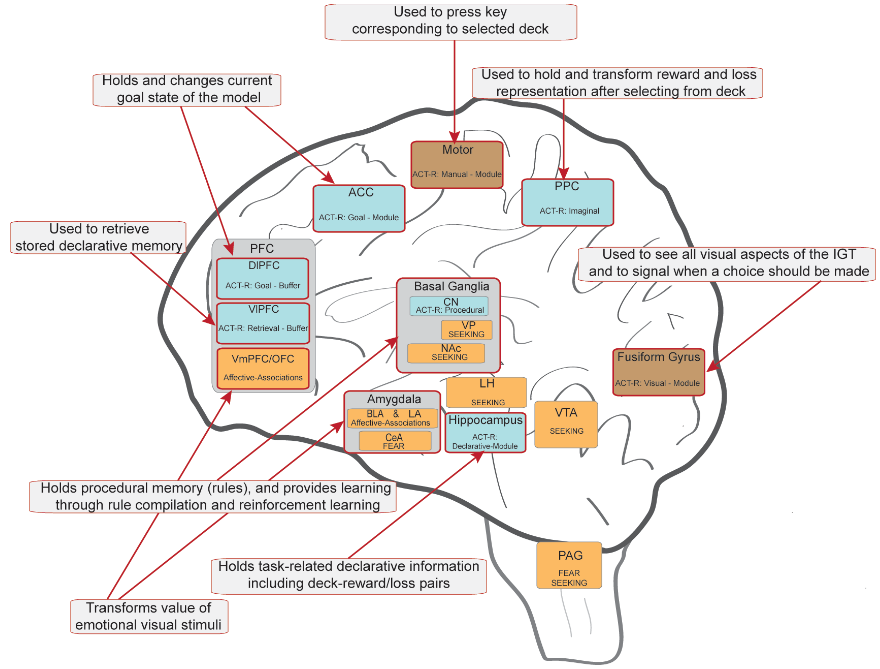

Used to press key corresponding to selected deck
Holds and changes current goal state of the model
Used to hold and transform reward and loss representation after selecting from deck
Motor
ACT-R: Manual - Module
ACC
ACT-R: Goal - Module
PPC
ACT-R: Imaginal
Used to retrieve stored declarative memory
PFC
DlPFC
ACT-R: Goal - Buffer
VlPFC
ACT-R: Retrieval - Buffer
VmPFC/OFC
Affective-Associations
Used to see all visual aspects of the IGT and to signal when a choice should be made
Basal Ganglia
CN
ACT-R: Procedural
VP
SEEKING
NAc
SEEKING
LH
SEEKING
Fusiform Gyrus
ACT-R: Visual - Module
Amygdala
BLA & LA
Affective-Associations
CeA
FEAR
Hippocampus
ACT-R: Declarative-Module
VTA
SEEKING
Holds procedural memory (rules), and provides learning through rule compilation and reinforcement learning
Holds task-related declarative information including deck-reward/loss pairs
PAG
FEAR
SEEKING
Transforms value of emotional visual stimuli

Figure 6. Predicted brain areas and main functions by the model/architecture.

Though the model did not predict several aspects of these presented data, it provides a useful framework for future work and related simulation. Indeed, using a system like ACT-R/Phi for the simulation also allows one to provide early predictions of brain areas involved in related affective decisions (Figure 1). This is due to the architectures use of ACT-R theory (which has various functional modules that have been associated with certain neural structures, Anderson, 2007) and theory from affective neuroscience (e.g., Panksepp, 1998; Panksepp & Biven, 2012). The predictions from Figure 6 can be further explored in future studies. Future plans for this particular model include running a ranging parameter sweep on potentially varying parameters (e.g., W_{rew} and $W_{aversion}$) to see if the model can more closely fit to these data presented here.

Existing theory, data, and these data presented here make it clear that affective processes can have an overt effect on decision-making behavior, even when the affective stimuli causing the activation of such processes isn't overt. It is important to understand these effects as they can be useful to positively, or negatively, influence our decisions in various ways that may fail to reach our awareness. The model and mechanisms presented provides a first step towards providing a more systematic and unified account of the modulating effects of affective stimuli on cognitive behavior.

Acknowledgments

Some of this work was supported by the Sloan graduate scholarship and the Bucknell Scholarly Development Grant.

References

Ahn, W., Busemeyer, J. R., Wagenmakers, E., & Stout, J. C. (2008). Comparison of decision learning models using the generalization criterion method. *Cognitive Science, 32*(8), 1376-1402.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: OUP.

Bechara, A., Damasio, H., & Damasio, A. R. (2000). Emotion, decision making and the orbitofrontal cortex. *Cerebral cortex, 10*(3), 295.

Bechara, A., Damasio, H., Tranel, D., & Damasio, A. R. (1997). Deciding advantageously before knowing the advantageous strategy. *Science, 275*(5304), 1293-1295.

Brainard, D. H. (1997). The Psychophysics toolbox. *Spatial Vision, 10*(4), 433-436.

Dancy, C. L. (2013). ACT-RΦ: A cognitive architecture with physiology and affect. *Biologically Inspired Cognitive Architectures, 6*(1), 40–45.

Dancy, C. L., & Ritter, F. E. (2016). IGT-Open: An open-source, computerized version of the Iowa Gambling Task. *Behavior research methods*, 1-7.

Lang, P. J., Bradley, M. M., & Cuthbert, B. N. (1997). *International Affective Picture System (IAPS): Technical manual and affective ratings*. The Center for Research in Psychophysiology, University of Florida. Gainesville, FL.

LeDoux, J. E. (2012). Rethinking the Emotional Brain. *Neuron, 73*(4), 653-676.

Marinier III, R. P., Laird, J. E., & Lewis, R. L. (2009). A computational unification of cognitive behavior and emotion. *Cognitive Systems Research, 10*(1), 48-69.

Marsella, S., Gratch, J., & Petta, P. (2010). Computational models of emotion. In K. R. Scherer, T. Banziger, & E. B. Roesch (Eds.), *A blueprint for affective computing: Crossfertilization between emotion psychology, affective neuroscience, and affective computing,* . New York, NY: OUP.

Napoli, A., & Fum, D. (2010). Rewards and punishments in iterated decision making: An explanation for the frequency of the contingent event effect. *In proceedings of the International Conference on Cognitive Modeling*, Philadelphia, PA.

Panksepp, J. (1998). *Affective neuroscience: The foundations of human and animal emotions*. New York, NY: OUP.

Panksepp, J., & Biven, L. (2012). *The Archeology of Mind: Neuroevoloutionary Origins of Human Emotions*. New York, NY: W.W. Norton & Company.

Panksepp, J., Fuchs, T., & Iacobucci, P. (2011). The basic neuroscience of emotional experiences in mammals: The case of subcortical FEAR circuitry and implications for clinical anxiety. *Applied Animal Behaviour Science, 129*(1), 1-17.

A New Direction for Attachment Modelling: Simulating Q Set Descriptors

Dean Petters (dean.petters@bcu.ac.uk)
Department of Psychology, 4 Cardigan Street
Birmingham, B4 7BD, UK

Abstract

Modelling infant-carer attachment relationships is an emerging field at the intersection of research in Attachment Theory and computational modelling of emotion. Existing attachment models vary from very abstract models to simulations of specific experimental protocols, such as the Strange Situation Procedure. This paper argues for the benefits in broadening attachment modelling of infants and young children to also include simulating attachment Q set descriptors. The attachment Q set (AQS) is a 90 item list of attachment related behaviors used to assess the balance between attachment and exploratory behavior in home and other naturalistic settings. The AQS descriptors provide a broader and more rounded challenge for attachment modelling than other types of systematic attachment measure because they can be observed in naturalistic contexts and are less dependent on the specific details of laboratory settings. A computational attachment model is presented from which a selection of 8 attachment Q set descriptors will be simulated. Thsee initial descriptors to be simulated are concerned with the time an infant takes to recover from anxiety. A 'route map' for progress towards capturing all 90 Q sort descriptors is discussed.

Keywords: Attachment Theory; Attachment Modelling; Agent-based modelling; Attachment Q sort

Introduction

Attachment Theory describes and explains the nature of emotional bonds which form in close relationships (Cassidy & Shaver,2016). There are a small but growing number of computational attachment models which have been implemented as software and robotic simulation. For recent reviews see (Petters & Waters,2015) and (Petters & Beaudoin,2017). This paper will illustrate how empirical data in the form of attachment Q set (AQS) descriptors is well suited for the purposes of forming test-cases in scenarios and specification of requirements for attachment models. Two key contributions of this paper are that (i) it illustrates different ways that empirical data that can be used for modelling affective phenomena, and in particular it highlights the constraints and biases for simulations in this domain; and (ii) it provides an examples of how an existing simulation has been adapted to model Q set descriptors.

A short introduction to Attachment Theory

In its early theoretical development, an idea which was important in distinguishing Attachment Theory from learning theory is that attachment between an infant and main caregiver is a rich 'love' relationship (Bowlby,1969). This means that whilst attachment relationships can be tracked by observable behaviour patterns, attachment arises from a complex internal information processing architecture, termed by Bowlby the 'attachment control system' (ACS) (Bowlby,1969). The ACS acts to maintain a balance between attachment behaviour and exploration. Cues to danger momentarily move this balance from exploration to attachment. Over longer ontogenetic timescales, the complex organisation of attachment behaviour is sensitive to environmental factors. This means that both normative routines and individual difference patterns of attachment are learnt, with the aid perhaps of some evolutionary biases in infants' learning abilities. Individual differences in attachment are conceptualised as differences in an individual's ability to use their attachment figure as a secure-base. This means that the attachment-exploration balance for any individual reflects its past history of sensitive and effective responses by its caregiver in support of exploration and when the infant is distressed. (Petters,2006a).

Initially in ontogenetic development, the ACS is composed of relatively simple mechanisms, such as reflexes and fixed action patterns. However, later in development the ACS becomes comprised of a diverse range of information processing structures and mechanisms, from simple reflexes to goal corrected mechanisms and processes of planning, deliberation about future consequences of possible actions, and representing aspects of the self and environment in natural language to facilitate these processes and to communicate with others (Bowlby,1969;Petters,2006a). In addition to better capturing the behavioral complexity and underlying processes in play during infant-mother interactions, viewing attachment in control system terms clarifies assessment criteria. In principle, it is much easier to evaluate whether an attachment system is tracking set goals such as maintaining access to the carer or regulating affect than to evaluate the "qualities" of attachment as a dyadic relationship or a social network. (Waters & Deane,1985).

Whilst Bowlby set out the details of the ACS, Ainsworth and co-workers initiated the 'individual difference phase' of attachment research by developing the Strange Situation Procedure (SSP) (Ainsworth, Blehar, Waters, & Wall,1978). The SSP is conducted in a 4m x 4m room with chairs for two adults, toys for the infant to explore, and one-way glass for observation and video recording. The assessment is divided into 8 three minute episodes. At two critical points, the carer leaves the infant in the room for three minutes (once with a responsive but unfamiliar adult and once all alone). In early research, it was thought that response to these separation episodes (esp. crying) would be the best predictor of prior experience with the carer and of later adjustment. However, smooth adaptive responses to reunion (as opposed to anger or avoidance) soon proved to be much more revealing of home environment. The context changes that occur in the transitions between the eight episodes, and the infant's responses to these transitions provide a valuable data-set for contemporary researchers interested in designing attachment behaviour simulations (Petters,2006a,2006b).

Normative behaviour patterns across episodes highlight the infant's sensitivity to context that would be difficult to explain in terms of traits or operant control and justify the use of a control systems approach (e.g., more play, different kinds of signalling, less proximity seeking when carer is present) . Although the SSP assesses rather complex behavior, it does so in a restricted context and time frame. Therefore, it has been important to validate SSP based assessments against observations in more naturalistic settings and over longer time intervals. Ainsworth undertook this using detailed ethological observations, For each dyad, infant and maternal behaviour observed in home for up to 16 hours toward the end of the infant's first year (Ainsworth et al.,1978). The creation of the SSP triggered the development of a huge number of diverse measurement tools in attachment research, ranging from trait questionnaire measures similar to those used in personality research to the AQS methodology, which can be compared with the ethograms used in ethological research. More recently, Waters and Deane developed a more economical method for observing and quantifying infant-mother interactions. Their AQS descriptors cover the full range of attachment and exploratory behaviors that Ainworth recorded. However, rather than generating narrative records of the observations, the items are scored and compared to a template that describes skillful, well-organized use of the carer as a secure base (Waters & Deane,1985).

Different ways to model attachment

Attachment phenomena have been modelled in a very abstract fashion using Artificial Neural Nets (ANNs) (Fraley,2007;Edalat & Mancinelli,2013). In these attachment models the ANN can be viewed as an extremely abstract representation of an individual. The 'experiences' and 'behaviour' of the individuals in these simulations are also extremely abstract, being constituted of data that are an independent sequence of discrete training exemplar and response pairs. The main result (finding) of these simulations matches the high level of abstraction that these models have been created at. This is that in these artificial neural network simulations early prototypes are not over-written, and so show greater continuity, when new relationship experiences are inconsistent. But consistent presentation of new prototypes does result in gradual change (Fraley,2007;Edalat & Mancinelli,2013).

Agent-based models have also simulated the SSP (Petters & Waters,2015) and infant secure-base behaviour. These models are less abstract then the models based on neural nets. The main result (finding) from these simulations is that within a design space for attachment architectures, some attachment architectures show system properties like sensitivity to initial conditions (c.f. the butterfly effect) and saddle points in developmental trajectories (Petters,2006a,2006b). So where the neural networks learn item by item in 'batch jobs', and provide a result in terms of how many new learning experiences it takes to undo existing learning, the agent-based models exist in online dynamically changing virtual environments and provide results consonant with this type of dynamic simulation. In the agent based modelling case, inputs to an agent at any given time are contingent on what occurred the moment before. This means that these simulations help explain findings in terms of repeated contingent interactions that result in positive feedback driving the system away from its starting conditions towards extreme levels of 'secure' or 'insecure' interactions.

In summary, whereas the ANN results describe change in an internal representation acted upon by an independent sequence of 'offline' discrete training exemplars, agent-based modelling (ABM) results follow the changing trajectory for an agent in a broader system as that agent is acted upon and in turn influences the broader system in 'online' fashion. These findings illustrate a key principle in the art and science of cognitive modelling is the importance in finding the right level of abstraction for a simulation. This paper is concerned with discussing the benefits and drawbacks for attachment modelling in taking various approaches to deciding upon an abstraction level for computational attachment models. The paper introduces the AQS as a source of empirical constraints and requirements specifications not used before in attachment modelling. It will illustrate how modelling AQS data will provide some specific benefits over simulating other sources of information in the form of trait measures, frequency and time sampling data, and the SSP.

The nature of empirical data constrains the nature of the simulation

The importance of structural fidelity

Gaining structural fidelity is an important objective when constructing psychological measurement tools, such as personality scales and related questionnaires (Simms & Watson,2010). This is because any behavioural measure should provide data congruent with the type of construct it is designed to assess. There are two aspects to structural fidelity (Simms & Watson,2010). The first is a structural component of construct validity which requires that structural relations between the chosen test items in the measurement tool parallel structural relations for other manifestations of the construct in question, which did not get chosen to be test items. So this is a requirement that test items are representative of the possible manifestations of the construct in terms of their structural relations (Simms & Watson,2010). This aspect of test item choice is clearly relevant to the computational modeller. To produce a model based on the underlying phenomenon rather than arbitrary aspects of observed data a modeller should not abstract and simulate test items that systematically differ from other manifestations of the construct they intend to model. The second aspect of structural fidelity regards the assumptions underlying the chosen test set matching the theoretical model underlying the construct (Simms & Watson,2010). Loevinger (1957 cited in (Clark & Watson,1995)) was the first researcher to highlight these issues, and contrasted scales and tools which were based on a *"deeper knowledge of psychological theory"* (Loevinger 1957, p. 641, cited in (Clark & Watson,1995)) with tools based on an atheoretic "answer-based" technology (Clark & Watson,1995). Clearly, for the computational modeller this issue is critical. When possible, computational modellers should draw upon empirical data that align with appropriate underlying theory.

Limitations of trait rating, frequency counts and time sampling behavioural measures for attachment modelling

Trait measures are flexible and economical, take context into account, and demonstrate coherence over time (Waters & Deane,1985). However, they are not suited to assessing non-quantitative data and they score low in structural fidelity because attachment is not a trait. Waters and Deane note: *"trait language should only be used to summarise behaviour - never as a substitute; never as an explanation"* ((Waters & Deane,1985), p.44). Waters and Deane suggest that trait rating are coercive and conservative in forcing researchers to view constructs in terms of pre-existing scales, and working against introduction of new scales or measures during the process of measurement (Waters & Deane,1985). It is also difficult to disentangle affect and cognition in trait rating data (Waters & Deane,1985). Observational data in the form of frequency counts and time sampling retain much more behavioural detail than trait rating methods. They also have good structural fidelity. However, their expense and difficulty to use mean that often only small numbers of behavioural categories are assessed in a single study. In addition, particularly interesting behaviours may occur at very low frequencies (Waters & Deane,1985). This means that they are a very good way of getting very detailed data on behaviours of specific interest if those behaviours occur relatively frequently. For practical reasons of resources and time, what these methods is not so good for is gaining a comprehensive overview of an entire behaviour domain that possessed many different salient behaviour types (Waters & Deane,1985).

Limitations of the SSP as a source for attachment modelling

The SSP (Ainsworth et al.,1978) involves a set of scoring protocols that includes behaviour coding, frequency and percentage measures. All these measures were developed to provide insight in to the underlying ACS. This is done partly by including separation and reunion episodes which are mildly to moderately stressful as a way of activating the ACS in a controlled manner. Because the procedure is designed specifically to uncover the state of an infant's ACS, the SSP affords very high structural fidelity. However, the behaviours produced in the SSP do not correlate directly to behaviours in naturalistic settings like the home environment. For example, crying rate in the SSP does not predict the rate of these behaviours at home. Rather, the behaviours produced in the SSP are used to infer the state of the ACS, and an ACS in this state will produce different behaviour patterns depending on context. Other limitations of the SSP for psychological research include the narrow age range it can be used (21-18 months), strong carry over effects (infants recognise the context if it is repeated soon after), the expense and difficulty of administration, and it does not capture developmental change well (Waters & Deane,1985). For the attachment modeller, it is also too narrow in measures used and number of contexts it describes.

Limitations of modelling in a 'method-bound' research domain

A further limitation for attachment modelling related to attachment measures arising from the large variety of attachment measures currently available. It might be imagined that having numerous attachment measures to choose from would help the attachment modeller. However, the current situation has given rise to what Fonagy terms the 'method-bound' nature of Attachment Theory:

> *"Attachment theory* [...] *has been in some ways method-bound over the past 15 years. Its scope was determined less by what fell within the domain defined by relationship phenomena involving a caretaking-dependent dyad and more by the range of groups and behaviors to which the preferred mode of observation, the strange situation, the adult attachment interview, and so forth, could be productively applied."*((Fonagy,1999), p. 5)

There is therefore drawback in attempting to model behaviour in a domain which is 'method-bound'. If the methods leave gaps in empirical data coverage, the gaps will not get modelled. So a researcher interested in behaviour will need to consider carefully how to get a representative sample of behaviour in this kind of domain. The next section describes the AQS, which overcomes the limitations for attachment modelling of behaviours described as traits, frequency counts, time samples and SSP patterns. It also allows strong structural fidelity between observable behaviours and internal processes and provides more comprehensive coverage of attachment any other single measure.

Overview of Attachment Q Set Behavioural Descriptors

Q sort methodology can be applied to research in any given area of behavioural science (Waters & Deane,1985). First, it involves developing a set of descriptive items. These should ideally be extensive enough to be an overview of the entire behavioural domain of interest. For example, Waters and Deane spent two years developing a 100 item Q set for infant attachment. They reviewed relevant literature; developed a list of relevant constructs (security, dependency, detachment, self-efficacy, aspects of object orientation, communication skills, predominant mood, response to physical comforting, fearfulness, anger and trust); rated infants and toddlers on these variables and then specified the behaviour that led to or was congruent with these ratings (Waters & Deane,1985). This is important because it provides an emphasis on simulating ordinary as opposed to traumatic experiences when attempting to model the development of information processing architectures for attachment. When 'ordinary' architecture development can be modelled, trauma modelling can follow.

Each AQS item refers to a particular behaviour patterns in a specific context. As Waters and Deane note, because the AQS *"covers a broad range of secure-base and exploratory behavior, affective response, social-referencing and other aspects of social cognition* [...] *it can be construed as an overview of the entire domain of attachment relevant behavior, as currently understood within an ethological/control systems perspective"* ((Waters & Deane,1985), p. 7. This means that the AQS captures a more comprehensive description of attachment relevant behaviour that other behavioural measures that might be used in computational modelling.

What is of particular interest for attachment modelling is how the AQS descriptors were initially constructed. Waters and Deane describe four stages in the initial development of the attachment Q-set (Waters & Deane,1985). These four stages involve procedures for developing sets of items which empirical psychologists use when observing behaviour, processing and analysing behaviour, and ultimately producing a classification for the individual observed. However, computational modellers can use these descriptions to construct representative behavioural scenarios from which to direct model design and simulation implementation, and guide model evaluation and validation.

The first stage of Q-set production is of most interest to computational modellers because it involves procedures for developing sets of items. Developing a Q set requires careful examination of extensive observational data. Even when initial descriptor sets are produced they need to be trialled to weed out highly correlated descriptor pairs (Waters & Deane,1985). It also requires close attention to detail, focusing on distinctions and ambiguities that may not be apparent in measurement tools at a higher abstraction level (Waters & Deane,1985). One of the major advantages in this methodology for empirical psychologists is that observers new to the domain will evaluate the same context as the experts who designed and calibrated the AQS (Waters & Deane,1985). This is precisely the property of a measurement tool that computational modellers require: providing broad and comprehensive coverage but also focused on behaviour of interest, filtering out irrelevant behaviours from analysis, and a level of clarity in actions and context that a novice can understand (and learn from). Another major advantage of the AQS methodology is it gives an helpfully strong focus on the role of context, and effectively defines behaviours as *"acts plus context"* as context is integral to each Q set item (Vaughn, Waters and Teti,forthcoming).

The second stage of a Q-sort methodology involves then assigning scores to descriptors when assessing individual study participants, depending how well the participants matches the behaviour. Then the third stage of a Q-sort methodology involves data reduction and analysis and there are a wide variety of procedures for doing this (Waters & Deane,1985). The Q-set methodology allows an infant or child's behaviour to be observed and measured so that it gives a set of scores which can be correlated against a hypothetical 'most secure baby' Q-sort. So a very secure Q set would give a high correlation (around r = 0.6 of a theoretical maximum of 1). Very insecure infants give correlations around r = 0, because insecure behaviour does not involve doing the exact opposite of secure behaviour. Details of how the Attachment Q sort procedure is actually used in empirical research by psychologists to assess infants is beyond the scope of this attachment modelling paper but described in more detail by Waters and Deane (Waters & Deane,1985), with the full set of Q sort items listed by Waters (Waters,1987).

Unlike other measurement tools, the AQS provides an abstract generalised template for computational attachment modellers. As Vaughn, Water and Teti note, it is similar to an 'ethogram', because it is *"rooted in observation and attempts to catalogue the full suite of behaviors associated with a particular behavioral system"*.(Vaughn, Waters and Teti,forthcoming, p.14).

Modelling results

Modelling of Q set descriptors has been undertaken using an existing agent-based model of the SSP as a point of departure (Petters,2006a,2006b). Figure 1 shows a hybrid infant architecture with reactive components and a simple deliberative subsystem. This architecture simulates the SSP by 'experiencing' the pattern of caregiving in a home 'training' stage and then producing typical SSP behavioural patterns in a 'test' stage. It has been used as the basis for implementing AQS descriptors by being augmented with further perceptual, memory and action mechanisms.

The most recent version of the AQS has 90 behaviour descriptors (Waters,1987). Waters and Deane present these items as a single list. The first task that has been undertaken in this current research is to analyse these descriptors to assess the best order to place them in a 'route map' for eventually capturing all Q sort descriptors in a single implemented simulation. So for this current modelling effort, the AQS descriptor list has been analysed into three main sets of descriptors: those that could be modelled by the existing agent-based architecture with manageable extensions to that architecture (20 items); those that were well beyond the capabilities of the existing implemented agent architecture and would require a significantly more sophisticated architecture to be simulated (35 items); and 'filler' items not linked to attachment phenomena and which were added to the Q set for pragmatic reasons to make the AQS sorting procedure run more smoothly (35 items) (Waters & Deane,1985). There were two main reasons that items were assessed as being significantly beyond the capabilities of the existing simulation: that the descriptor

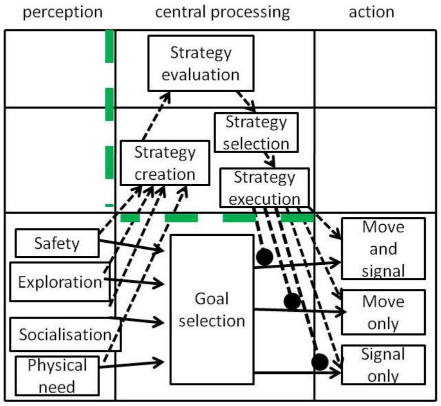


Figure 1: A hybrid attachment architecture with reactive, deliberative and meta-management subsystems. This archiecture has been extended to store in infant agent memory the details of episodes when it interacts with carer agents, with the responsiveness and sensitivity of the interaction stored along with other details of context

required a more sophisticated perception and understanding of others than is currently implemented in existing attachment simulations (for example, AQS descriptor 42: *'Child recognizes when mother is upset. Becomes quiet or upset himself. Tries to comfort her. Asks what is wrong, etc.'*); and that the descriptor requires a more complicated model of the simulated world than is currently implemented (for example, AQS descriptor 53: *'Child puts his arms around mother or puts his hand on her shoulder when she picks him up'*. Filler items included AQS descriptors such as number 89: *'Childs facial expressions are strong and clear when he is playing with something'*.

The 20 items in the set of AQS descriptors which were assessed as being able to implemented with an extension of the existing agent-based model have been categorised into 7 subsets focusing upon: affective communication (2 items); predisposition to cry or be demanding (2 items); the interplay of exploration, anxiety and relief (2 items); aspects of physical need (3 items); how sensation and perception operate in the attachment domain (2 items); time to become anxious (2 items); and time to recover from anxiety (8 items). Figure 2 presents the eight AQS items concerned with the subset of descriptors concerned with 'time to recover from anxiety'. The initial modelling in this AQS simulation project has concentrated on capturing these eight descriptors. This has been done by implemented extra percptual, memory and action mechanisms to support simulation of infant expectations about the immediate future likely responses of the carer agent.

The existing agent-based model of the SSP (in figure 1) already simulates individual differences in the behavioural patterns that result when infant agents return to the proximity of their carer agent after a separation (Petters,2006a,2006b). This occurs because the existing agent-based model possesses 'behaviours' for attachment proximity, exploration, social need, and physical need. These all operate independently and in parallel in proposing new active action goals for the agent. The action selection mechanism is a 'winner-take-all' mechanism which selects the candidate goal with the highest activation. The 'behaviour' subsystem for attachment anxiety goal is activated when the distance between the infant agent and carer agent is beyond a parameter termed the 'safe-range'. This safe-range parameter is learned from the results of all previous episodes when the infant agent has attachment anxiety as its active 'behaviour' goal. If the infant agent has experienced a history of prompt and sensitive responses from its carer agent it will have a large 'safe-range'. This means that the carer agent can move further away before the infant agent's attachment anxiety 'behaviour' goal starts to become activated. If the infant agent has experienced a history of tardy and insensitive responses to its requests for proximity and attention then it will have a small 'safe-range'. This not only means that attachment anxiety will be experienced more often, but that anxiety will take longer to drop back to a normal value when reunions occur. However, the 'safe-range' parameter is a very economical record of previous interactions because the results of the quality of interaction in all different contexts are collapsed into a single numerical value. What the newly implemented 'AQS' extensions to the existing simulations involve is the recording of much more context for each individual episode where attachment anxiety becomes the active goal and the infant agent records carer agent responsiveness and sensitivity. In the 'AQS' extension architecture when the infant agent experiences an episode of active attachment anxiety and signals and moves to reduce its anxiety level the context at initiation and conclusion of the goal is recorded. This context includes external measures, such as the agents and objects present in sensory data, and also internal context, such as relative activations for inactive goals, such as physical and social need. This means that when a new episode of anxiety is experienced this more detailed and specific 'episodic memory' is available to influence responses in a 'recovering from anxiety' time period. This mechanism will therefore support simulating the expectations of carer response apparent in the AQS 'time to recover from anxiety' descriptor subset. The production of the AQS simulation extension is a work-in-progress with the aim of ultimately capturing all 90 AQS behavioural descriptors. Mechanisms for encoding very simple episodic memories for anxiety episodes have been implemented. Detailed mini-simulations of AQS descriptor items 2, 13 and 33 from the 'time to recover from anxiety' subset have been completed with progress ongoing for the AQS descriptor items 34, 43, 70, 71, and 78.

Conclusion

Evaluation and validation of attachment models is less well defined than the quantitative evaluation and validation which can occur with some cognitive models that involve simulating quantitative data like reaction times or accuracy measures.

AQS number	**Descriptor**
2	When child returns to mother after playing, he is sometimes fussy for no clear reason
13	When the child is upset by mother's leaving, he continues to cry or even gets angry after she is gone.
33	Child sometimes signals mother (or gives the impression) that he wants to be put down, and then fusses or wants to be picked right back up.
34	When child is upset about mother leaving him, he sits right where he is and cries. Doesn't go after her.
43	Child stays closer to mother or returns to her more often than the simple task of keeping track of her requires.
70	Child quickly greets his mother with a big smile when she enters the room. (Shows her a toy, gestures, or says "Hi, Mommy").
71	If held in mother's arms, child stops crying and quickly recovers after being frightened or upset.
78	When something upsets the child, he stays where he is and cries.

Figure 2: A set of eight AQS descriptors related to descriptions of the time an infant takes to recover from anxiety have been grouped together to act as a starting point for the AQS modelling project. (AQS descriptor numbers relate to the ordering given in (Waters,1987)).

This paper has demonstrated the benefit of using AQS descriptors in attachment modelling because of their structural fidelity, comprehensive coverage as attachment 'ethograms', and ready incorporation in modelling scenarios. In comparison to the AQS item pool, past modelling research has focused on a narrower, and arguably less theoretically interesting range of behaviors and processes. Thus, this paper has examined the AQS item pool with an eye toward identifying content that could be incorporated into existing models and architectures. It has also highlighted content that seems too complex to be easily incorporated and suggested some of the problems that would have to be solved before doing so. New mechanisms have been described to simulate how infants retreat to the caregiver when distressed, and establish and maintain contact until comfortable enough to resume exploration. The next stage is to simulate how infants explore away from the caregiver, evaluate and maintain caregiver access and availability, and seek information or assistance while exploring or manipulating objects or locations.

References

Ainsworth, M., Blehar, M., Waters, E., & Wall, S. (1978). *Patterns of attachment: A psychological study of the strange situation*. Hillsdale, NJ: Erlbaum.

Bowlby, J. (1969). *Attachment and loss, volume 1: Attachment*. New York: Basic books.

Cassidy, J., & Shaver, P. (2016). Handbook of Attachment, 3rd Ed. London: Guilford Press.

Clark, L., & Watson, D. (1995). Constructing validity: Basic issues in objective scale development. *Psychological Assessment*, *7*, 309–319.

Edalat, A., & Mancinelli, F. (2013). Strong attractors of Hopfield neural networks to model attachment types and behavioural patterns. IEEE. Available from `http://dx.doi.org/10.1109/IJCNN.2013.6706924`

Fonagy, P. (1999). Points of contact and divergence between psychoanalytic and attachment theories: Is psychoanalytic theory truly different. *Psychoanalytic Inquiry*, *19:4*, 448-480.

Fraley, R. (2007). A connectionist approach to the organization and continuity of working models of attachment. *Personality and Social Psychology Review*, *6*, 1157–80.

Petters, D. (2006a). *Designing agents to understand infants*. Unpublished doctoral dissertation, School of Computer Science, The University of Birmingham. ((Available online at http://www.cs.bham.ac.uk/research/cogaff/))

Petters, D. (2006b). Implementing a theory of attachment: A simulation of the strange situation with autonomous agents. In *Proceedings of the seventh international conference on cognitive modelling* (pp. 226–231). Trieste: Edizioni Golardiche.

Petters, D., & Beaudoin, L. (2017). Attachment Modelling: From Observations to scenarios to designs. In P. Erdi, B. Bhattacharya, & A. Cochran (Eds.), *Computational neurology and psychiatry: Volume 6 of springer series in bio-/neuroinformatics.* (pp. 227–271).

Petters, D., & Waters, E. (2015). Modelling Emotional Attachment: An Integrative Framework for Architectures and Scenarios. In *Proceedings of IJCNN*. IEEE.

Simms, L., & Watson, D. (2010). The construct validation approach to personality scale construction. In Handbook of Research Methods in Personality Psychology, *eds. r.w. robins, r.c. fraley & r.f. krueger* (pp. 240–258). London: Guilford Press.

Waters, E. (1987). Attachment q-set (version 3) [Computer software manual]. Available from `http://www.psychology.sunysb.edu/attachment/measures/content/aqs_items.pdf` (accessed February 2, 2017)

Waters, E., & Deane, K. (1985). Defining and assessing individual differences in attachment relationships: Q-methodology and the organisation of behaviour in infancy and early childhood. *Monographs of the Society for Research in Child Development*, *50*(1-2), 41–65.

A computational model of focused attention meditation and its transfer to a sustained attention task

Amir J. Moye (amir.moye@psy.unibe.ch)
Institute of Cognitive Psychology, Perception and Research Methods, University of Bern
Fabrikstrasse 8, 3012 Bern, Switzerland

Marieke K. van Vugt (m.k.van.vugt@rug.nl)
Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen
Nijenborgh 9, 9747 AG Groningen, The Netherlands

Abstract

Although meditation and mindfulness practices are widely discussed in the scientific literature, there is little formal theory about the cognitive mechanisms that comprise it. Here we begin to develop such a theory by creating a computational cognitive model of a particular type of meditation: focused attention meditation. This model was created within *Prims*, a cognitive architecture similar to and based on *ACT-R*, which enables us to make predictions about the cognitive tasks that meditation experience may affect. We implemented a model based on an extensive literature review of how the meditation experience unfolds over time. We then subjected the Prims model to a session of the *Sustained Reaction to Response Task*, a task typically used to study sustained attention, a faculty that may be trained with meditation practice. Analyses revealed that the model was significantly more sensitive to detecting targets and non-targets after the meditation practice than before. These results agree qualitatively with empirical findings of a longitudinal study conducted in 2010. These results suggest that our approach to modeling meditation and its effects of cognition is feasible.

***Keywords*:** Focused attention meditation, mindfulness, sustained attention, SART, PRIMS, transfer

Introduction

Meditation consists of a set of mental exercises that have been developed and practiced reaching as far back as 4000 years (Riley, 2004). In the last 50 years there has been more and more interest in the effects of the various meditation styles on cognition and emotion. The spectrum of empirically examined effects has grown quite vast, with some being reasonably well-replicated and of medium to large effects while others have been inconsistent (Khoury, Sharma, Rush, & Fournier, 2015; Sedlmeier et al., 2012). However, there are no comprehensive computational frameworks of meditation and its effect on cognition (e.g., Vago & Silbersweig, 2012).

Meditation is often conceptualized as a family of attentional and emotional regulation exercises, the former being the aspect that virtually all styles share to some degree. However, it needs to be stressed that meditation techniques differ strongly. They originate from distinctive cultures and religions (Buddhism, Hinduism, Taoism, Sufism, Christian Centering Prayer, etc.) as well as secular settings (acceptance and commitment therapy, mindfulness-based stress reduction, mindfulness-based cognitive therapy; Hayes, 2004; Kabat-Zinn, 1990; Teasdale et al., 2000). They can differ greatly concerning the emphasis of the mental faculties used (attention, feeling, reasoning, visualization, etc.), the objects they are focused on (thoughts, images, concepts, internal energy, breath, love, God, etc.; Shear, 2006) and lastly with what aim they are employed (relaxation, heightened sense of well-being; attentional balance, insight, etc.; Lutz, Slagter, Dunne, & Davidson, 2008; Wallace, 1999). That being said, the common typology to categorize this vast family of practices is based on what meditators are purportedly doing from a first-person perspective: 'Focused Attention' (FA) meditation and 'Open Monitoring' (OM) meditation (Lutz et al., 2008). In OM practices – in contrast to FA meditation – there is no clear focus of attention and the task is to be continuously aware of phenomena appearing and to return to this monitoring when one gets caught up with the content.

In this paper, we begin to develop a computational theory of meditation practices by creating a cognitive model of focused attention (FA) meditation, as this kind of meditation is most amenable to computational modeling. In this practice, the meditator brings her/his attention to an object such as the breath, and then monitors with non-judgmental attention whether attention is still there. As soon as the meditator realizes attention has wandered, s/he brings the attention back to the object of focus, minimizing any further mental elaboration.

The particular type of FA meditation that was practiced by the subjects relevant for this article was so-called *Samatha* meditation (MacLean et al., 2010). According to Wallace (1999), the meditation instructor of the retreat, the main goal of this practice is to cultivate a stability and vividness concerning attention. In order to pursue this cultivation there are two crucial faculties that must be refined in turn: *mindfulness* and *introspection*, mindfulness being the primary faculty. In the setting of Samatha, mindfulness may be reduced to the aspects of recollection and steadiness: the ability to remember to sustain the attention on a given object and to remember to return when there has been a distraction nevertheless (Wallace, 1999). Introspection, on the other hand, is the faculty to monitor the meditation process, a type of meta-cognition that is tuned to the detection of increases in phenomenological excitation or laxity. When these two faculties fail, mind wandering may take over: an unintended shift of focus to a sensory or mental event, which then leads to habitual affective responding, which in turn triggers related mental events such as episodic or procedural memories, that then lead to more habitual affective responses and so on (Vago & Silbersweig, 2012).

The meditation model was constrained in two ways: (i) qualitatively through taking testimonials and existing theories on meditation into account and (ii) quantitatively by taking existing data into account. Because meditation itself produces virtually no behavioral output to which one could compare a model output, our model was constrained indirectly by having it predict transfer to a similar task that does produce output. This transfer was compared to empirical data of a three month FA meditation retreat (MacLean et al., 2010). The specific transfer was from multiple FA meditation sessions to a *Sustained Attention to Response Task* (SART; Robertson, Manly, Andrade, Baddeley, & Yiend, 1997). The similarity between the modeled and actual transfer effect is then an indirect measure for the fit of the meditation model to the actual meditation process. The rationale here is that an adequate model of meditation would be expected to make reasonably good predictions about transfer to other tasks.

The SART is a useful task to examine the effects of meditation practice, because both meditation and this task involve maintaining attention over a long period. In the SART, typically performance is quite good at first but quickly decreases. This vigilance decrement is characterized by a reduction in speed and accuracy as well as reductions in perceptual sensitivity and increases in response bias (Warm, 1980). According to Lutz et al. (2008) there are significant parallels between conceptualizations of sustained attention in cognitive sciences and processes involved in FA meditation. Moreover, there is consensus between Western scientists and Buddhist scholars that both processes require “skills involved in monitoring the focus of attention and detecting distraction, disengaging attention from the source of distraction, and (re)directing and engaging attention to the intended object” (Lutz et al., 2008, p. 2).

Computational models for the SART already exist (Gunzelmann, Gross, Gluck, & Dinges, 2009; van Vugt, Taatgen, Sackur, & Bastian, 2015). The SART model created for this paper was inspired by the model by van Vugt et al. (2015), which–contrary to other models that leave mind-wandering abstract–models mind-wandering explicitly as a cognitive process of memory retrieval. The advantage of modeling mind-wandering explicitly is that it allows you to model the actual thoughts that are mind-wandered about, and the change in attitude towards these thoughts that is so characteristic of meditation practice (Desbordes et al., 2015; Vago & Silbersweig, 2012). Even though there are several comprehensive theoretical frameworks of meditation (e.g., Vago & Silbersweig, 2012), to the best of our knowledge there are not yet any computational models of meditation, let alone FA meditation.

We implemented our model in the *Prims architecture* (Primitive Information Processing Elements; Taatgen, 2013). It is a recent extension of the well-established *Adaptive Control of Thought – Rational*, or *ACT-R* (Adaptive Control of Thought-Rational; Anderson & Lebiere, 2012) and has been developed to be able to explain transfer between different cognitive tasks, which is crucial for our project. As in ACT-R, cognitive processing is distributed across specialized modules, which are implied by some theories of cognition (Anderson & Lebiere, 2012):

- A goal module, which stores active goals and applies their influence.
- An input module, which models perception (e.g., vision)
- An output module, which model outward actions (e.g., button presses)
- A retrieval module, which models declarative memory and memory retrieval processes.
- A working memory module, which stores information that is immediately accessible and intermediate steps in calculations

Cognitive processing itself takes place in cycles of applying if-then-rules. These rules are called *operators* in Prims (and *productions* in ACT-R). In every cycle, the information in the buffers of the modules is compared to the conditions of the operators. If multiple operators have conditions that fit the information in the system, a competition between them occurs and the operator with the highest activity – which depends among other factors on a baseline activity plus a random noise variable – is chosen to be executed.

Method

When the model is run for several rounds it simulates roughly four processes that a meditator cycles through:

1. Remembering (or keeping in mind) what is supposed to be done again and again: In this case, this is the task of being aware of the breath.
2. Being aware of breath sensations, which is simulated as copying the perception into working memory.
3. Remembering something else and wandering off into daydreams, worries, etc.
4. Remembering to come back to the task when one has wandered off.

The model does this by assuming two competing goals[1] – focusing on the breath (the focus goal) and mind wandering (the wander goal) – which each have operators associated with them (van Vugt et al., 2015). Which operator wins depends on three factors in this model: the baseline activation of the operator, the random activation added and the spreading activation from the goal it is associated with. Goals can furthermore be activated or deactivated by operator actions (a unique feature of Prims that ACT-R does

[1] These goals – especially the goal to mind wander – are not necessarily explicit/conscious to the individual.

not have). In the latter case, their activation is automatically 0. This does not mean that operators associated with an inactive goal cannot win a competition; it just makes it a lot less likely.

As can be seen in Figure 1, all of the operators are triggered by the retrieval of the last cycle and as can be seen in Table 1, there are three kinds of memory chunks. 30 of them are meant to model mind-wandering contents (not necessarily single memories but rather representative instances of narratives or overarching themes). The 31st is the memory of the meta-task, which is the memory of refreshing the goal itself before checking what the low-level task at hand is. The 32nd is the memory of the low-level task, which entails feeling the breath.

Table 1: The three types of memories in the declarative memory of the meditation model and their slots.

Meta-task (n=1)	Task (n=1)	Mind-wandering (n=30)
Memory	Memory	Memory
Intention	Intention	Mind-wandering
Meta-task	Task	Memory-4*
Focus	Breath	Approach*

Note: * These are examples. The memory slot ranges from ‘Memory-1’ to ‘Memory-30’ and the valence slot can contain ‘Approach’, ‘Avoid’ or ‘Stay’.

A mind-wandering memory could have the following slot contents: Memory, Mind-wandering, Memory-17, Avoid. The first slot indicates that this chunk is a memory, which is a very general label to allow for general requests. The second slot distinguishes the mind-wandering chunks from the memories of intentions, while the third slot is a placeholder for a specific memory topic (e.g. ‘Memory-21’ might be a future-oriented and attractive topic – going on vacation). Finally, the fourth slot contains the valence or motivational connotation. Both intention memories have lower activations to begin with, 1.00 as opposed to the mind-wandering chunk’s average activation of 3.07. This models the intention memories being less salient and engaging (at first) than the mind-wandering memories.

The model starts off with the focus goal activated and ‘Breath’ in the input buffer (which remains there). As nothing has been retrieved, the retrieval operators of both goals will compete. At this point the focus operator will usually win, as the wander goal is not active yet. If it does, it requests a general memory and since it has associations with the task and meta-task memories, they have a better chance than the daydream memories of being recalled (if they have the same baseline activation anyway). If the task memory is remembered this directly triggers being aware of the breath, however if the meta-task memory is recalled this first triggers the refresh-focus-operator. This activates the focus goal if it was inactive or reinforcing it if it was already active. Next the opposing goal is deactivated if it is active and the concrete task at hand is requested, modeling a meta-cognitive process that consist of reinforcing the goal to focus and remembering the task to focus on. After feeling the breath nothing is retrieved and the retrieval operators once again are triggered. If the wander operator wins it will initiate a similar process as outlined for the focus goal, thereby reinforcing the wander goal. Once a goal has been activated its operators tends to go into a stable loop. However, as can be seen in figure 1 there are multiple interception points to interrupt this.

The model of sustained attention simulates the

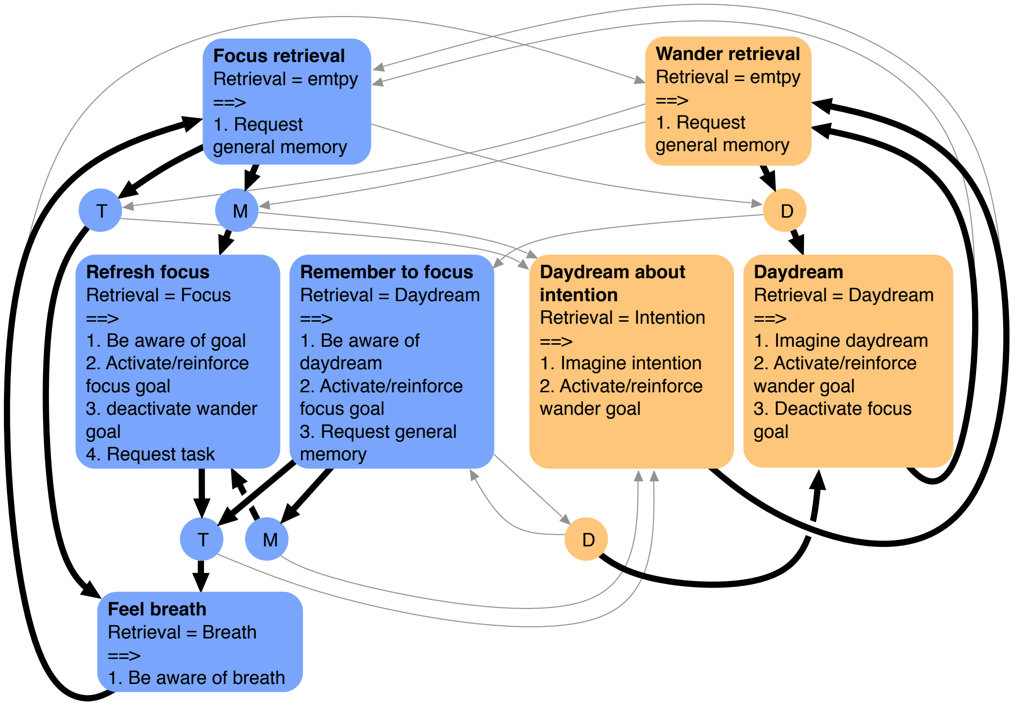


Figure 1: Meditation model. Blue objects are related to the focus goal, yellow ones are related to the wander goal. The boxes are operators, while the small circles are memories that are retrieved due to a request by an operator. ‘T’ stands for task, ‘M’ stands for meta-task, ‘D’ stands for daydream. The arrows represent possible transitions. Thick black arrows represent high probability, while thin gray arrows indicate lower probability. The represented probabilities always signify the chances if both goals were active and the memories had similar baseline activations.

performance of the meditators in a SART that the participants of the meditation retreat performed (MacLean et al., 2010). It consists of frequent non-targets (long lines, with 90% probability) and rare targets (short lines, with 10% probability). The screen switched between the display of a mask (1.55-2.15s) and the display of a stimulus (0.15s). There was a practice block of 120 trials and 4 contiguous test blocks of 120 trials each, which lasted for about 18min. The main measure was *A'* (Stanislaw & Todorov, 1999), a measure of sensitivity combining hit rates and false alarms.

The model (Figure 2) is made up of operators for modeling the mind wandering as well as operators for modeling the execution of the SART task. The operators for mind wandering are almost identical to their respective copies from the meditation model. In a sense, the model consists of SART operators (identifying the stimulus, pressing, etc.) and a modification of the meditation model missing the primary and secondary focus operators.

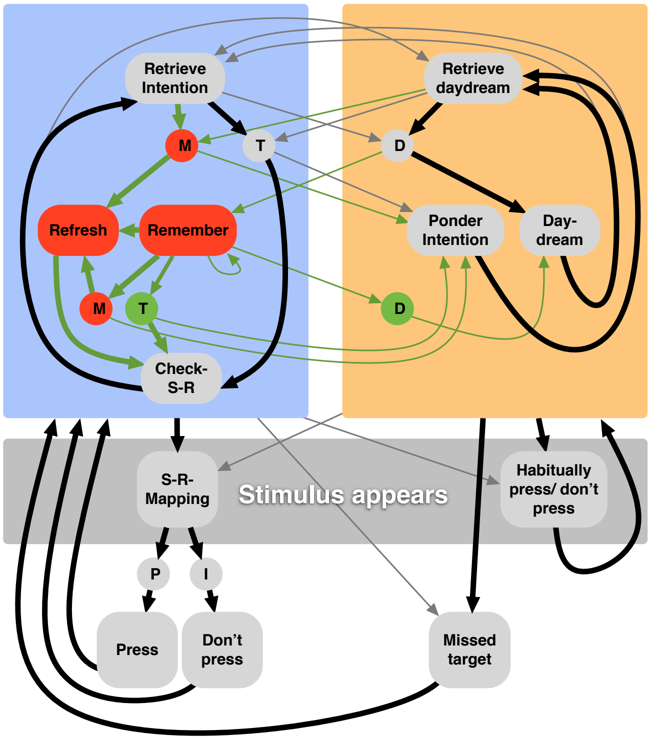


Figure 2: Model of the SART after transfer. 'T' stands for task, 'M' stands for meta-task and 'D' stands for daydream. The red objects are the transferred operators and the meta-task memory. The green memories and transition lines appear in the diagram as a consequence of this transfer.

The transfer consisted of copying the meta-task memory and two meditation model operators into the SART model, transferring the following processes: Reinforcing/activating the focus goal, deactivating the wander goal, reinforcing the focus related memories and the process of remembering the task at hand when mind wandering. Importantly, the low-level task and its memory differed from their counterparts in the meditation model: in the SART the low-level task was to check the stimulus-response-mapping in case a stimulus appeared.

Results

Prims has a vast spectrum of parameters, most of which influence the performance of the models. A majority of them were kept at the default level, while some were adjusted to allow for both models to perform at least somewhat realistically. Specifically, the activation noise was set to 0.4 (default is 0.1), which allowed for slower transitions, more interference and shorter loops. The amount of goal buffer spreading activation was set to 0.75 (default is 1), which decreases the impact the goal activation/deactivation has, with similar effects as the increased activation noise parameter. The amount of working memory buffer spreading activation was set to 0.3 (default is 0), which allows for association between daydreams during mind wandering. The latency factor was set to 0.15 (default is 0.2) to make the SART model faster in responding to the stimulus. The learning parameter for production compilation was set to 0.2 (default 0.1) to allow the SART model to assemble the prims faster in the training phase.

The meditation model was tested for a simulated 18 hours at which point it seemed to have reached a dynamic equilibrium (representing the process of learning to stay focused on the breath). The analyses reported here pertain to only one run, as there was very little variation between the runs. As can be seen in figure 3, the model starts off with a lot of mind wandering but slowly begins to shift to more focus and then drops below the rising focus percentage out at about 5 hours. In the end almost all retrieved memories are focus related.

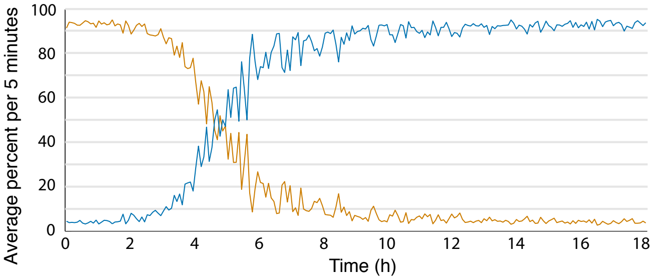


Figure 3: The average percentage (of 5-minute periods) of focus (blue) and wander (yellow) operators during a simulated 18h run.

The SART model was run for 1 training block and 4 test blocks like in the empirical study. The results presented are the average of 30 runs, as the SART model was somewhat variable in its performance, partly due to the relatively short simulated time span (18 min as opposed to 18 hours for the meditation model).

The SART model with transfer was run with a meta-task memory at the low starting activation level of the meditation model: 1.00. As can be seen in Figure 4, the mind wandering percentage is lower, while the focus percentage

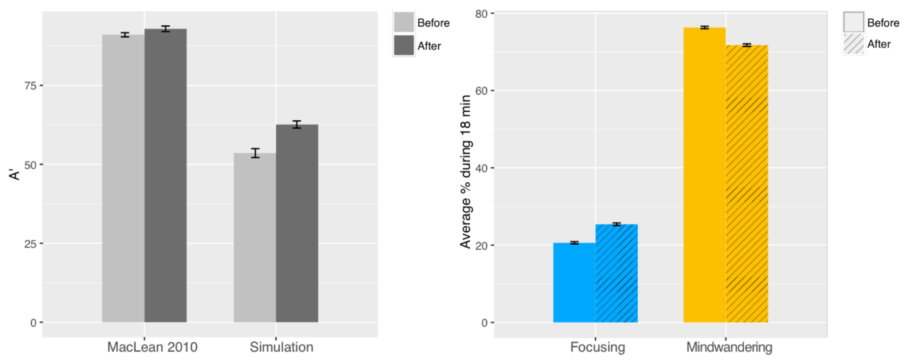


Figure 4: Measures in the SART before and after meditation training. The plot on the left compares the results found in the study by MacLean et al. (2010) before and after the retreat with the performance of the model before and after transfer of two operators and a memory at an activation level of 1.00. The gray bars represent the average sensitivity *A'*. The plot on the right compares the average percentage of focus and mind wandering operators respectively during 18min runs.

has increased. Furthermore, the hit rate was increased and the false alarm rate was lower (not displayed in the figure), leading to an increased sensitivity. An independent t-test of the mean sensitivity over time revealed that the difference was highly significant ($t(58) = 4.49$, $p < 0.001$) and that Cohen's effect size of the difference was large ($d = 1.18$). Examination of the Q-Q plot and the Shapiro-Wilk test showed no significant deviation from normality ($W(60) = 0.99$, $p = 0.85$). The difference was even more pronounced when the meta-task memory was transferred at above average activation levels (4.50): $t(47.82) = 14.05$, $p < 0.001$, $d = 3.69$. The assumption of normality was rejected ($W(60) = 0.92$, $p = 0.001$). Therefore a bootstrap test was conducted, which corroborated the significance of the effect ($p = 0.001$).

Discussion

This paper set out to explore the processes underlying FA meditation by creating a cognitive model to simulate it. To constrain the model and test its plausibility, a cognitive model of a SART was analyzed before and after the transfer of two meditation operators and an affiliated memory.

The meditation model transitions from mainly mind wandering to being almost entirely focused on the task at hand. There seem to be two main causes for this development: The increasing dominance of the meta-task memory over the task memory as well as the increasing dominance of both intention memories over the mind wandering memories. The fact that the meta-task memory becomes stronger than the task memory leads to more instances of the following sequence: focus retrieval → refresh focus → feel breath, and less of this sequence: focus retrieval → feel breath. This in turn allows for more reinforcement of the focus goal and the meta-task memory because the refresh-focus-operator involves goal management actions and imagination (strengthens the memory). The second cause–the domination of the intention memories over the mind wandering memories–leads to more of their retrieval and less retrieval of the mind wandering memories. In other words, it decreases the probability of interference by mind wandering memories and increases the probability of the intention memories (mostly the meta-task memory) interfering with the wander-retrieval.

This raises the question why the meta-task and the task memory increase in activation so dramatically over time. The intention memories probably increased because they are retrieved a lot more than any single mind-wandering memory. Even though the mind-wandering memories as a whole are retrieved a lot more frequently at first than the intentions and even though they spread the resulting reinforcement amongst each other to some degree (due to their associations), the reinforcement per single mind-wandering memory is a lot smaller than for the meta-task. What gives the mind-wandering memories the upper hand at first–their numbers–becomes a handicap as the reinforcement they receive is spread too evenly among them. This has interesting implications. It could mean that an important aspect of how FA meditation calms the mind lies in its simplicity and unidirectionality: it only focuses on a small group of memories, while mind-wandering has a broad focus. It could indicate that if the goal management strategy is such that it is sufficient for combating mind-wandering loops and interference–even if only rarely at first–it can reinforce its associated memories, causing it to be more effective in turn, which leads to more reinforcement and so on. In other words, if the goal management strategy is effective enough in the beginning (even if only barely) it can create a feedback loop. And while the mind-wandering process creates a feedback loop as well, it is less effective, presumably because the loop is a lot more dispersed.

What is interesting about mind-wandering is that it seems to creep up stealthily and is often easy to snap out of, but only for a few moments, which reflects what we think are two core factors in mind-wandering's longevity: tenacity and momentum. The meditation model explored in this paper suggests that FA meditation functions on the same principles supplemented with the benefits of unidirectionality. Yet, what this model leaves out is that mind-wandering is typically not a deliberate choice, while a main aspect of FA meditation is the conscious, voluntary and therefore effortful deciding from moment to moment. The model cannot distinguish between bringing something to mind consciously and something appearing on its own (Seli, Carriere, & Smilek, 2015).

Possibly the central question is how plausible the meditation model is. The meditation model was almost entirely constrained by internal consistency and basic assumptions about Samatha meditation, which is not a strong constraint. In order to increase the credibility of the meditation model, transfer to other tasks would be necessary. Nevertheless, the positive transfer effect of the goal management operators to the SART indicates some valuable points. It suggests that the mechanisms of the meditation model are at least somewhat generalizable and are not merely artifacts of a specific modeling situation. It furthermore indicates that the mind-wandering paradigm, which was very similar in both models, is plausible. Furthermore, the transfer was congruent with the kind of change one would predict. What is more, the meditation model is quite robust, simple and produces reasonable behavior considering its parsimony. In other words, there is reason to believe that the model captures one important aspect that might underlie FA meditation: a feedback loop effect induced by patient and deliberate application of a goal management strategy. On the other hand, it does not capture aspects of meditation that reflect cultivation of a non-judgmental attitude and transformation of mental habits.

In short, we have presented the first computational model of meditation and have shown that it makes predictions for transfer to cognitive task performance. The model suggests that the transfer consists of goal management faculties and that it enhances performance through a feedback loop mechanism.

Acknowledgments

Mind & Life visiting scholarship to Marieke van Vugt. We thank Trudy Buwalda and Stefan Huijser for their help with *Prims*.

References

Anderson, J. R., & Lebiere, C. J. (2012). *The Atomic Components of Thought*. New York, NY, US: Psychology Press.

Desbordes, G., Gard, T., Hoge, E. A., Hölzel, B. K., Kerr, C., Lazar, S. W., . . . Vago, D. R. (2015). Moving beyond mindfulness: defining equanimity as an outcome measure in meditation and contemplative research. *Mindfulness, 6*(2), 356-372.

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, 33*(5), 880-910.

Hayes, S. C. (2004). Acceptance and commitment therapy, relational frame theory, and the third wave of behavioral and cognitive therapies. *Behavior Therapy, 35*(4), 639-665.

Kabat-Zinn, J. (1990). *Full catastrophe living: Using the wisdom of your body and mind to face stress, pain, and illness*. New York, NY, US: Dell Publishing.

Khoury, B., Sharma, M., Rush, S. E., & Fournier, C. (2015). Mindfulness-based stress reduction for healthy individuals: a meta-analysis. *Journal of Psychosomatic Research, 78*(6), 519-528.

Lutz, A., Slagter, H. A., Dunne, J. D., & Davidson, R. J. (2008). Attention regulation and monitoring in meditation. *Trends in Cognitive Sciences, 12*(4), 163-169.

MacLean, K. A., Ferrer, E., Aichele, S. R., Bridwell, D. A., Zanesco, A. P., Jacobs, T. L., . . . Wallace, B. A. (2010). Intensive meditation training improves perceptual discrimination and sustained attention. *Psychological Science, 21*(6), 829-839.

Riley, D. (2004). Hatha yoga and the treatment of illness. *Alternative therapies in health and medicine, 10*(2), 20-21.

Robertson, I. H., Manly, T., Andrade, J., Baddeley, B. T., & Yiend, J. (1997). "Oops!" performance correlates of everyday attentional failures in traumatic brain injured and normal subjects. *Neuropsychologia, 35*(6), 747-758.

Sedlmeier, P., Eberth, J., Schwarz, M., Zimmermann, D., Haarig, F., Jaeger, S., & Kunze, S. (2012). The psychological effects of meditation: A meta-analysis. *Psychological Bulletin, 138*(6), 1139-1171.

Seli, P., Carriere, J. S. A., & Smilek, D. (2015). Not all mind wandering is created equal: Dissociating deliberate from spontaneous mind wandering. *Psychological Research, 79*(5), 750-758.

Shear, J. (2006). *The Experience of Meditation: Experts Introduce the Major Systems*. St. Paul, MN, US: Paragon House.

Stanislaw, H., & Todorov, N. (1999). Calculation of signal detection theory measures. *Behavior Research Methods, instruments, & computers, 31*(1), 137-149.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review, 120*(3), 439-471.

Teasdale, J. D., Segal, Z. V., Williams, J. M. G., Ridgeway, V. A., Soulsby, J. M., & Lau, M. A. (2000). Prevention of relapse/recurrence in major depression by mindfulness-based cognitive therapy. *Journal of Consulting and Clinical Psychology, 68*(4), 615-623.

Vago, D. R., & Silbersweig, D. A. (2012). Self-awareness, self-regulation, and self-transcendence (S-ART): a framework for understanding the neurobiological mechanisms of mindfulness. *Frontiers in Human Neuroscience, 6*, 1-24.

van Vugt, M. K., Taatgen, N. A., Sackur, J., & Bastian, M. (2015). Modeling mind-wandering: a tool to better understand distraction. In N. A. Taatgen, M. K. van Vugt, J. P. Borst & K. Mehlhorn (Eds.), *Proceedings of the 13th International Conference on Cognitive Modeling* (pp. 1-6). Groningen, the Netherlands: University of Groningen.

Wallace, B. A. (1999). The Buddhist tradition of Samatha: Methods for refining and examining consciousness. *Journal of Consciousness Studies, 6*(2-3), 175-187.

Warm, J. S. (1980). The psychophysics of vigilance *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 24, pp. 605-605): Sage Publications.

Building an ACT-R reader for eye-tracking corpus data

Jakub Dotlačil (jdotlacil@gmail.com)
Institute for Logic, Language and Computation, University of Amsterdam
Amsterdam, the Netherlands

Abstract

ACT-R has been successfully used in psycholinguistics to model processing data of individual experiments. In this paper, I show how it could be scaled up to model a much larger set of data, eye-tracking corpus data. It is shown that the resulting model has a good fit to the data for the considered (low-level) processes. The paper also argues that free parameters of ACT-R could and should be estimated using the well-established methods in other fields, rather than by manually searching through parameter space. The latter option is simply impossible to use once we hit the amount of data considered here. The latter option also makes it hard, if not impossible, to compare parameters across different (ACT-R) models since manual search is subjective and usually not well documented in research papers.

Keywords: parsing; eye tracking; modeling eye tracking; ACT-R; modeling eye-tracking corpus data; Bayesian inference of ACT-R parameters

Introduction

ACT-R (Adaptive Control of Thought–Rational) (see (Anderson, Bothell, & Byrne, 2004) for an introduction) is a cognitive architecture that has been successfully applied to various language processing phenomena, for example, syntactic parsing, memory retrieval of arguments and quantifiers, syntactic priming or the reanalysis of syntactic structures (Lewis & Vasishth, 2005; Vasishth, Bruüssow, Lewis, & Drenhaus, 2008; Reitter, Keller, & Moore, 2011, a.o.). The successes of ACT-R in modeling natural language strongly suggests that the cognitive architecture can be insightful for linguistics, alongside many other domains of inquiry (cf. (Anderson, 2007)).

Previous applications of ACT-R focused on modeling of (some) results of carefully chosen experiments. This leaves open the question as to how ACT-R fares once we move beyond such a domain. If ACT-R is to be useful for language modeling it should be shown that it can scale up, that is, it can fare well when modeling a large amount of processing data (cf. (Taatgen & Anderson, 2002) for such a large scale ACT-R application in a different psycholinguistic domain). Furthermore, it is important to see how it fares when modeling data that are naturally occurring, not carefully composed by experimentalists to target one phenomenon. Second, previous models were hand-crafted to match analyzed phenomena. This can be seen in two ways: (i) grammar rules are not created automatically, rather, they are manually written, (one exception here being (Reitter et al., 2011)) (ii) parameters used in the sub-symbolic part of ACT-R are plugged in by modelers.

In this paper, I will focus on the second issue: the manual search of parameters. The problem with that is that it makes model fitting subjective. As a consequence, it is very hard if not impossible to compare various models. For example, (Vasishth et al., 2008) differ from (Lewis & Vasishth, 2005) in the values they assume for the latency factor (0.46 vs. 0.14). The model in (Reitter et al., 2011) differs from both papers in its assumption about the value of the maximum associative strength (50.0 in the latter vs 1.5 in the former papers). It is not clear whether these differences are meaningful or accidental. We do not know how good the model fit would be if the values of these parameters were matched. We also do not know what values were considered before settling on these. Finally, we also do not know whether other parameters were searched before these were modified. All these concerns make it hard, if not impossible, to consider model comparisons. Maybe even more importantly, selecting the values of parameters by hand is almost impossible once we scale up and model more data, especially if we want to fit more than one parameter.

In this paper, I take first steps to address the worries discussed above. Further improvements should follow in the future. First, I consider the application of an ACT-R parsing model to eye-tracking *corpus* data (the GECO corpus, (Cop, Dirix, Drieghe, & Duyck, 2016)). Second, I show how the model can be fitted using Markov-Chain Monte Carlo (MCMC) methods, rather than a manual selection of parameters. Importantly, using MCMC methods makes it easy to compare the parameters of the current model to other models. As an example, I make one such comparison, which will reveal a match between some (but not other) parameters, potentially opening a window into more detailed research into the role of ACT-R free parameters across models.

Modeled data

The paper presents a model of (a subset of) reading measures of the Ghent Eye-Tracking Corpus, GECO (Cop et al., 2016). The corpus consists of eye movement measures collected during reading of the book *The Mysterious Affair at Styles* by Agatha Christie. The data were collected from 14 English monolingual readers and 19 Dutch-English bilingual readers. For the current purposes, we are not interested in the effect of bilingualism and thus, only monolingual data will be studied.

A desirable feature of the GECO is that the whole corpus is freely downloadable and its text is in the public domain. Furthermore, the fact that readers read an entire book, rather than the collection of random articles/sentences might potentially be useful in the future if we want to model long-lasting effects (e.g., discourse structures). However, this will not be attempted here. For the details of the corpus and its comparison to other eye-tracking corpora, see (Cop et al., 2016).

I am afraid I showed my surprise
rather plainly.

Figure 1: Example of a parsed sentence.

$${}_{\text{AFRAID}}\begin{bmatrix}\text{ISA:} & \text{word}\\ \text{FORM:} & \text{afraid}\\ \text{CATEGORY:} & \text{adjective}\end{bmatrix}$$

Figure 2: Example of the chunk AFRAID.

Basic ACT-R reader

The reader considered in this paper is very basic. It serves as the starting point and it can be further expanded.

The reader will visually encode and retrieve words from English sentences of the GECO corpus, an example of which is in Fig. 1 (the figure encodes the original line breaks).

The reader starts at the first word of the sentence. It stores the word in its visual buffer and retrieves information about the word from its mental lexicon. Once retrieved, the reader shifts its focus to the next word of the sentence, repeating the process. When getting to the end of the line (the word *surprise*), the reader shifts its visual focus to the beginning of the next line and proceed in reading. After the last word of the sentence, the first word of the next sentence will be parsed.

Obviously, the reader in its current form is primitive. It models only visual processes present in reading and processes tied to lexical retrieval. This limitation is intentional. It is important to show that even such primitive models are tangible and useful in modeling eye-tracking corpus data. Once the model is in place, we can move to more complex cases.

Details of the model

Symbolic part As is well-known, ACT-R subsumes two types of knowledge: declarative knowledge and procedural knowledge (cf. (Newell, 1990) on the difference). While the declarative knowledge represents our knowledge of facts, procedural knowledge is knowledge that we display in our behavior (cf. (Newell, 1973)). Following all previous works on ACT-R processing I will assume that lexical information is part of our declarative knowledge. In contrast to that, reading itself is part of our procedural knowledge. The reading consists of finding a word, retrieving the information about the word from the declarative memory and moving one's attention from word to word (in the left-to-right, top-to-bottom fashion).

The declarative knowledge is instantiated in chunks. The procedural knowledge is instantiated in production rules (productions for short).

The chunks storing lexical knowledge can be kept simple, given the basic aims of the presented ACT-R reader: they only store the information about the form and its category, see Fig. 2.

The procedural knowledge consists only of a handful of rules, shown in Fig. 3 to Fig. 6.

```
=g >
state               start
=visual_location >
?visual >
state               free
buffer              empty
==>
=g >
state               retrieve
+visual >
cmd                 move_attention
screen_pos          =visual_location
```

Figure 3: Rule ATTEND WORD.

```
=g >
state           retrieve
=visual >
value           =val
?retrieval >
state           free
==>
=g >
state           shift
word            =val
−visual >
+retrieval >
form            =val
```

Figure 4: Rule RETRIEVE WORD.

The first rule (Fig. 3) attends the currently considered word. The second rule retrieves a word from the declarative memory. The third and the fourth rule (Fig. 5 and Fig. 6) shift attention to a new word in the same line and to a new word on a new line respectively. The first rule mimics the left-to-right reading due to the interplay of two requirements: (i) it is required that the new word should have the lowest x-value on the same line as the current word, (ii) at the same time, it is required that the word should not have been attended previously (by setting :attended as false). This leaves the closest word to the right as the only candidate. The jump to the left-most word in the closest lower line is achieved in a parallel way in the second rule.

One thing to notice in both rules is the value LASTWORD. This value is not specified here further, but in the actual model it would carry the position of the rightmost words on the screen, allowing the ACT-R model to shift to a new line only after the reader got to the end of the line.

As is standard, it was assumed that every rule needs 50 ms to fire.

Subsymbolic part The subsymbolic part of the ACT-R cognitive architecture is used to match human performance. Basic ACT-R reader will model eye fixations of GECO as the function of word length, frequency of the word and word

```
=g >
state               shift
=retrieval >
cat                 =x
=visual_location >
screen_y            =ypos
screen_x            −LASTWORD
==>
=g >
state               start
+visual_location >
:attended           False
screen_x            lowest
screen_y            =ypos
−retrieval >
```

Figure 5: Rule MOVE ATTENTION IN LINE.

```
=g >
state               shift
=retrieval >
cat                 =x
=visual_location >
screen_y            =ypos
screen_x            LASTWORD
==>
=g >
state               start
+visual_location >
:attended           False
screen_x            lowest
screen_y            closest
−retrieval >
```

Figure 6: Rule MOVE ATTENTION TO A NEW LINE.

position. For this reason, only two parts of the cognitive architecture will be relevant: vision module and the module of declarative memory. The rest of this section summarizes the relevant properties of these modules.

ACT-R can be used with various implementations of vision. Here, we will consider an ACT-R implementation of the EMMA (Eye Movements and Movement of Attention) model (Salvucci, 2001), which in turn is a generalization (and a simplification) of the E-Z Reader model (Reichle, Pollatsek, Fisher, & Rayner, 1998). While the latter model is used for reading, the goal of EMMA is to model any visual task, not just reading. Given the fact that the E-Z Reader model is one of the most successful models for eye-tracking data, it is natural to use its ACT-R application, EMMA, for the current purposes (see also (Engelmann, Vasishth, Engbert, & Kliegl, 2013) for another application in psycholinguistics).

Following E-Z Reader, EMMA disassociates eye focus and attention: the two processes are related but not identical.

A shift of attention to a visual object triggers (i) an immediate attempt to encode the object as an internal representation, and (ii) eye movement.

The encoding takes the time shown in Eq 1.[1]

$$T_{enc} = K \cdot D \cdot e^{kd} \tag{1}$$

In the equation, d is the distance between the current focal point of the eyes and the object to be encoded measured in degrees of visual angle (in other words, d is the eccentricity of the object relative to the current eye position), k is a free parameter, scaling the effect of distance; D is a time parameter of the object to be focused that will affect visual encoding, and K is a free parameter, scaling the encoding time itself.

In (Salvucci, 2001), it is assumed that D is a function of the (normalized) frequency of the object, $D = -log(\text{Freq})$. This assumption is present to capture the fact that high-frequent objects (words, numbers) tend to be focused shorter and skipped more often than low-frequent objects. The same effect is encoded in the E-Z reader, in which encoding time is scaled by the frequency of the object.

There is a less stipulative way to capture the effect of frequency in Basic ACT-R Reader. Objects (words) have to be retrieved from declarative memory during reading and the retrieval itself is sensitive to frequency effects. The way our symbolic system is set up will then derive the observed role of frequency on fixations and skipping indirectly and by a mechanism that is needed anyway, lexical retrieval, as we will see below. This frees Eq 1 from an extra stipulated parameter, frequency of objects. Instead of frequency, we can therefore consider other properties relevant for visual encoding. As is well-established, the length of words affects fixations and it is natural to assume that such a property would play a role when encoding an object (but not during lexical retrieval). I will assume that D is equivalent to the number of characters of a word, see Eq 2.

$$D = \text{NChar}(\text{Word}) \tag{2}$$

The time needed for eyes to move to a new object is split into two sub-processes in EMMA: preparation and execution. The preparation requires 135 ms. The execution, which follows the preparation, requires 70 ms + 2 ms for every degree of visual angle between the current eye position and the targeted visual object.[2] At the end of the execution eyes focus on the new position. If a new command to shift an attention yet again is issued during the preparation phase, the old eye movement is discarded and a new one takes place. This situation could be used to model word skipping. For more details on the interplay between attention shift and eye movements, see (Salvucci, 2001).

[1] The equation captures the time needed to encode an object if we do not assume any noise in the vision module. Otherwise, the encoding of an object is modeled using a gamma distribution with the mean T_{enc} and sd $\frac{T_{enc}}{3}$.

[2] If eye movement is assumed to be noisy, both measures are means of a gamma distribution, see the previous footnote.

The second part of the subsymbolic system important for us concerns lexical retrieval.

Simplifying somewhat and focusing only on currently relevant parameters, we can say that the time needed to retrieve a word is a function of its base-level activation. In more technical terms, we will assume that the activation of a chunk i, A_i, determining retrieval latencies, is equivalent to its base-level activation, B_i (normally, chunk activation is modulated by other chunk properties, and is distributed as Logistic(B_i, s) with s being a free parameter):

$$A_i = B_i \tag{3}$$

The base activation of a chunk in ACT-R, B_i, is in Eq 4, where d is a free parameter and t_k is the time elapsed since the chunk was presented (stored in memory).

$$B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right) \tag{4}$$

The time needed to retrieve a chunk, T_i is shown in Eq 5. f is a free parameter, scaling the effect of the (base) activation, F is a free parameter, scaling the latency itself.[3]

$$T_i = F \cdot e^{-f \cdot A_i} \tag{5}$$

Summing up, fixation times will be affected in several ways in our model:

- The frequency of words will modulate fixation times, due to Eq 5, which becomes relevant when the rule RETRIEVE WORD (Fig. 4) fires. Frequencies will affect retrieval latencies because they affect the number and moments of chunk presentations. How frequencies are related to the number and moments of chunk presentations will not be discussed here in detail due to the lack of space. See (Reitter et al., 2011) for details, which I follow in this respect.

- The length of words will modulate fixation times, due to Eq 1 and Eq 2. These equations are relevant when the rule ATTEND WORD fires, Fig. 3. Furthermore, the length of words also influence fixation times in a less direct way. Assuming that fixations always appear at the center of a word, a word of length, say, 6 letters will make the words to the left and right appear one letter further than a word of length 4 letters. Due to the fact that executing eye movement is sensitive to distance, we should see an increase of fixation times on long words and on words preceding long words.

- Words appearing at the end of line or close to the end of line should be fixated longer. This is due to the execution time of eye movement: executing eye movement to a new line should take more time than executing eye movement to a new word on the same line.

Modeling reading

Eye-tracking reading measures are commonly split into several subtypes. The three most important ones are listed below:

- **gaze duration**: the sum of the time of all the first-pass fixations (in ms) made on a word until the point of fixation leaves the word

- **total reading time**: the sum of the time of all the fixations made on a word

- **re-reading time**: the difference between total reading time and gaze duration

The paper aims to model the effect of frequency and word properties (position, length). Such properties are standardly associated with first-pass measures. This is in fact directly encoded in E-Z Reader in which (modeled) gaze durations are functions of such factors, while re-reading measures less so (see, e.g., (Staub, 2011) for discussion and empirical evidence). Following this insight, I will focus on modeling gaze durations.

The GECO corpus stores the information about the position of each word on the screen. This enables us to fully reconstruct what each participant saw. Using this information, I re-created the reading materials of GECO.[4] I let Basic ACT-R Reader run and recorded its fixation times on every word (the value was 0 if a word was skipped). On one third of the materials, Basic ACT-R reader was run in order to find good estimates for some of its free parameters (more on this below). On one half of the materials, the model with the found parameters was studied. (The last sixth of materials was left out for possible future model comparisons.)

In the previous section, we saw five free parameters. Of these, only three were estimated: k, see Eq 1, f, see Eq 5, and F, see Eq 5. I did not model K since it would strongly correlate with F and the latter parameter might be sufficient, at least at this point (frequencies correlate with length in the data set, $r = -0.37, p < .001$). The d parameter (Eq 4) was not estimated either. Rather, its default value was used (0.5) since that is the standard and extremely common practice in ACT-R research.

As was mentioned in the introduction, parameter estimation is often done by hand in ACT-R. However, that is almost impossible to do with the amount of data that we analyze here, especially if we consider more than one parameter, as is the case here. Rather than manually finding parameter values, they were estimated using Bayesian inference and MCMC procedures. I used the Python implementation of ACT-R called PYACTR (see https://github.com/jakdot/pyactr).

[3] In ACT-R literature, f is not always mentioned or used. However, see (Anderson & Lebiere, 1998). The parameter will be important for our purposes.

[4] The materials were also cleaned and prepared for modeling. Two most important changes: frequencies from the British National Corpus based on (Leech, Rayson, et al., 2014) were added; some of the sentences had two words recorded as one if they were separated by three dots (…) – such sentences were excluded for two reasons. First, they would complicate the ACT-R model. Second, GECO only reports one reading measure for them and it is not clear how fixations are distributed across the two words.

(This Python implementation yields the same reaction time values for the considered parameters as the canonical implementation in Lisp.) The parameter estimation was done using the Python package for Bayesian modeling PYMC3. The Bayesian model was specified as in Eq 6. GD is the dependent variable gaze duration (in ms), Basic ACT-R(f, F, k) is a deterministic function that yields gaze duration per word by supplying Basic ACT-R Reader with the values of the three free parameters and letting the ACT-R model run. HALFNORMAL is a folded normal distribution, GAMMA is a gamma distribution, UNIFORM a uniform distribution.[5]

$$\begin{aligned} f &\sim \textsc{HalfNormal}(\mu = 0, sd = 0.5) \\ F &\sim \textsc{Gamma}(\alpha = 2, \beta = 6) \\ k &\sim \textsc{HalfNormal}(\mu = 0, sd = 0.7) \\ \alpha &\sim \textsc{Uniform}(0, 200) \\ \sigma &\sim \textsc{HalfNormal}(\mu = 0, sd = 10) \\ \mathrm{GD} &\sim \textsc{Normal}(\alpha + \text{Basic ACT-R}(f, F, k), \sigma) \end{aligned} \tag{6}$$

Notice that when retrieval and time needed to encode a word is (hypothetically) at 0 Basic ACT-R(f, F, k) should correspond only to the time needed to fire the relevant production rules. However, our current production rules are over-simplifying reading (e.g., there is no role for syntax or semantics) and thus, it is likely that they underestimate this value. This is why another parameter was added, α, and its prior was set as a non-negative value, ranging between 0 and 200 ms.

The parameters were sampled using the Metropolis sampler, with 400 steps, first 30 steps discarded and values initialized at maximum a posteriori point estimates.[6] The posterior results:

$$\begin{aligned} f - \text{mean} : 0.15; sd : 0.09 \\ F - \text{mean} : 0.0001; sd : 0.0001 \\ k - \text{mean} : 0.61; sd : 0.04 \\ \alpha - \text{mean} : 27.8; sd : 0.5 \end{aligned} \tag{7}$$

Notice that the found values f, F, k differ from the default values, which are set at 1.0. However, the default values of the last two parameters are often changed (e.g., F appears to carry the values between 0.1 and 0.4 in psycholinguistics, and k is set at 0.4 in (Salvucci, 2001)). Still, such changes do not match our found values. Unfortunately, as far as I know, previous (psycholinguistic) studies did not make systematic well-documented investigations of parameter estimates, and thus, it is completely unclear whether the differences reveal any significant discrepancies or are just accidental. The current paper is a step forward in this regard. We need to investigate free parameters of cognitive architectures in a replicable, methodical and objective way, otherwise model comparisons become impossible.

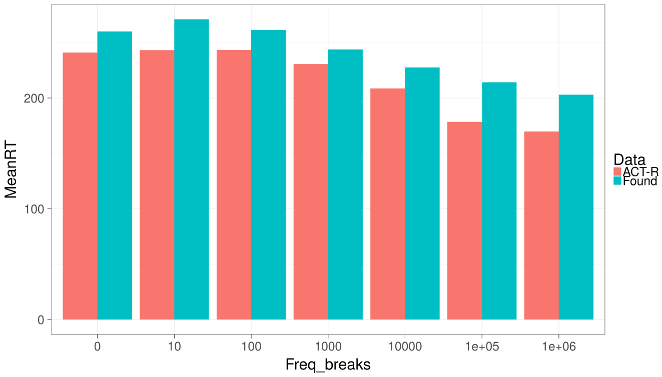


Figure 7: SimRTs and gaze durations split by word frequencies.

The mean values were plugged back into Basic ACT-R Reader. The model then simulated the reading of one half of all the sentences appearing in the GECO corpus (different sentences than the ones used in the parameter estimation). The simulated reading times (SimRTs) were used as predictors in a linear model, with mean GD (averaged across participants) as the dependent variable. The model revealed a significant effect of SimRTs ($\beta = 1.08, t = 470, p < .001$). Notice that the slope parameter β close to 1 shows that not only does Basic ACT-R Reader predict gaze durations, it does so in a way we want it to: 1 ms increase on the side of Basic ACT-R Reader corresponds to approximately 1 ms increase in actual gaze duration. The validity of the model can be also seen in Fig. 7, which plots RTs in seven frequency bands: from 0 to 10 occurrences in the BNC, from 10 to 100 etc. In each band, the red (left) bar shows mean fixation times as simulated by Basic ACT-R Reader. The right (blue) bar shows actual mean fixation times. The ACT-R model underestimates (roughly by 20 ms, which corresponds to the α estimate above) but it linearly decreases across frequency bands, closely copying the actual data. This is an encouraging finding given that the parameters were not estimated on this set of data. Fig. 8 shows that the model simulates the effect of word length well, even though it underestimates very short words, and overestimates very long words.

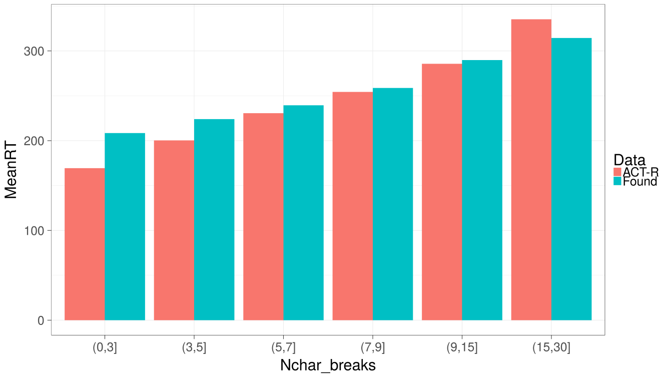


Figure 8: SimRTs and gaze durations split by word length.

[5] When estimated, the ACT-R parameters are commonly below 0.5. I tried to reflect this by selecting prior distributions whose c.d.f at 0.5 is greater than 0.5 and have positive skew.

[6] This is a small number of steps, mainly for practical reasons: the model is slow since it has to run simulations for every word of every sentence. However, the probabilistic model is simple and the found values generate good predictions.

An interesting question is whether the estimates of the model can be independently validated, using the same technique as above. For this reason, I used ACT-R to model a different psycholinguistic task, a lexical decision task of (Murray & Forster, 2004) (their Experiment 1). In the task, the ACT-R model (and humans) fixated the center of the screen. At that position a sequence of 5-7 letters appeared. The model (or human) then had to decide whether the sequence is an actual English word and press the corresponding key. The only manipulation relevant in the modeled experiment was that of the frequency of the appearing word. Thus, only two parameters were estimated using the data: f and F.

It is known that ACT-R is good at modeling the role of frequency in lexical decision tasks (cf. (Anderson, 1982), (Anderson, Fincham, & Douglass, 1999), (Murray & Forster, 2004)). Thus, estimates found this way might significantly strengthen our previous findings. Interestingly, f was estimated at 0.14 ($sd : 0.01$), thus being very close to the previously found estimate. F, in contrast, was estimated at 0.13. The difference from the previously estimated F is large, see Eq 7. In other words, while the estimated f might be close to its real value, the value of F fluctuates too wildly to be taken seriously. It remains to be seen whether it might help to model more parameters, add more information to the models or modify some other properties of the models.

Conclusion

ACT-R has been successfully used in psycholinguistics to model processing data. In this paper, I showed how it could be further expanded to model eye-tracking corpus data. The resulting model had a good fit to the corpus data, at least in the considered (low-level) processes.

Furthermore, I showed that free parameters could and should be estimated using the well-established methods in other fields, rather than by a manual search through parameter space. The latter option is impossible to use once we hit the amount of data considered here. The latter option also makes it hard, if not impossible, to compare parameters across different models since manual search is subjective and usually not well documented in research papers.

The resulting ACT-R model is a step in the direction of using ACT-R to simulate not just results of individual processing experiments, but diverse and rich corpus data. The model could be expanded to capture higher level processes (e.g., syntactic parsing). However, that is beyond the scope of this paper.

Acknowledgments

I thank SURFsara (www.surfsara.nl) for the support in using the Lisa Compute Cluster. The research presented in this paper was supported by the NWO VENI grant 275-80-005.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological review*, *89*(4), 369.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.

Anderson, J. R., Bothell, D., & Byrne, M. D. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060.

Anderson, J. R., Fincham, J. M., & Douglass, S. (1999). Practice and retention: a unifying analysis. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(5), 1120–1136.

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

Cop, U., Dirix, N., Drieghe, D., & Duyck, W. (2016). Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior research methods*, 1–14.

Engelmann, F., Vasishth, S., Engbert, R., & Kliegl, R. (2013). A framework for modeling the interaction of syntactic processing and eye movement control. *Topics in cognitive science*, *5*(3), 452–474.

Leech, G., Rayson, P., et al. (2014). *Word frequencies in written and spoken english: Based on the british national corpus*. Routledge.

Lewis, R., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 1-45.

Murray, W. S., & Forster, K. I. (2004). Serial mechanisms in lexical access: the rank hypothesis. *Psychological Review*, *111*(3), 721.

Newell, A. (1973). Production systems: Models of control structures. In W. Chase et al. (Eds.), *Visual information processing* (pp. 463–526). New York: Academic Press.

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

Reichle, E. D., Pollatsek, A., Fisher, D. L., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological review*, *105*(1), 125.

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive science*, *35*(4), 587–637.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*(4), 201–220.

Staub, A. (2011). Word recognition and syntactic attachment in reading: Evidence for a staged architecture. *Journal of Experimental Psychology: General*, *140*, 407–433.

Taatgen, N. A., & Anderson, J. R. (2002). Why do children learn to say "broke"? a model of learning the past tense without feedback. *Cognition*, *86*(2), 123–155.

Vasishth, S., Bruüssow, S., Lewis, R. L., & Drenhaus, H. (2008). Processing polarity: How the ungrammatical intrudes on the grammatical. *Cognitive Science*, *32*, 685–712.

Analysis of a Common Neural Component for Finger Gnosis and Magnitude Comparison

Terrence C. Stewart (tcstewar@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo,
200 University Avenue W, Waterloo, ON, N2L 3G1, Canada

Marcie Penner-Wilger (mpennerw@uwo.ca)
Department of Psychology, King's University College at Western University
266 Epworth Avenue, London, ON, Canada N6A 2M3

Abstract

We recently developed a spiking neuron model that performs magnitude comparison and finger gnosis tasks using a common underlying neural system, explaining why performance on these tasks is associated in humans. Here, we explore the parameters in the model that may vary across individuals, generating predictions of error patterns across the two tasks. Furthermore, we also examine the neural representation of numbers in the magnitude comparison task. Surprisingly, we find that the model fits human performance only when the neural representations for each number are *not* related to each other. That is, the representation for TWO is no more similar to THREE than it is to NINE.

Keywords: magnitude comparison; finger gnosis; neural engineering; number representation

Introduction

We have recently proposed a neural model of a cognitive component underlying two disparate tasks: finger gnosis and magnitude comparison (Stewart et al., 2017). These tasks have been shown to be related via behavioural, fMRI imaging, and stimulation experiments, and our model describes a neural system that could be involved in both tasks, explaining this relation. However, in the initial paper, we did not perform an analysis of the effects of parameter variation on this model. Our goal in this paper is to present this parameter analysis in order to better understand the performance of this model.

Finger gnosis is the ability to indicate which fingers have been touched, out of the view of the participant. Typically two fingers (on the same hand) are touched while the participant's hand is occluded, and they must then indicate which fingers were touched (Baron, 2004).

The *magnitude comparison* task considered here is symbolic single-digit number comparison. Participants are visually shown two single-digit numbers and they are asked to indicate which one is larger.

Individual performance on the finger gnosis task predicts a variety of mathematical measures in both children (Fayol et al., 1998; Noel, 2005; Penner-Wilger et al., 2007, 2009) and in adults (Penner-Wilger et al., 2014, 2015). In particular, this relation is partially mediated by performance on the single-digit symbolic magnitude comparison task used here (Penner-Wilger el al., 2009, in prep.). Individuals who perform better at magnitude comparison also perform better at the finger gnosis task.

In addition to this behavioural result, representation of number and finger gnosis both activate the same brain regions (Andres, Michaux & Pesenti, 2012; Dehaene et al., 1996; Zago et al., 2001), both tasks are disrupted by rTMS and direct cortical stimulation to the same regions (Rusconi, Walsh, & Butterworth, 2005; Roux et al., 2003), and the tasks interfere with each other when performed at the same time (Brozzoli et al., 2008). For these reasons, we believe that there is a common component underlying these tasks. In other words, there is some set of neurons performing some operation that is used in each task. This makes it an example of neural *redeployment* (Penner-Wilger & Anderson, 2008, 2013)

In the current paper, we first outline in more detail a model that performs finger gnosis and number comparison, which we initially reported in Stewart et al. (2017). Second, we examine the behavioural effects in these two tasks, as different parameters in the neural model are varied. Given that the same neural components are used, changing an aspect of the model will affect both tasks. The results of these variations form a set of predictions about individual differences in performance on these tasks. Finally, we examine a parameter that only exists for the magnitude comparison task. Here, we need to decide how the different numbers are represented neurally. One possibility is to assume that the number SEVEN should have a neural representation that is more similar to the neural representation for EIGHT than it is to ONE, as this might explain why more mistakes are made when the numbers being compared are close to each other. As is shown below, the modelling result instead shows that there is a better match to human performance if the neural representation for each number is unrelated to the others.

A Common Component

We postulate that the shared system for these two tasks is a neural implementation of *an array of pointers*. That is, a neural system that can store a small set of arbitrary values, each of which can represent something. For example, one pointer could be set to the neural representation of the number SEVEN, while another pointer could be set to represent concepts like DOG or CAT or BLUE or QUIET or TOUCHED.

Importantly, we do not need to make a strong claim about the nature of the neural representation of these concepts here. Instead, we merely make the weak claim that there is some pattern of neural activity for each concept, and all we need is a system that can store an arbitrary pattern. Here we generally randomly choose these patterns, but we investigate what happens when these patterns are related to each other below.

To describe this system mathematically, we can say that there are a small set of vectors p_1, p_2, p_3, p_4, and p_5, one for each pointer. In order for this system to maintain information over time, if there is no input, these values should stay as they are. However, if there is an input, we also need to indicate which pointer(s) will be changed. To do this, we introduce a mask *m*, which controls which pointers will be affected by the input *x*. For example, if *m*=[0,1,0,0,0], then the second pointer p_2 will be affected by the input *x*. Mathematically, this can be written as:

$$p_i \Leftarrow \begin{cases} p_i & \text{if } m_i = 0 \\ x & \text{if } m_i = 1 \end{cases} \qquad \text{(Eq. 1)}$$

For the magnitude comparison task, this would be used as follows. First, the vector for one of the numbers (e.g. SEVEN) would be loaded into the first pointer value by setting *x* to the vector for SEVEN and setting m to [1,0,0,0,0]. Next the other number (e.g. THREE) would be loaded into the second pointer by setting *x* to THREE and m to [0,1,0,0,0]. Over time, the values stored in the pointers would be as follows:

Magnitude Comparison Task

step	*x*	*m*	p_1	p_2	p_3	p_4	p_5
1	--	00000	--	--	--	--	--
2	SEVEN	10000	SEVEN	--	--	--	--
3	THREE	01000	SEVEN	THREE	--	--	--
4	--	00000	SEVEN	THREE	--	--	--

Once these values are stored in the pointers, the rest of the task can be completed by reading the values out and performing the comparison. The details for this are provided below.

For the finger gnosis task, a similar process is followed, but we use the pointers in a different way. In particular, the value that is being loaded in is always the vector for TOUCHED (indicating that this finger was touched), but the particular pointer that we load it into is what is important. In the following chart, we show the process when the second and fourth fingers are touched.

Finger Gnosis Task

step	*x*	*m*	p_1	p_2	p_3	p_4	p_5
1	--	00000	--	--	--	--	--
2	TOUCHED	01000	--	TOUCHED	--	--	--
3	TOUCHED	00010	--	TOUCHED	--	TOUCHED	--
4	--	00000	--	TOUCHED	--	TOUCHED	--

Once these values are loaded in, the rest of the finger gnosis task involves reading out these values and reporting them, as is detailed below. Importantly, while the remainder of the finger gnosis task is quite different from the magnitude comparison task, both tasks make use of this same array of pointers component.

Neural Implementation

Though this basic idea of an array of pointers is simple (and, indeed, is trivial to implement in a traditional computational model), here we implement this system using spiking neurons. The important point here is that neurons will not perfectly implement this algorithm; rather, their actual behaviour will only approximate this ideal. Importantly, this approximation can serve as an explanation for the mistakes made by people performing these tasks. Furthermore, changing the details of this neural implementation (for example, how many neurons are used, or how strong the mask is) can change the resulting behaviour, providing an explanation for the individual differences, and how errors on one task relate to errors on the other task.

To convert this model to spiking neurons, we use the Neural Engineering Framework (Eliasmith & Anderson, 2003). In this approach, different groups of neurons are used to represent each vector (e.g. *x* or p_1). Connections between groups of neurons implement functions on those variables. For example, if one group of neurons represents *x* and another group of neurons represents *y*, then we can form a connection from *x* to *y* such that *y*=*f*(*x*). Given any particular function *f*, we can solve for the optimal synaptic connection weights between those groups of neurons that will best approximate that function.

When we solve for these synaptic connection weights, we are not making any claim about how these connection weights are learned, or how they are formed in a developmental process. Rather, we are simply finding the best possible way that the given neurons can perform this task, and leaving these larger developmental questions to future research.

With this in mind, our model is presented in Figure 1. Each box represents a group of neurons representing one vector. Arrows between boxes indicate connections between groups of neurons. In each case, these connections are optimized to compute the *identity function*. This is the simple function that just transmits information without changing it in any way.

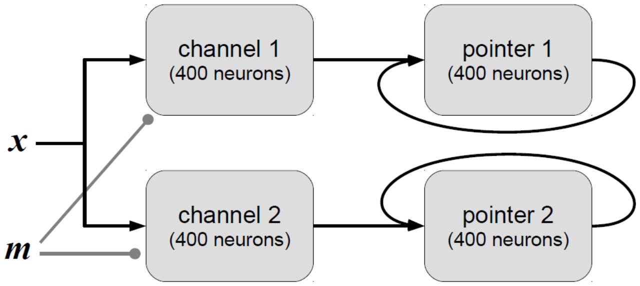


Figure 1: A neural implementation of an array of pointers. Only two pointers are shown.

The recurrent connection on the pointer neurons indicates that those 400 neurons are recurrently connected to themselves such that they will pass their own information back to themselves. In other words, whatever pattern of neural activity is generated in that group *will be self-sustaining*. That is, the pattern of activity will be maintained over time. Of course, since neurons are not perfect, this pattern will not be perfectly maintained, leading to a gradual decay of this memory system.

To load values into this system, we place the desired vector as the input *x*. This will drive the various *channel* neural populations to fire, representing that value *x*. This will then in turn drive the pointer populations to store that value. However, with just this system, any *x* value as input would be loaded into *all* of the pointers. In order to implement Equation 1 completely, we need a *mask* term to control which pointers will be affected. We accomplish this by selectively inhibiting the activity of the *channel* populations. If a channel is inhibited, the corresponding pointer population will not be affected by *x*.

As described more completely in Stewart et al. (2017), we implement all of this using standard Leaky Integrate-and-Fire spiking neurons using the simulation software Nengo (Bekolay et al, 2014). The resulting behaviour of the system loading two pointers (FIVE and SEVEN) is shown in Figure 2.

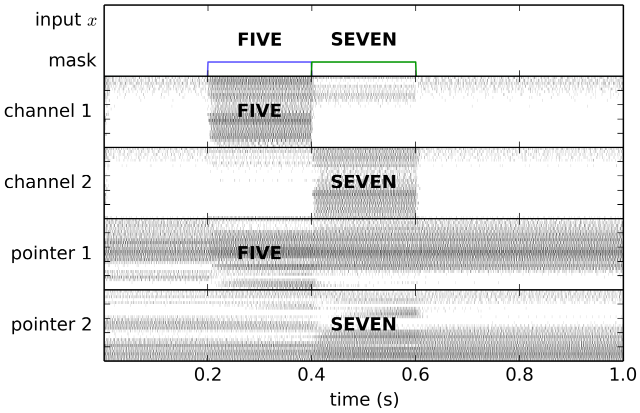


Figure 2: Spiking activity for an example magnitude comparison task. Top row shows input to the model. Other rows show spiking neuron activity over time. The text indicates which vector ***x*** is represented by the pattern of activity. Note that pointer 1 and pointer 2 maintain their spiking pattern (approximately) after the input has been removed. Figure from (Stewart et al., 2017).

In order to perform the two separate tasks, we then connect this same common component to one of two different output systems. For the finger gnosis task, the output is simply the identity function again, as all we need to do is to report the information stored in the pointers.

For the magnitude comparison task, we need a slightly more complex output. Rather than reporting the two stored numbers, we need to report whether the first number is larger or smaller than the second number. This is, itself, a function. So, in order to compute this, we use the NEF to solve for the optimal connection weights that will best approximate the function that maps the vectors for the two numbers to a single scalar output that is +1 if the first number is larger, and -1 if the second number is larger. We can think of this as training a group of neurons to memorize this list of desired inputs and outputs:

input	output
[ONE, TWO]	-1
[TWO, ONE]	+1
[ONE, THREE]	-1
[THREE, ONE]	+1
[TWO, THREE]	-1
...	...
[NINE, EIGHT]	+1

When we run this model, we treat a positive output as selecting the first number, and a negative output as selecting the second number. In Stewart et al. (2017) we also use the magnitude of this output to predict reaction times, but do not do that here.

Results

The basic results presented in Stewart et al. (2017) are shown in Figure 3. This includes both the model result and the empirical result gathered from human participants. We plot the percent error for the magnitude comparison task and for the finger gnosis task. Importantly, we fit the model parameters based on the magnitude comparison task *only*, leaving the finger gnosis task as a pure prediction based on those same parameter values.

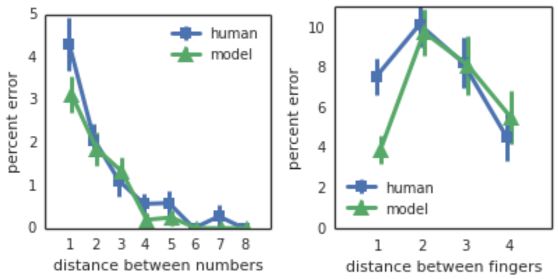


Figure 3: Best-fit model results for the magnitude comparison task (left) and finger gnosis (right). Parameters are fit on the magnitude task and then applied to the finger gnosis task. Standard errors are shown.

The best-fit parameters are as follows:

parameter		value
# neurons for combining pointer values:	n_c	1000
Standard deviation of training noise:	$noise_t$	0.15
Amount of channel inhibition:	c	0.875
Dimensionality of x vector:	D	8
Uniqueness of digit representation:	u	1.0

This core result indicates that the model captures the basic characteristics of the behavioural data. For magnitude comparison, we see the standard *distance effect*, where numbers that are farther apart (e.g. 2 and 7) are easier than numbers that are closer together (e.g. 5 and 6). We see a similar effect for finger gnosis, with the exception of when two fingers right next to each other are touched. In the human participant data, fingers next to each other are easier than fingers that are two apart. The model shows this same effect, but it is much more pronounced.

Parameter Exploration

To further characterize this model, we systematically varied these parameters. Importantly, since our theory is that both of these tasks use the same common neural system, whatever parameter value is used for one task should also be used for the other task.

However, this is only true for an individual person. It is plausible that, if this model is correct, different people may have different parameter values for this system. Thus, by changing these parameter values we make predictions about how performance on these two tasks may co-vary in individuals.

Parameter 1: n_c

The first parameter is the number of neurons to use to combine together the outputs from all of the pointers. Once combined together in this way, we can either create output connections that compute which of the two numbers is biggest (for magnitude comparison) or that just compute the identity function (for finger gnosis). However, the accuracy of this computation will be affected by the number of neurons used. This is shown in Figure 4.

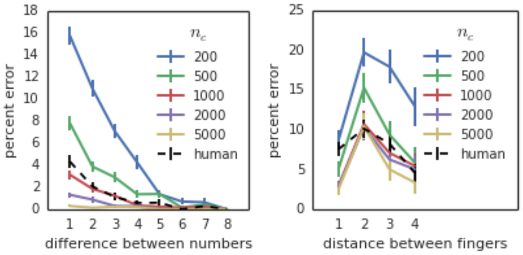


Figure 4: The effects of varying the number of neurons used to combine the represented pointer values together for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

From this, we note that 500 or fewer neurons gives significantly higher error rates than the mean human performance on both tasks. Having more than 1000 neurons gives improved performance for the magnitude comparison task and most of the finger gnosis task, but does not improve the peak error at a finger distance of two.

Parameter 2: $noise_t$

Next, we look a the amount of random noise used when finding the connection weights out of this combined population. That is, the neural activity from this combined population must cause change in a separate population that represents the network's response to the task. This change is, of course, due to synaptic connection weights. When we use the Neural Engineering Framework to solve for these weights, we can specify how much random variability is added. The right amount noise should make the network more robust to random variations, but too much noise will cause it to lose accuracy. The results are shown in Figure 5.

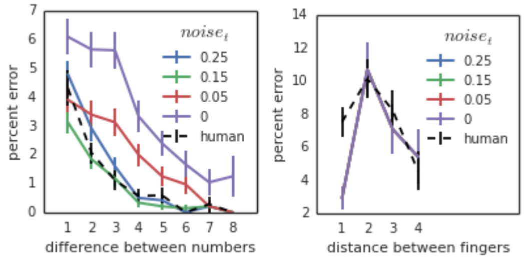


Figure 5: The effects of varying the amount of noise used in training the weights for the tasks for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

For the magnitude comparison task, we see the expected effect where there is an optimal value for this noise (0.15). Less noise than this gives extremely poor results for all distances. Interestingly, having more noise than this only increases the error for small differences between the numbers.

For the finger gnosis task, we get the surprising result that the model is unaffected by the amount of noise.

Parameter 3: c

Next, we examine the inhibition factor which turns off the channels leading into each pointer. With a value of 1, this inhibition would perfectly inhibit all of the neurons in the non-active channels, leading to no activity in those channels, and thus no change in the other pointer values. If this is less than one, however, the neurons will not be perfectly “turned off”, and so there will be some small influence on the other pointers when one of them is set. For example, in Figure 2, we see some neural activity in the channels that are not being set, reflecting c<1. We assume this amount of inhibition scales linearly with the distance from the target item, so values sent into pointer 1 have more influence on pointer 2 than they do on pointer 3. Results are in Figure 6.

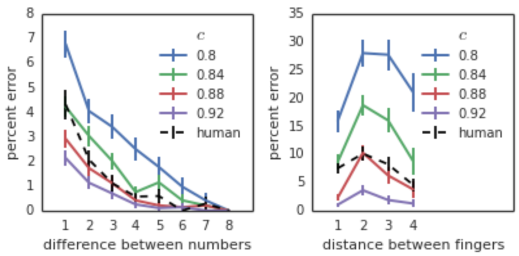


Figure 6: The effects of varying the amount of channel inhibition for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

In this case, we get a clear result that there is an optimal value for c (we found 0.875 best). Importantly, this optimal value works for both the magnitude comparison task and the finger gnosis task.

Parameter 4: *D*

Finally, we vary the dimensionality of the input stimulus x. This controls the degrees of freedom in the randomly chosen patterns for each represented concept (ONE, TWO, TOUCHED, etc). Results are shown in Figure 7.

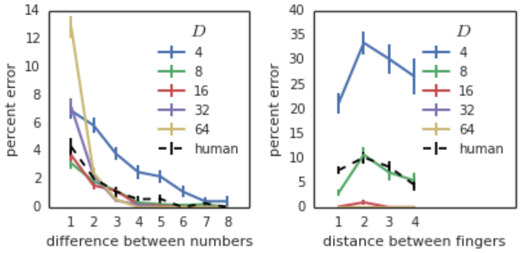


Figure 7: The effects of varying the dimensionality of *x* for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

Here, we see that the magnitude comparison task indicates that if *D* is too large (i.e. 32 or 64), it produces a large increase in the error, but just for the case where the difference between the numbers is 1. It also produces a large increase in error overall if *D* is too small (i.e. 4). For the finger gnosis task, small *D* produces a very large increase in error as well, but large *D* (above 8) causes a massive decrease in the error.

Number Representation

If we consider just the magnitude comparison task, there is a further parameter that is worth investigating. This is the question of how numbers are represented in the model. In particular, should the representation for TWO be more similar to THREE than it is to NINE? After all, as can be seen in the human data, participants are more likely to make mistakes when number are close to each other, which seems to imply that the neural activity for TWO should be more similar to the activity for THREE than it is to NINE.

As we are using a vector representation in this model, this becomes the question of how to choose what vector to use for ONE, TWO, THREE, etc. In the simplest case, we can choose these vectors completely randomly, so that there is no similarity structure. At the other extreme, we could randomly choose a vector for ONE, a different vector for NINE, and then smoothly interpolate between these two to create the vectors for TWO, THREE, FOUR, etc. To explore this, we define a parameter u which interpolates between fully random representation where each number is represented with a different random unique number (*u*=1.0) and fully structured representation where TWO is halfway between ONE and THREE (*u*=0.0).

The effects of varying this uniqueness parameter are shown in Figure 8. Crucially, if there is low uniqueness (i.e. if the neural representation of TWO is more similar to ONE than it is to NINE), then we reach a much higher error rate than is observed in the human data.

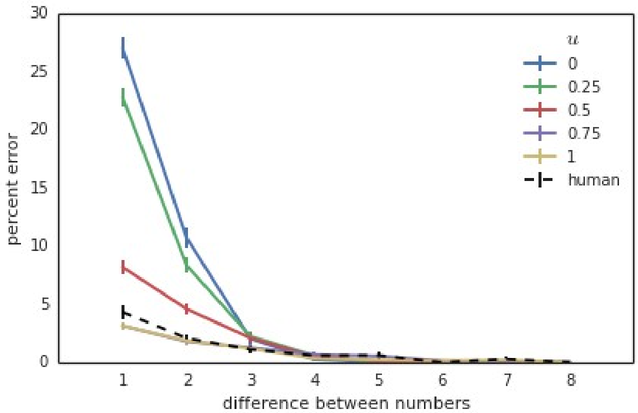


Figure 8: The effects of varying the uniqueness of the number representation in the magnitude comparison task. Standard errors are shown.

This was a surprising result for us. The observed error pattern in the human data (where numbers that are close to each other are more likely to produce errors) is *not* the result of the actual neural representation of the numbers being similar to each other. Rather, this pattern of errors is due to mistakes made in *extracting* the information from the group of neurons. When neurons are used to approximate the "which number is larger" function, the optimal connection weights lead to a system which is more likely to make mistakes between nearby numbers, *even though they are not "nearby" in terms of neural activity*. They are, however, nearby in terms of the function being computed.

Conclusions

We have examined the behaviour of a model of how magnitude comparison and finger gnosis can both rely on the same common neural component: a system for storing an array of pointers. Since this neural system is believed to be used in both tasks, by varying the parameters of this system we produced predictions of how an individual's error performance on both tasks can be related. However, it should be noted that all of the comparisons performed in this paper were to the *mean* human performance. The next step is to look at individual differences in this task and determine if the same patterns occur in the participant data. If it does, then we may have an explanation for this variation in terms of different people having different parameter settings for this common component.

Furthermore, we have a novel explanation as to why the *distance effect* exists. In our model, the distance effect (the fact that more errors are made when two numbers are close in magnitude) is *not* due to those two numbers having similar neural representations. Rather, the neural representation of each number is completely random. If we do impose some similarity in the neural representation, then the distance effect becomes much larger than it is in the participant data. This means that in our model, the distance effect emerges purely from the difficulties involved in generating synaptic connections that determine which of the two numbers is larger, rather than the more typical interpretation that it comes from similarities in the neural representation itself.

References

Andres, M., Michaux, N., & Pesenti, M. (2012). Common substrate for mental arithmetic and finger representation in the parietal cortex. *Neuroimage*, 62(3), 1520-1528.

Baron, I. S. (2004). *Neuropsychological Evaluation of the Child*. New York, NY: Oxford University Press.

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T.C., Rasmussen, D., Choo, X., Voelker, A., & Eliasmith, C.. (2014). Nengo: a python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*

Brozzoli, C., Ishihara, M., Göbel, S. M., Salemme, R., Rossetti, Y., and Farnè, A. (2008). Touch perception reveals the dominance of spatial over digital representation of numbers. *Proc. Natl. Acad. Sci. U.S.A*. 105, 5644–5648.

Dehaene, S., Tzourio, N., Frak, V., Raynaud, L., Cohen, L., Mehler, J., & Mazoyer, B. (1996). Cerebral activations during number multiplication and comparison: A PET study. *Neuropsychologia,* 34, 1097-1106.

Eliasmith, C. & Anderson, C.. (2003). *Neural engineering.* MIT Press, Cambridge, MA.

Fayol, M., Barrouillet, P., & Marinthe, C. (1998). Predicting arithmetical achievement from neuro-psychological performance. *Cognition*, 68, B63-B70.

Noël, M.-P. (2005). Finger gnosia: A predictor of numerical abilities in children? *Child Neuropsychology*, 11, 413-430.

Rusconi, E., Walsh, V., & Butterworth, B. (2005). Dexterity with numbers: rTMS over left angular gyrus disrupts finger gnosis and number processing. *Neuropsychologia*, 43(11), 1609-1624.

Penner-Wilger, M., & Anderson, M. L. (2013). The relation between finger gnosis and mathematical ability: Why redeployment of neural circuits best explains the finding. *Frontiers in Psychology,* 4, 877.

Penner-Wilger, M., & Anderson, M. L. (2008). An alternative view of the relation between finger gnosis and math ability. *30th Cognitive Science Society*. 1647-1652.

Penner-Wilger, M., Fast, L., LeFevre, J., Smith-Chant, B. L., Skwarchuk, S., Kamawar, D., & Bisanz, J. (2007). The foundations of numeracy: Subitizing, finger gnosia, and fine-motor ability. *29th Cogn. Science Society,* 1385-1390

Penner-Wilger, M., Fast, L., LeFevre, J., Smith-Chant, B. L., Skwarchuk, S., Kamawar, D., & Bisanz, J. (2009). Subitizing, finger gnosis, and the representation of number. *31st Annual Cognitive Science Society*, 520-525

Penner-Wilger, M., Waring, R. J., & Newton, A. T. (2014). Subitizing and finger gnosis predict calculation fluency in adults. *36th Cognitive Science Society*, 1150-1155.

Penner-Wilger, M., Waring, R. J., Newton, A. T., & White, C. (2015). Finger gnosis and symbolic number comparison as robust predictors of adult numeracy. 37th Conference of the Cognitive Science Society, 2963.

Penner-Wilger, M., Waring, R. J., Newton, A. T., & White, C. (in prep). Finger gnosis as a robust predictor of numeracy in children and adults.

Roux, F.-E., Boetto, S., Sacko, O., Chollet, F., & Tremoulet, M. (2003). Writing, calculating, and finger recognition in the region of the angular gyrus: a cortical study of Gerstmann syndrome. *J. Neurosurgery*, 99, 716-727.

Stewart, T.C., Penner-Wilger, M., Waring, R., and Anderson, M. (2017). A Common Neural Component for Finger Gnosis and Magnitude Comparison. *Annual Meeting of the Cognitive Science Society.*

Zago, L., Pesenti, M., Mellet, E., Crivello, F., Mazoyer, B., & Tzourio-Mazoyer, N. (2001). Neural correlates of simple and complex mental calculation. *NeuroImage*, 13, 314-327.

Parameter exploration of a neural model of state transition probabilities in model-based reinforcement learning

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

Abstract

We explore the effects of parameters in our novel model of model-based reinforcement learning. In this model, spiking neurons are used to represent state-action pairs, learn state transition probabilities, and compute the resulting Q-values needed for action selection. All other aspects of model-based reinforcement learning are computed normally, without neurons. We test our model on a two-stage decision task, and compare its behaviour to ideal model-based behaviour. While some of these parameters have expected effects, such as increasing the learning rate and the number of neurons, we find that the model is surprisingly sensitive to variations in the distribution of neural tuning curves and the length of the time interval between state transitions.

Keywords: neural model; reinforcement learning; model-based reinforcement learning; Neural Engineering Framework

Introduction

Reinforcement learning (RL), a formalization of reward-based decision making, is often divided into two sub-types: model-free and model-based (Sutton & Barto, 1998). This distinction has been used by neuroscientists to explain aspects of instrumental conditioning in humans and other animals. Daw, Niv, and Dayan (2005) drew parallels between the habit system (where actions are performed automatically) and model-free RL; and between the goal-directed system (where actions show evidence of planning) and model-based RL. Model-free and model-based learning have been proposed to be realized in the brain with separate systems that rely on different prediction error signals (Glascher, Daw, Dayan, & O'Doherty, 2010). There has been extensive research on model-free RL, including work on how it may be instantiated in the brain according to the reward prediction error theory of dopamine (e.g., Barto, 1995). There is significantly less focus on model-based RL, including a particularly evident lack of suggestions as to how it may happen in the brain (Friedrich & Lengyel, 2016).

We have developed a novel model of model-based RL that uses spiking neurons to represent state-action pairs, learn state transition probabilities, and compute the resulting Q-values needed for action selection. The present work aims to investigate the factors that influence the behaviour of this neural model.

Background

In both model-free and model-based RL approaches, the goal is to learn from experience how valuable different actions are, given the current state (Sutton & Barto, 1998). This is written as $Q(s,a)$, where s is the state, and a is the action.

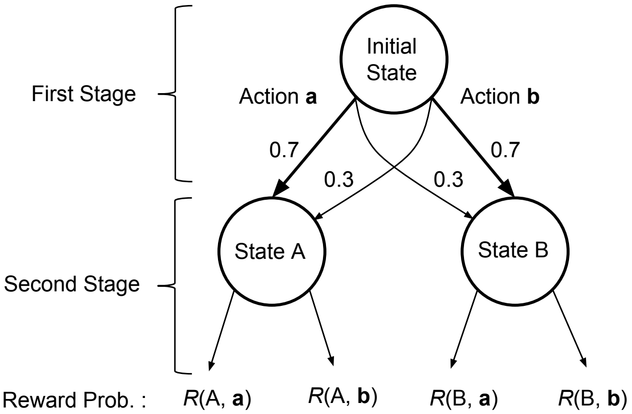


Figure 1: Schematic diagram of the two-stage task. The numbers on the arrows in the first stage indicate the probability of a particular state transition, given the chosen action. The reward after performing an action in the second stage is randomly determined based on the reward probability. Adapted from Akam et al. (2015).

In model-based RL, the value Q of an action in a particular state is given by the Bellman equation (Bellman, 1957):

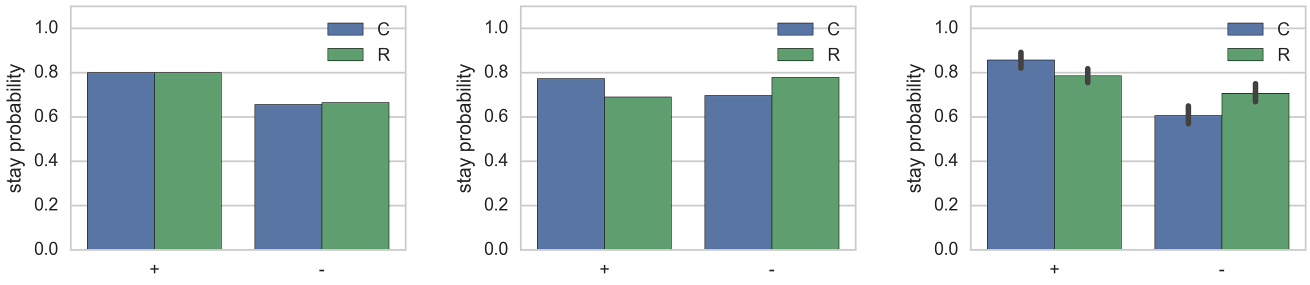
$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s,a,s') \max_{a} Q(s',a), \qquad (1)$$

where R is the expectation of reward, P is the probability of transitioning from state s to s' if action a is performed, and γ is a future discounting parameter. Importantly, in order to compute this, the system needs to know the state transition probabilities $P(s,a,s')$. This set of probabilities can be thought of as a model of the environment, and is why this approach is called model-based.

In contrast, model-free RL does not create an explicit representation of the environment. Rather, it just uses whatever state s_{t+1} occurs after the action a in the current state s_t. This can be thought of as an estimate of the typical next state that will occur. This leads to an approximation of Eq. 1 that does not take into account the transition probabilities, but is much simpler to compute:

$$Q(s_t,a) = R(s_t,a) + \gamma \max_{a} Q(s_{t+1},a). \qquad (2)$$

Model-free RL constructs an estimate of Eq. 1 through direct experience in the environment, which leads to an implicit



(a) Ideal model-free behaviour

(b) Ideal model-based behaviour

(c) Human behaviour

Figure 2: Ideal and human stay probability behaviour on the two-stage task. C denotes common, and R denotes rare, state transitions. Trials are either rewarded (+) or unrewarded (−). Adapted from Daw et al. (2011).

representation of environmental statistics. In contrast, model-based RL constructs an internal model of the probabilities of reward *R* and state transitions *P*. This explicit learning of the statistics is used to directly calculate the Bellman equation.

Given this difference, there are situations where the learning trajectories will differ between model-free and model-based RL. As discussed in the next section, particular learning tasks can be defined to distinguish the two approaches.

Two-stage task

We test our model on the two-stage task described in Daw et al. (2011). A schematic diagram of this task is shown in Fig. 1. The first stage consists of an initial state, which has two possible actions (**a** and **b**). These actions lead probabilistically to one of the two second stage, or terminal, states (A and B), with action **a** commonly transitioning to state A and action **b** commonly transitioning to state B. These common state transitions each have a probability of 0.7; correspondingly, rare transitions have a probability of 0.3. In states A and B, actions **a** and **b** are again available, and are rewarded with probability determined by a Gaussian random walk, so that the immediate reward after performing an action in a second stage state is either 0 or 1. This randomness was introduced to enforce continued learning throughout the task.

This task was developed to discriminate model-based from model-free behaviours, using the stay probability, i.e., the likelihood of choosing the same initial-state action in trial $n+1$ as in trial n (Daw et al., 2011). In particular, if an agent finds itself in a rare (R) second-stage state (given their action in the initial state), and performs an action that is rewarded (+), a purely model-free strategy would increase the value of performing that first-stage action, as shown in Fig. 2a, while a purely model-based strategy would increase the value of being in that second-stage state, thus increasing the value of the unchosen first-stage action and decreasing the stay probability (see Fig. 2b, R+). By similar logic, in a rare, unrewarded state (R−), a model-based agent would increase the value of choosing the initial action, thus increasing the stay probability.

As an example of model-based reasoning, say an agent has found itself in state B after performing action **a**. The agent "knows" this is a rare transition. Now say the agent performs some action and receives a reward. This increases the value of being in state B, so the agent wants to return to this state. Since the agent knows that state B is more commonly reached after performing action **b** in the initial state, it will also increase the value of performing action **b** in the initial state and correspondingly *decrease* the value of performing action **a** in that state. Conversely, a model-free reasoner would simply increase the value of every state-action pair it performed before receiving the reward, and so the value of performing action **a** in the initial state would *increase*.

Daw et al. (2011) found that human behaviour on this task showed characteristics of both model-free and model-based strategies (see Fig. 2c). In particular, there is a statistically significant difference between rare-rewarded (R+) and common-unrewarded (C−) probabilities not evident in model-based behavior, and there are elevated rare-unrewarded (R−) and common-rewarded (C+) probabilities relative to the C− probabilities that is not evident in model-free behaviour.

Model

Fig. 3 shows a schematic diagram of the model we built using the Nengo neural simulator (Bekolay et al., 2014), which is based on the principles of the Neural Engineering Framework (NEF; Eliasmith & Anderson, 2003). The NEF provides methods to generate neurons with random properties (such as tuning curve distributions) and then arrange them so that they best approximate a given representation or transformation.

The neural model includes two components of model-based RL: 1) the representation of the state transition probabilities $P(s,a,s')$, and 2) the multiplication of these probabilities by the Q-values of the future states to produce an estimate of the Q-values of the current state's actions. We implement these components using spiking leaky-integrate-and-fire (LIF) neurons (Lapicque, 1907) via the NEF, while the rest of the model-based RL system is implemented using traditional computation.

In the model, the states and actions are represented by vectors. For simplicity of explanation, we refer to these vectors as orthogonal, with the smallest possible dimensionality (three dimensions to represent the three states, and two for the two actions); however, our methods allow these vectors to be arbitrarily large. In the model, we use 5-dimensional vectors. For purposes of explanation, we will use 3-dimensional vectors, and assume state A is represented by $S_A = [1,0,0]$, state B by $S_B = [0,1,0]$, and the initial state by $S_0 = [0,0,1]$. The rest of the model components, shown in rectangles in Fig. 3, are implemented directly without neuron approximation. These components perform action selection, track and update the environment's actual state transitions and reward, learn the model-free Q-values of actions in states A and B, and store all the Q-values (for use in action selection).

In the design of our model, we exploit the natural parallelism of a neural implementation. In traditional model-based approaches, the state transition probabilities $P(s,a,s')$ are stored in lookup tables. However, in our model, these probabilities are represented by a function computed in a connection between neural populations that maps state-action inputs to a second-state probability distribution as follows:

$$P(s,a) = [P(s,a,S_A), P(s,a,S_B), P(s,a,S_0)]. \tag{3}$$

For example, $P(S_0,\mathbf{a}) = [0.7,0.3,0.0]$ in the two-stage task.

In a traditional model-based agent, the value of actions in states in the first stage (in this case, the single initial state) are recalculated at the beginning of each trial, with:

$$Q(s,a) = \sum_{s'} P(s,a,s') \max_{a} Q(s',a). \tag{4}$$

In our model, this calculation is done in neurons. Specifically, using our modified representation of the state transition probabilities, we calculate the following dot product:

$$Q(s,a) = P(s,a) \cdot Q(a'), \tag{5}$$

where $Q(a')$ is the vector of the best possible action for every state.

Although multiplications are non-linear, they have a well-characterized implementation in neurons that can be implemented accurately with the NEF (Gosmann, 2015). A neural population, called *Product* in Fig. 3, performs an element-wise multiply based on this characterization, and a summation is performed by the output connections to compute Eq. 5.

To illustrate how this would be done by a model-based agent using the two-stage task, consider an agent that is in the initial state and considering performing action **a** (i.e., $s = S_0, a = \mathbf{a}$). The agent remembers the Q-values of the best possible action a' for every state, say $Q(a') = [0.25,0.75,0.33]$, as well as the probability of reaching that state given the current state and action (as before, $P(S_0,\mathbf{a}) = [0.7,0.3,0.0]$). To calculate the Q-value of the current state and considered action, it performs the dot product of these two vectors, producing a value of $Q(S_0,\mathbf{a}) = 0.4$. It then follows the same process to consider action **b**, and finds $Q(S_0,\mathbf{b}) = 0.6$.

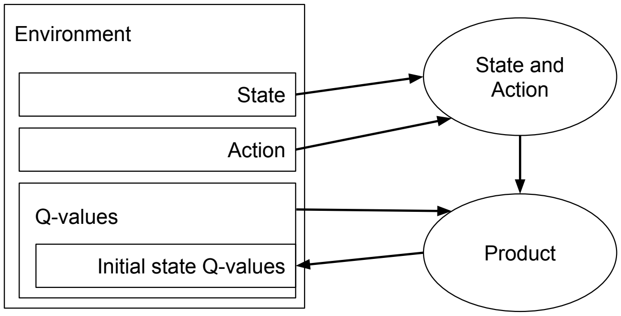


Figure 3: Model diagram. The components of the model shown in oval shapes are simulated populations of neurons that perform representations and transformations. The rectangular components are directly computed without neuron approximation. The connection between the *State and Action* population and the *Product* population calculates Eq. 3 according to the ideal state transition probabilities. The *Product* population performs an element-wise multiply between the transition probabilities and the Q-values stored in the environment. The connection from *Product* to the environment adds the results of that multiply. These last two steps together calculate Eq. 5.

The resulting Q-values need to be updated for both possible actions in the initial state. One possibility is to perform this in parallel (i.e., to have separate groups of neurons that perform this computation for each possible action). However, since this may be problematic if the number of actions grows to be large (or is unknown), we consider a serial strategy. Specifically, we have assumed the neural system considers actions one after the other over time. Although the actions are considered sequentially, all of the possible future states following those actions are considered in parallel.

Consistent with previous (non-neural) models of this task (Daw et al., 2011), the values of actions in the terminal, stage-two states are updated using a version of Q-learning:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha r, \tag{6}$$

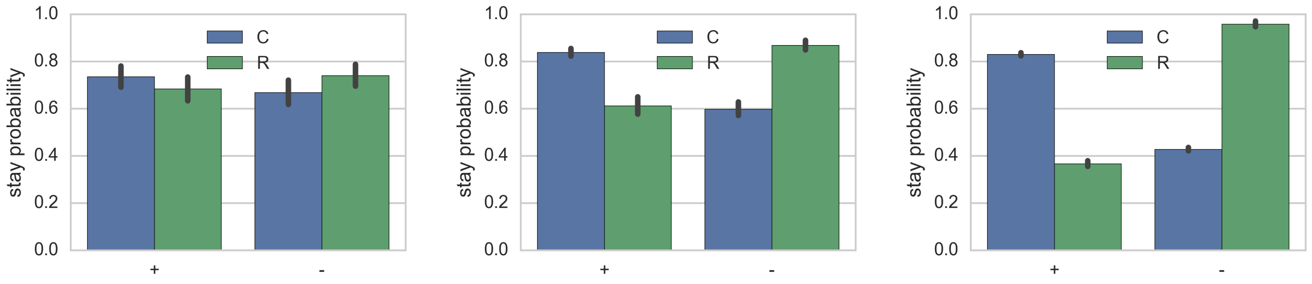
where α is a learning rate parameter and r is the immediate reward (Akam et al., 2015). This calculation is done directly, rather than with neurons.

Action selection is performed by approximating a softmax. That is, to determine the action performed in a given state, a small amount of random noise is added to the Q-values for all the actions in that state before selecting the action with the highest Q-value.

Method

We explore four parameters that influence the model's behaviour on the two-stage task:

1. The **learning rate** α (from Eq. 6), which affects how much the Q-values of the terminal states are changed after receiving a reward.



(a) learning rate = 0.05

(b) learning rate = 0.3

(c) learning rate = 1

Figure 4: Examples of different stay probability behaviours for various learning rates with a Nengo seed of 1. Error bars are 95% confidence intervals.

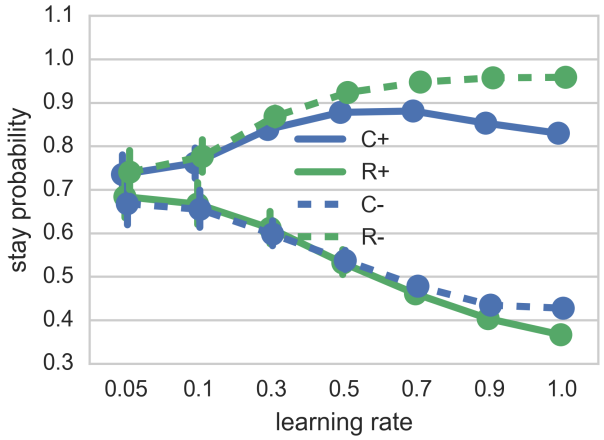


Figure 5: Stay probabilities of different learning rates. Error bars are 95% confidence intervals, and are sometimes smaller than data point markers.

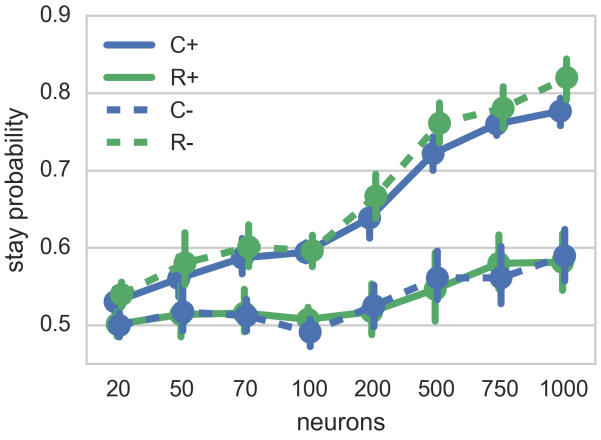


Figure 6: Stay probabilities of different numbers of neurons in the *State and Action* population. Model-based behaviour is clearly distinguishable with 100 or more neurons. Error bars are 95% confidence intervals.

2. The **number of neurons** in the *State and Action* and *Product* populations, which affects the accuracy of the representation and multiplication.
3. The random properties of the neurons used in the model are determined by Nengo according to a random seed. Different values of this seed produce different random distributions of neural tuning curves, so versions of the model instantiated with these different seeds can be thought of as different **individuals**.
4. The **time interval** between state transitions. The effect this is expected to have is that if the time interval is too short, the neurons will not have adequate time to compute the Q-values, and so the stay probability may be uniform in all rewarded (+) or unrewarded (−) and common (C) or rare (R) cases, since the agent is choosing actions based on essentially random information.

For each tested case, we run our simulations with twenty sessions of 10000 trials each. Each session is run with a different random seed for the environment, which determines the random behaviour of state transitions, the random noise in the action selection, and the random walks of reward probabilities.

Results

Learning rate

As shown in Figs. 4 and 5, as the learning rate α is increased, the effect of model-based reasoning is also increased; that is, the stay probabilities of the C+ and R− cases are increased, while the stay probabilities of the R+ and C− cases are decreased. When $\alpha = 1.0$, the most extreme example, there is no model-free learning; the Q-value of the terminal states is simply the reward (0 or 1) that was most recently received. In this situation, when calculating the Q-values of the initial state as in Eq. 4 or 5, the Q-values will either be exactly the

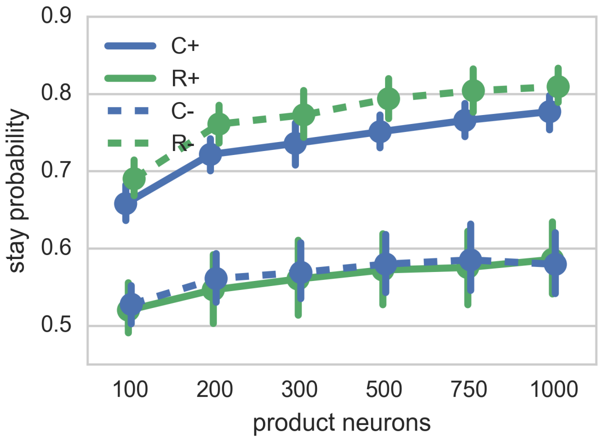


Figure 7: Stay probabilities of different numbers of neurons in the *Product* population. Error bars are 95% confidence intervals.

state transition probability, or 0. When α is a lower, for example α = 0.05, much more emphasis is put on the learned values of the terminal states, and those values are learned much more slowly, and so they interfere with the model-based reasoning.

Number of neurons

In general, decreasing the number of neurons in the *State and Action* population has a similar effect to decreasing the learning rate: the stay probabilities of the C+ and R− cases decrease and the stay probabilities of the R+ and C− cases increase as the number of neurons decreases. This trend is shown in Fig. 6. Populations of at least 100 neurons were sufficient for producing clearly model-based stay probability behaviours. For most other simulations, a default value of 500 neurons was chosen because it produces a clear separation between C+, R− and R+, C− stay probabilities.

Increasing the number of neurons in the *Product* population above 200 per dimension did not produce any significant benefits, as shown in Fig. 7.

Individual

As shown in Fig. 8, there are large individual differences between different Nengo seeds. Many of them produce pure model-based stay probability plots (Fig. 8a), while some have a significant difference in stay probabilities between the R+ and C− cases that is reminiscent of human data (Fig. 8b), and in others, that significant difference is in the opposite direction to the human data (Fig. 8c). However, when averaged across individuals, the stay probabilities are characteristically model-based.

Time interval

As expected, it is necessary for the time interval between state transitions to be sufficiently long in order for the neurons to compute the Q-values. However, the individual differences discernible between different Nengo seeds are also dependent on the length of the time interval between state transitions; surprisingly, there is no apparent relationship between the length of the time interval and the stay probability behaviour. Three examples of stay probability behaviour with a Nengo seed of 1 are shown in Fig. 9. These can also be compared to Fig. 8b, which shows the same seed with a time interval of 50ms. Of particular interest is the discrepancy between Figs. 9b and 9c, since these time intervals have only a 10ms difference, yet show almost opposite stay probability behaviours.

Discussion

The core result of the research presented here is that, in general, the neural model of model-based reinforcement learning matches the expected results of a model-based agent. This is demonstrated by data aggregated across individual Nengo seeds, as well as particularly clearly by the trend produced by varying the learning rate α. The value of alpha that produced the greatest difference between the C+, R− and R+, C− stay probabilities (demonstrating a strong *model-based* effect) was α = 1.0. This suggests that it may not be necessary for models of the two-stage task to use the model-free Q-learning component to estimate the values of terminal states (Eq. 6), since model-based stay probability behaviour can be produced when the values of the terminal states are taken to be the immediate reward.

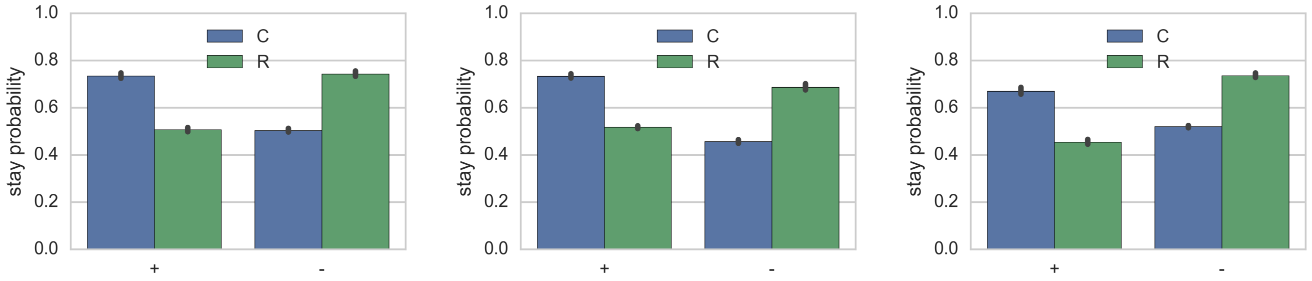
The effect of varying the number of neurons in the *State and Action* population is also as expected for a purely model-based agent. As the number of neurons increases, the representation of the current state and action is improved, which increases the likelihood of calculating the correct state transition probability as the input to the *Product* network.

The stay probability pattern reminiscent of human data reappeared with a number of Nengo seeds and time intervals. The individual differences between Nengo seeds demonstrates that the stay probability behaviour is surprisingly sensitive to the distributions of neurons. Future work will be done to further investigate this result.

The most unforeseen result was that of the length of the time interval and its interaction with the Nengo seed. Increasing the length of the synaptic filter may eliminate this irregular effect; future work will investigate this and other possibilities.

Conclusion

Our investigation of the effects of four parameters on the stay probability behaviour of a neural model of model-based reinforcement learning established that it typically performs as expected of a model-based agent. However, individual differences between certain parameter values demonstrated the model's sensitivity to the distribution of neural tuning curves and the time interval between state transitions.

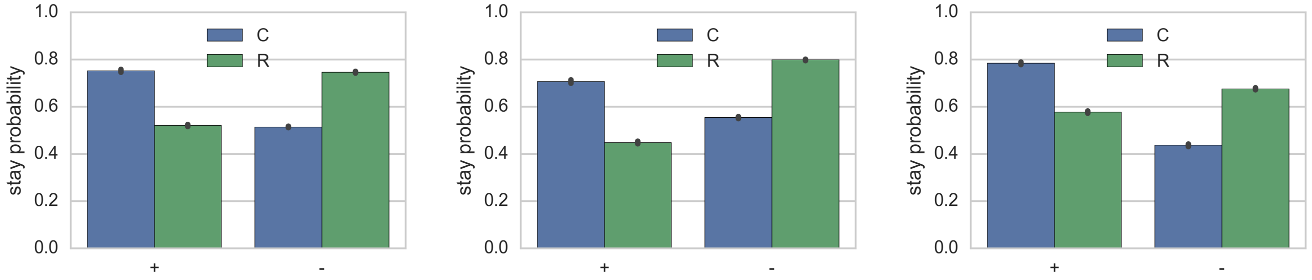


(a) seed = 8

(b) seed = 1

(c) seed = 7

Figure 8: Examples of individual differences between Nengo seeds: (a) pure model-based stay probability behaviour, (b) stay probability behaviour suggestive of human data, and (c) stay probabilities opposite to those in (b). All trials were run with a time interval of 50ms. 95% confidence intervals are shown.



(a) time interval = 200ms

(b) time interval = 500ms

(c) time interval = 510ms

Figure 9: Examples of different behaviours for various time intervals with a Nengo seed of 1. Error bars are 95% confidence intervals.

Notes Supplemental material, including python scripts, is available at https://github.com/ctn-waterloo/cogsci17-rl/refactor.

Acknowledgments

This work was supported by CFI and OIT funding, the Canada Research Chairs program, NSERC Discovery grant 261453, ONR grant N000141310419, AFOSR grant FA8655-13-1-3084, and OGS and NSERC CGS-M graduate funding.

References

Akam, T., Costa, R., & Dayan, P. (2015). Simple plans or sophisticated habits? State, transition and learning interactions in the two-step task. *PLoS Computational Biology*, *11*(12).

Barto, A. G. (1995). *Adaptive critics and the basal ganglia* (J. C. Houk, J. Davis, & D. Beiser, Eds.). Cambridge, MA: MIT Press.

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., ... Eliasmith, C. (2014). Nengo: A python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, *7*(48).

Bellman, R. E. (1957). *Dynamic programming*. Princeton: Princeton University Press.

Daw, N. D., Gershman, S., Seymour, B., Dayan, P., & Dolan, R. J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, *69*, 1204–1215.

Daw, N. D., Niv, Y., & Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience*, *8*(12), 1704–1711.

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

Friedrich, J., & Lengyel, M. (2016). Goal-directed decision making with spiking neurons. *Journal of Neuroscience*, *36*(5), 1529–1546.

Glascher, J., Daw, N., Dayan, P., & O'Doherty, J. P. (2010). States versus rewards: Dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron*, *66*(4), 585–595.

Gosmann, J. (2015). *Precise multiplications with the nef* [Technical Report]. University of Waterloo, Waterloo, Ontario, Canada.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *Journal de Physiologie et de Pathologie Générale*, *9*, 620–635.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge: MIT Press.

Basal Ganglia-Inspired Functional Constraints Improve the Robustness of Q-value Estimates in Model-Free Reinforcement Learning

Patrick J. Rice (pjrice@uw.edu)
Department of Psychology, University of Washington
Campus Box 351525, Seattle, WA 98195 USA

Andrea Stocco (stocco@uw.edu)
Department of Psychology, University of Washington
Campus Box 351525, Seattle, WA 98195 USA

Abstract

Due to the correspondence between the striatal dopamine signal and prediction error signal utilized by model-free reinforcement learning methods, computational psychological research has found much success in modeling the basal ganglia as a biological implementation of a reinforcement learning mechanism. A large majority of these modeling efforts have focused on applying the tenets of reinforcement learning to the proposed functions of the basal ganglia, but few (if any) have attempted to apply crucial aspects of basal ganglia neurophysiology to reinforcement learning mechanisms. Here, we propose a basal ganglia-plausible model that explicitly utilizes two symmetric sets of actions (analogous to the basal ganglia's direct and indirect pathways), to simultaneously update value estimates of both available actions (i.e. chosen and not chosen) in the Probabilistic Stimulus Selection (PSS) task. We demonstrate that this proposed model architecture outperforms a standard reinforcement learning model of the PSS task by eliminating the standard model's bias towards estimation of the most valuable available actions, while granting improved resistance to noise in the internal selection process.

Keywords: Reinforcement learning; basal ganglia; dopamine; computational models

Introduction

Model-free reinforcement learning (RL) is a powerful approach for obtaining an optimal long-term action policy in the absence of transition probability and reinforcement functions. In other words, a model-free RL agent must interact with an unknown environment (i.e., sample the environment repeatedly through action) in order to construct an optimal control policy, based on the pattern of reward received by interaction with the environment. This framing of RL methods makes clear their power in modeling human and animal decision-making. Policies refined through RL mechanisms are oriented such that the agent's (i.e., human/animal) actions consider both immediate and future reward, optimized to maximize some value over time. The key idea that enables an agent to determine an optimal policy within an unknown environment is that of temporal-difference (TD) learning (Sutton, 1988).

The ideas behind TD methods have since been expanded, including a proposal by Watkins and Dayan (1992) that defined a TD control algorithm now known as Q-learning. Q-learning is an off-policy method that allows the agent to choose to take non-optimal actions while still estimating an optimal value function. By updating action values based on the best action available while allowing the agent to make inferior choices, this procedure increases the rate of learning under a suboptimal action selection process. Both TD learning and Q-learning have been shown to converge to the optimal value function with probability $P = 1$ (Sutton & Barto, 1998).

However, in some circumstances, these model-free RL methods produce suboptimal results. As defined, these methods emphasize learning of rewarding actions – updating the value of a state/state action (SA) pair increases the likelihood that the agent will choose the action that leads to that state/SA pair when it is next given the opportunity to do so. As a result, even though it converges to an optimal value function, an agent still does not have complete knowledge of its environment – namely, it does not know much (if anything) about the least rewarding states/SA pairs. This is an instance of the general exploration-exploitation trade-off that many models encounter. Lacking knowledge of the least rewarding alternatives is not an issue while the agent has full access to its actions, but what if learned "good" (i.e. optimal) states/SA pairs are blocked from the agent? In this case, the agent cannot take the actions it usually would by following its policy and value function, and as a consequence, cannot act optimally within the "new" environment. In essence, the agent has no knowledge of how to navigate bad options – how to choose the "least bad", when forced to.

The Probabilistic Stimulus Selection Task

A situation in which this circumstance arises is when modeling a well-known psychology task paradigm, the Probabilistic Stimulus Selection (PSS) task (Frank, Seeberger, & O'Reilly, 2004). The PSS task is a repetitive, two-alternative forced-choice task made up of two consecutive phases, a *training phase* in which a participant repeatedly makes choices between fixed pairs of stimuli, and a *test phase* where the participant is presented with new combinations of options (see Figure 1).

Across both phases, there are six possible stimuli, implemented as symbols that are difficult to describe (in order to make memorization of each stimulus history of success more difficult). Each stimulus carries an intrinsic probability of success, ranging linearly from 20% to 80%. During the training phase, the stimuli are presented a fixed pairs, for a total of three sets: (A,B) (C,D), and (E,F), with associ-

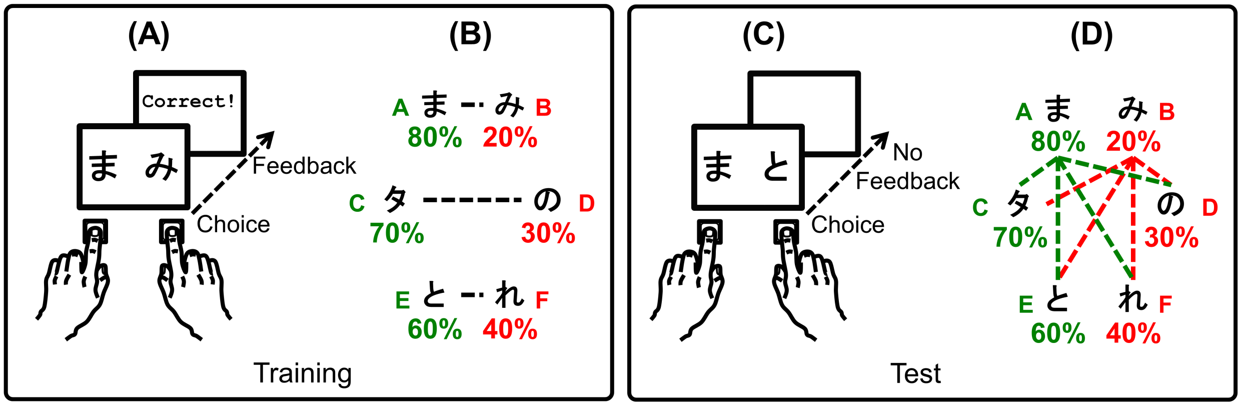


Figure 1: An overview of the Probabilistic Stimulus Selection task. In the *training phase*, participants learn to identify the best option within three pairs. In the]*test phase*, the six options appear in new paired combinations.

ated reward probabilities of $(80\%, 20\%)$, $(70\%, 30\%)$, and $(60\%, 40\%)$, respectively. Participants receive feedback regarding the outcome of their decision directly after making a selection, and are instructed to attempt to maximize their success by choosing what they believe to be the "correct" option on each trial. esponses that are probabilistically determined to be errors are associated with negative reward (i.e. "feedback"), while those deemed correct are associated with positive reward, with the consequence that a component of "good" performance is avoiding "bad" (i.e. low probability of success) choices. Once a participant's performance reached a predefined criterion (different for each pair: 65%, 60%, and 50% probability of choosing the higher valued option for the sets of (A,B) (C,D), and (E,F), respectively), the test phase begins. During the test phase, participants are shown all possible combinations of the six stimuli(fifteen total, four times each, for a total of 60 trials), and do not receive feedback upon selection. From the test phase, two different measures are calculated: the participant's *Choose accuracy*, that is, the probability of choosing the highest valued alternative (A; 80% reward probability) when it is paired with any other alternative, and the participant's *Avoid accuracy*, that is, the probability of not choosing the lowest valued alternative (B, 20% reward probability) when it is paired with any other alternative (excepting A). These measures can generally be interpreted to be the participant's tendencies to pursue reward and avoid punishment, respectively.

Human participants perform close to criterion in the test phase, with an average of about 70% accuracy in both Choose and Avoid accuracies (Frank et al., 2004; Frank, Moustafa, Haughey, Curran, & Hutchison, 2007; Stocco et al., 2017).

Model Comparisons

General Model Implementation

The PSS task poses a number of important constraints for the design of RL agents. In this section, we outline these constraints, and how they were addressed in the implementation of our agents.

The first constraint is that the set of actions available to an agent corresponds to the decision options in the task, that is, the six options $A, B \dots F$.

The second constraint is that an agent should be able to generalize the Q value of an action to an different state. This is essential to permit generalization of the Q-values learned during the training phase (Figure 1A and 1B) to the new set of pairs in the test phase (Figure 1C and 1D). A number of mechanisms have been proposed to generalize Q-values to new states. In this paper, we have taken the minimalistic approach of associating all actions to a single state s, but changing the set of actions available at every trial depending on the options presented. Thus, in a trial where the options A and B are presented, only the actions a_A and a_B will be selectable by the agent.

The third constraint is related to the second, and concerns the relationship between subsequent states in the PSS task. Because the PSS task consists of a sequence of *independent* trials, the probability of a state s_{t+1} following another state s_t does not depend on the action taken a_t. Canonical RL algorithms based on temporal difference rely on the environment states to be concatenated in some way, since the update term for the Q-value of an action taken at state s_t depends on the Q-value of the agent's actions at state s_{t+1} For example, in the Q-learning algorithm, the error term depends on the *best action* available at state s_{t+1}.

$$Q_{s_t,a_t} \leftarrow Q_{s_t,a_t} + \alpha[r_{t+1} + \gamma \max(Q_{s_{t+1},a_{t+1}} - Q_{s_t,a_t})] \qquad (1)$$

Other algorithms, such as SARSA, similarly rely on the measuring the Q-value of the action taken at state s_t, i.e. $Q(s_{t+1}, a_{t+1})$. Since the trials are randomized, however, the contribution of the term $Q_{s_{t+1},a}$ is going to be statistically identical, in the long term, across all states in the long-term. For convenience, in these simulations we set this term to be zero, so that the final learning equation reduces to:

$$Q_{s_t,a_t} \leftarrow Q_{s_t,a_t} + \alpha[r_{t+1} - Q_{s_t,a_t})] \tag{2}$$

Note that, under these conditions, the Q-value of an action a converges to the probability of reward $P(R_t)$ associated with each corresponding option.

The participant's policy in the PSS task was modeled as as a Gibbs softmax action selection function:

$$P(a_i) = \frac{e^{\frac{Q(a_i)}{T}}}{\sum_j e^{\frac{Q(a_j)}{T}}}. \tag{3}$$

Under this mechanism, the probability of the agent choosing a given action increases proportionally with the action's value $Q(s,a)$, divided by a parameter T, defined as the *temperature* of the system. Higher values of T inject more noise into the action selection process, causing action selection to be less deterministic.

Standard RL Model

At relatively high values of T, where the estimated utility of actions has a smaller effect on the action selection mechanism, the RL model's Choose and Avoid accuracies are approximately equal, revealing that the model has estimated the value of choosing A, when presented with any option other than B, as approximately equal to the value of *not* choosing B, when presented with any option other than A. This is desired model behavior–the model should estimate that choosing A is equal to *not* choosing B. However, as a consequence of the high level of noise in the action selection process, the model has not estimated the actual value of these two actions appropriately (relative to the value of all other options), as indicated by the low Choose and Avoid accuracies 2.

On the other hand, when the value of T is low, and the action selection process is largely dependent on the estimated Q-values of the actions associated with the current state, some alarming results occur.

Specifically, the model learns the value of of the desirable options A, C, and E well, reflected as a increasing Choose accuracy as T decreases (Figure 2, grey line). This is the expected behavior of the model – as a deterministic action selection process based on the estimated value of actions allows exploration to suggest relatively "better" options, the model quickly switches to exploiting them, learning their true values well in the process [1].

However, when the Avoid accuracy of the model is inspected, it becomes clear that the model has learned the value of some, but not all, options well. As the value of T begins decreasing, the Avoid accuracy of the model does begin increasing, as the Choose accuracy did. However, the model's Avoid accuracy actually begins to decrease (Figure 2, black line) as T continues to decrease. This indicates that for lower values of T, the model does not sufficiently explore the "bad" options (B, D, and F) during the training phase, and as a consequence, does not value them appropriately. For higher values of T, the model does explore both bad and good options approximately equally – however, it does not value neither good nor bad options appropriately. Additionally, the maximum Avoid accuracy achieved at the point of inflection (less than 70%) is much lower than the maximum Choose accuracy achieved by the model (which is when the value of T is at a minimum; approximately 90%), as well as the Choose accuracy at the point of inflection.

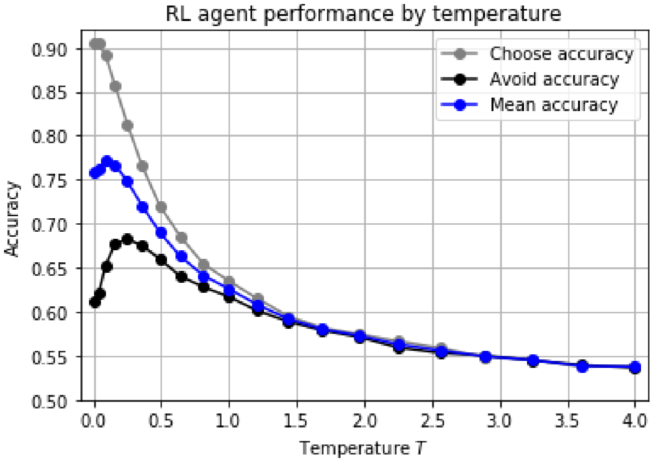


Figure 2: Performance of a canonical RL model in the PSS task for various levels of temperature T. *Grey*: Choose accuracy; *Black*: Avoid accuracy; *Blue*: Mean accuracy.

This pattern of Choose and Avoid accuracies over the range of T values tested suggests the existence of an accuracy/bias trade-off – to become more accurate on average for a given option, the model must bias its action choices to exploiting that option (in other words, the model increases the quality of its estimates of the "good" options, while becoming more uncertain about the value of the "bad" options). This effect can be seen as tendency of RL agents to converge towards overly optimistic estimates, which has been noted in the literature (Hasselt, 2010). Note that this trade-off effect does not manifest in human performance. To visualize the model's trade-off issues, the model's estimate error (defined as the bias towards choosing a given option, with respect to the probability of avoiding the same option) can be plotted as a function of its mean accuracy (Figure 3). An ideal PSS task agent would be able to obtain unbiased estimates for every level of accuracy (the vertical dashed black line). However, as made clear by Figure 3, the model's estimate error increases as mean accuracy increases–the model becomes more uncertain about its "bad" options in order to do well when presented with "good" options.

Another way in which this apparent accuracy/bias trade-off can be demonstrated is by defining the model so that it learns

[1] Although here we report the results obtained using the softmax function, the same results have been replicated with another common policy that balances exploration and exploitation, the ε-greedy policy.

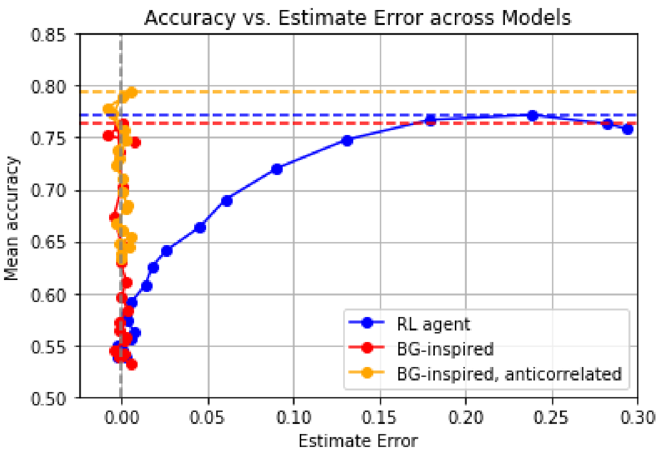


Figure 3: Mean accuracy vs. Q-value estimate errors for the three models examined in this paper. Solid lines indicate the accuracy-error trade-off curves; dotted lines indicate the maximum mean accuracy for each model. *Blue*: Standard RL model; *Red*: BG-inspired model; *Yellow*: Anti-correlated BG-inspired model.

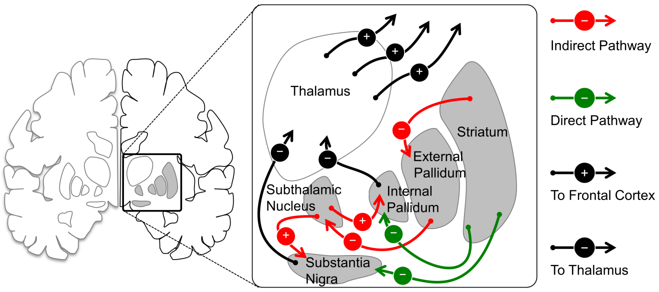


Figure 4: Overview of functional anatomy of the basal ganglia. The main basal ganglia nuclei are in grey; the arrows indicate the major projections between nuclei. The indirect pathway is shown in red, while the direct pathway is shown in green.

the value of NOT choosing actions, rather than the value of choosing actions. In other words, the model chooses to "not choose" a given option, learning the value of such in the process. In this case, as the value of T increases, Avoid accuracy decreases while Choose accuracy exhibits the inflection behavior seen in Avoid accuracy under the original model. Now, the model has learned how to navigate amongst "bad" options–it knows the value of not choosing a given option, and so it "doesn't choose" the "bad" options more often as T decreases. However, it does not learn about the value of "good" options during the learning process.

Basal Ganglia-Inspired RL Model

Reinforcement learning is known to be a reliable method of modeling the function of the basal ganglia (BG) system, a network of subcortical nuclei including the striatum, globus pallidus, substantia nigra, and subthalamic nucleus (Alexander & Crutcher, 1990).

The striatum receives input from cortical structures, and subsequently propagates the signal to later nuclei of the BG through two distinct pathways, termed the "direct" and "indirect" pathways (Smith, Beyan, Shrink, & Bolam, 1998). Of particular interest to neurological/psychological research is the fact that the striatum also receives strong dopaminergic (dopamine; DA) input from the substantia nigra *parscompacta* (SNc). Dopaminergic signaling originating from the SNc has long been thought to reflect a neural "reward" signal associated with internally-generated action and external stimuli that the organism has learned is (or expects to be) rewarding in some manner, and corresponds closely with the prediction error signal utilized in RL methods (Schultz, 2000; Schultz, Dayan, & Montague, 1997). Additionally, dopaminergic input is a defining characteristic of the "direct" and "indirect" pathways mentioned above – striatal neurons that express D1 receptors (for which DA is an excitatory ligand) form the origin of the direct pathway, while those that express D2 receptors (for which DA is an inhibitory ligand) form the origin of the indirect pathway.

For the PSS task, although the standard RL model does fairly well overall (approximately 77%), its performance does not match that of human participants, especially when considering Avoid accuracy. As the model's results demonstrate, it learns well about one set of options (either the "good" options or the "bad" options, depending on if it is learning what to choose or what to not choose, respectively), but it does not do well at valuing all options appropriately at all values of T. Ideally, the model could instead learn the values of choosing an option and not choosing the alternative simultaneously, allowing it to train once in order to appropriately value all possible options. Superficially, there seems to be an obvious compatibility between the necessity for a RL model to simultaneously estimate the value both the "chosen" and "not chosen" alternatives within a PSS trial, and dopamine's opposing influence on the direct and indirect pathways. Would a model-free RL agent with two "action pathways" perform any better than the standard RL model described above?

In order to implement the two-pathway concept, the Q-learning agent described above was modified to include an opposite set of "don't" actions $(\neg A, \neg B, \ldots, \neg F)$, which, when chosen by the agent, result in the selection of the other option that they are paired with. Thus, this agent contains a double set of actions and a stores a double set of Q-values; in this, it is reminiscent of double Q-learning (Hasselt, 2010; Van Hasselt, Guez, & Silver, 2016), an algorithm devised to address the overly-optimistic estimates of the original Q-learning algorithm (Watkins & Dayan, 1992).

The original set of actions $(A, B, \ldots, F)$ can be conceptualized as the set of actions available to be suggested by the direct pathway (restricted by actions possible within the current state), while the "antiset" can be conceptualized as the set of actions available to be suggested by the indirect pathway (also restricted by the state). So, if the current trial allows for actions *A* and *B*, and the agent selects the indirect pathway's action ¬*A*, the result is the selection of option *B*. However, if the current trial allows for actions *A* and *C*, and the agent selects ¬*A*, the result is the selection of option *C*.

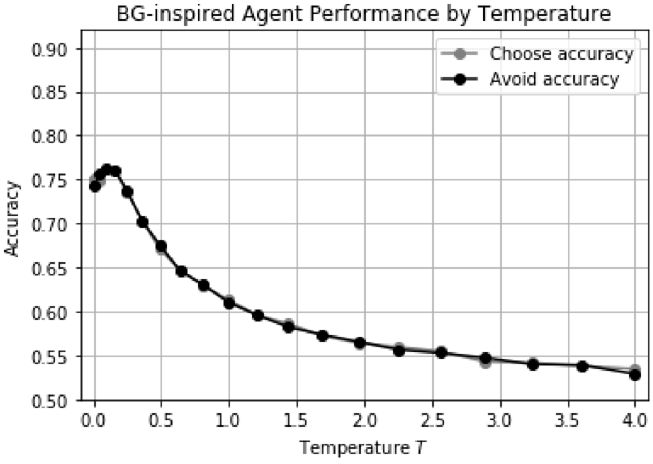


Figure 5: Performance of the BG-inspired reinforcement learning agent in the PSS task for various levels of temperature T. Note that there is no difference in the Choose *Grey* and Avoid *Black* accuracies.

Figure 5 shows that the simple addition of an "indirect pathway" to the RL model results in a marked absence of the bias observed in the standard RL model–as the value of T decreases, both choose and avoid accuracies increase commensurately. As such, the model no longer needs to "trade-off" increasing the accuracy for one class of action by becoming less confident in the valuations of the other class of action. Instead, for every choice made, it simultaneously learns both the value of the option chosen, and the value of not choosing the alternative. However, note that the maximum Choose and Avoid accuracies of the BG-plausible model do not quite achieve the same level of accuracy as the standard RL models–the uncertainty that the standard model had been attributing to the option not chosen has now been distributed across both available options. Figure 3 demonstrates that overall, the BG-plausible model (red line) achieves essentially the same level of global mean accuracy as the standard RL model (blue line), but *without* the cost of increasing estimate error.

Making the Model More Plausible

As described, this implementation of "direct" and "indirect" pathways in the RL model does well at capturing the competition between the direct and indirect pathways of the BG, and alleviates the problem of increasing estimate error with increasing accuracy. However, the BG-plausible model still performs similarly to the standard RL model in terms of global mean accuracy, indicating that although the BG-plausible model has improved ability to estimate the value of all options in the environment, this does not translate to improved fitness within the environment. However, just as the standard models were missing a crucial aspect of BG physiology (the presence of dual pathways), the BG-plausible model is missing a crucial feature of these dual pathways – the fact that DA signaling has opposite effects on the direct (excitation, mediated through D1 receptors) and indirect (inhibition, mediated through D2 receptors) pathways.

To capture this aspect of BG neurodynamics, the BG-plausible RL model was modified so that the learning algorithm results in *opposite* changes for the actions to the two pathways (an anti-correlated BG-plausible model). Specifically, if action *A* was selected and resulted in an update of it *Q*-value of size δ, then the *Q*-value of the corresponding anti-action ¬*A* would be updated by the quantity −δ. As in the biological BG, this mechanism forces the values of one set of actions to be anti-correlated to the values of the other set.

Figure 6 shows the results of simulations ran with this model. At minimum values of T, the maximum mean Choose and Avoid accuracies increase slightly (when compared to the original BG-plausible model). Figure 3 shows that, similar to the original BG-plausible model (red line), the mean accuracy of the anti-correlated BG-plausible model (yellow line) increases without a subsequent increase in estimate error. Additionally, the small increase in Choose and Avoid accuracies at minimum values of T translate into significantly better overall performance for the anti-correlated BG-plausible model.

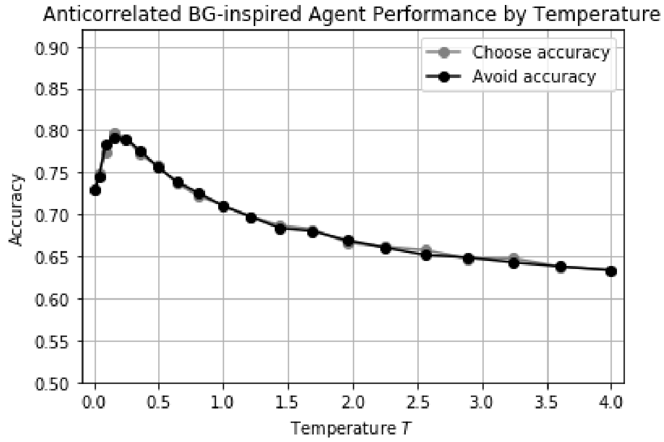


Figure 6: Performance of the anti-correlated, BG-inspired RL-learning model in the PSS task for various levels of temperature T.

However, what is most striking about the anti-correlated BG-plausible model is that at relatively large values of T (where the action selection process is noisy), the model performs much better than either the original BG-plausible model, or the standard RL models. This is an indication that the presence of the anti-correlated pathways in the second

BG-plausible model bestow a greater resistance to internal noise than the original BG-plausible model and standard RL models possess. Figure 7 more clearly demonstrates this effect: across the range of tested values of T, the mean accuracies of the original BG-plausible model are almost identical to the standard RL model. However, across the same range of T values, the anti-correlated BG-plausible model performs much better in almost every circumstance.

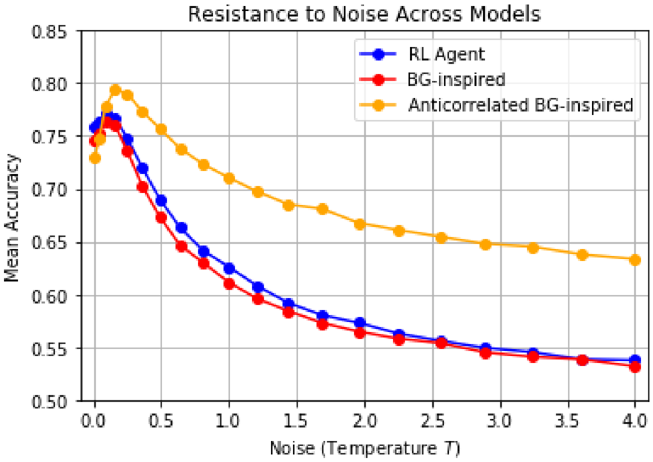


Figure 7: A direct comparison of the mean accuracy of three RL models tested in this paper. *Blue*: Standard RL model; *Red*: BG-inspired model; *Yellow*: Anti-correlated BG-inspired model.

The model does not perform as well as the standard “original” BG-inspired model only when the value of T is very close to zero, indicating almost no noise in the action selection process (an unrealistic assumption for biological systems). A similar analysis can be performed for the model’s estimate error, as seen in Figure 3. This again shows that for every tested value of T, there is little or no difference between either BG-plausible model–the presence of the two pathways allows each model to accurately estimate the value of both the most rewarding (A, C, and E) and least rewarding (B, D, and F) options. However, the standard RL model shows significant estimation biases as the lowest levels of noise, when the model’s performance is at a maximum.

Conclusions

In conclusion, the improved performance of the BG-plausible RL models implies that psychological researchers looking to model the functions of the basal ganglia could do well by taking inspiration from the characteristics of the phenomena they model, even when the modeling effort is largely theoretical. The addition of opposed action sets, representative of the well-known direct and indirect pathways within the basal ganglia, allowed the original BG-plausible model to properly estimate the value of both the “good” (relatively high probability of reward) and “bad” (relatively low probability of reward) options available in the PSS task, eliminating the bias towards “good” options displayed by the standard RL model. Furthermore, by forcing the updates of the two action sets to be anti-correlated (thereby mimicking the opposed excitatory/inhibitory effect of dopamine on the direct and indirect pathways), the model displayed a marked resistance to greater levels of noise within the selection mechanism.

Acknowledgments

This research was suported by a grant from the Office of Naval Research (ONRBAA13-003) entitled Training the Mind and Brain: Investigating Individual Differences in the Ability to Learn and Benefit Cognitively from Language Training” and by a start-up grant from the University of Washington to Andrea Stocco. Simulations and code for this study can be found at the Cognition and Cortical Dynamics Laboratory’s GitHub repository: https://github.com/UWCCDL/BGRL.

References

Alexander, G. E., & Crutcher, M. D. (1990). Functional architecture of basal ganglia circuits: Neural substrates of parallel processing. *Trends in neurosciences*, *13*, 266–271.

Frank, M. J., Moustafa, A. A., Haughey, H. M., Curran, T., & Hutchison, K. E. (2007). Genetic triple dissociation reveals multiple roles for dopamine in reinforcement learning. *Proceedings of the National Academy of Sciences*, *104*(41), 16311–16316.

Frank, M. J., Seeberger, L. C., & O’Reilly, R. C. (2004). By carrot or by stick: cognitive reinforcement learning in parkinsonism. *Science*, *306*(5703), 1940–1943.

Hasselt, H. V. (2010). Double q-learning. In *Advances in neural information processing systems* (pp. 2613–2621).

Schultz, W. (2000). Multiple reward signals in the brain. *Nature Reviews Neuroscience*, *1*, 199–207.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, *275*, 1593–1599.

Smith, Y., Beyan, M. D., Shrink, E., & Bolam, J. P. (1998). Microcircuitry of the direct and indirect pathways of the basal ganglia. *Neuroscience*, *86*, 353–388.

Stocco, A., Murray, N. L., Yamasaki, B. L., Renno, T. J., Nguyen, J., & Prat, C. S. (2017). Individual differences in the simon effect are underpinned by differences in competitive dynamics of the basal ganglia: An experimental verification and a computational model. *Cognition*, *164*, 31–45.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambrdige, MA: MIT Press.

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Aaai* (pp. 2094–2100).

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292.

Toward a Neural-Symbolic Sigma: Introducing Neural Network Learning

Paul S. Rosenbloom[a,b] **(Rosenbloom@USC.Edu), Abram Demski**[a,b] **(ADemski@ICT.USC.Edu), Volkan Ustun**[a] **(Ustun@ICT.USC.Edu)**

[a]Institute for Creative Technologies & [b]Department of Computer Science, University of Southern California
12015 Waterfront Dr., Playa Vista, CA 90094 USA

Abstract

Building on earlier work extending Sigma's mixed (symbols + probabilities) graphical band to inference in feedforward neural networks, two forms of neural network learning – target propagation and backpropagation – are introduced, bringing Sigma closer to a full neural-symbolic architecture. Adapting Sigma's reinforcement learning (RL) capability to use neural networks in policy learning then yields a hybrid form of neural RL with probabilistic action modeling.

Keywords: cognitive architecture; neural-symbolic; neural networks; learning; reinforcement learning

Introduction

One of the greatest overall challenges in cognitive modeling is developing cognitive architectures that bridge the biological and cognitive bands – spanning, respectively, 100 μs - 10 ms and 100 ms - 10 s – from Newell's (1990) analysis of the time scales of human action. The boundary between these bands sits somewhere in the region of 10-100 ms and conventionally divides symbolic from subsymbolic behavior, although the relationship between them may be in reality both subtler and more complex.

One approach to this challenge provides distinct mechanisms for the two bands that can cooperate in prescribed ways (Sun, 2016); a second seeks the emergence of cognitive mechanisms from biological ones (Eliasmith, 2013); and a third replaces components of existing cognitive architectures with neural models that yield similar results (Cho, Rosenbloom & Dolan, 1991; Jilk et al., 2008).

The approach taken in Sigma (Rosenbloom, Demski, & Ustun, 2016a) has been to generalize the notion of a biological band to that of a *graphical band* – which in Sigma is based on factor graphs, a general form of *graphical model*, plus the summary product message-passing algorithm (Kschischang, Frey & Loeliger, 2001) – that then implements the cognitive band. Recently it was discovered, however, that with one simple enhancement to this graphical band it was possible to include *feedforward neural networks*, without yet learning, among the graphs supported (Rosenbloom, Demski & Ustun, 2016b). This inspired a rethinking of Sigma's graphical band to a broader graphical notion within which factor graphs became just one particularly useful specialization and neural networks another. It also raised the possibility of a broader variation on the third approach mentioned above.

This preliminary work is extended here to weight learning in feedforward neural networks. A general form of parameter learning, via gradient descent on factor functions, was first implemented in Sigma for probability distributions (Rosenbloom et al., 2013) and then later extended to distributed vectors (Ustun et al., 2014). These are both forms of *generative learning* that learn patterns of coactivation across variables, much as in *Hebbian learning*.

Starting with this approach for distributed vectors, a variant of *target propagation* (Lee et al., 2015) has been implemented in Sigma via normal undirected (bidirectional) factor graphs, by backward propagating target values for the units' outputs, and discriminatively learning weights from differences between target and actual outputs. However, issues with this approach led us also to implement *backpropagation*, the standard discriminative approach to neural learning (Rumelhart, Hinton & Williams, 1986), that is based instead on a unidirectional forward-backward arc.

Both of these approaches reuse Sigma's message passing for backward propagation and its gradient descent for parameter learning. Backpropagation also leverages a variant of affective *appraisal* (Rosenbloom, Gratch & Ustun, 2015) to compute the error needed to initiate the backward pass. The net result is *functionally elegant* neural learning that is largely based on new combinations of existing mechanisms rather than on new modules cut from whole cloth. By extending Sigma's graphical band in this way, neural networks are potentially usable wherever factor graphs already were used, including in long-term memory, perception and learning. When combined with the earlier work on distributed vectors, a general *neural-symbolic architecture* begins to emerge that may, among other things, provide principled architectural guidance in how to combine deep learning (Goodfellow, Bengio, & Courville, 2016) with other critical cognitive capabilities.

The core result in this article thus concerns the relatively abstract yet fundamental problem of building a functionally elegant bridge from a cognitive architecture to the biological band rather than specific matches to human data. In service of this, after a review of Sigma and its earlier extension to feedforward neural networks, we will introduce neural-network learning in Sigma, followed by experiments with *classification* and *regression* networks, and the leveraging of such networks in *neural reinforcement learning*.

Sigma and Feedforward Neural Networks

Sigma is composed of two distinct architectures, one for the cognitive band and one for the graphical band. In the *cognitive architecture*, knowledge is based on *predicates* for specifying relations over typed – numeric (discrete or continuous) or symbolic – arguments; and *conditionals* for specifying patterns over combinations of predicates. *Functions* may be included in predicates to provide distributions over their arguments, and in conditionals to provide distributions over combinations of their variables.

A segment of working memory exists for each predicate, as does also a segment of long-term memory if there is a predicate function. An additional segment of long-term memory is also created for each conditional. A pattern in a conditional may be a *condition*, which acts like a rule condition by matching to working memory; an *action*, which acts like a rule action by changing working memory; or a *condact*, which combines the effects of a condition and an action to yield bidirectional constraints on the contents of working memory. Procedural memory is largely based on conditions and actions – i.e., rules – and declarative memory on condacts. Decisions are made by selecting values from predicate arguments based on distributions over them.

Figure 1, for example, displays two conditionals – each effectively a (non-symbolic) rule with an associated weight function – that together implement the two-layer neural network in Figure 2. All argument types here are discrete numeric, but with three elements for `Input` and two each for `Hidden` and `Output`. The single argument (`arg`) in each pattern is specified here by variables – *`i`*, *`h`*, and *`o`* – with the function in each conditional being defined over its pair of variables. The **`s`** in the conditionals' actions denotes that a sigmoid/logistic function is to be applied before working memory is changed (other possibilities include `r` for RELU, `t` for tanh, `e` for exponential, and `x` for softmax). The one modification required to make this work in Sigma was extending to these functions its existing ability to include non-linear transformations in conditional patterns.

This particular way of encoding a neural network in Sigma involves one conditional per layer, with the structure of the layers implicit in the argument types and conditional functions. Although it is also possible to encode such networks via one conditional per link, with one element per type and a single weight per function, here the focus is on the more concise representation illustrated in Figure 1.

```
CONDITIONAL C-Layer1
  Conditions: (Input arg:i)
  Actions: (Hidden s arg:h)
  Function<i,h>: .2:<0,0>, .7:<0,1>, …

CONDITIONAL C-Layer2
  Conditions: (Hidden arg:h)
  Actions: (Output s arg:o)
  Function<h,o>: 1.1:<0,0>, 3.1:<0,1>, …
```

Figure 1: Conditionals for the network in Figure 2.

Sigma's compiler converts knowledge specified in its cognitive architecture into undirected bipartite graphs of variable and factor nodes – essentially factor graphs – in the *graphical architecture*. Functions are stored in factor nodes. Processing occurs via message passing – essentially the summary product algorithm – with each message encoding a distribution over the variables in the variable node on the link. Incoming messages are pointwise multiplied together at nodes, along with the node function at factor nodes, and then variables not needed in outgoing messages are summarized out, typically via either integral or maximum. For conditions and actions, messages are passed in only one direction, from working memory for conditions and towards working memory for actions, whereas condact message passing is bidirectional. Learning occurs by gradient descent at factor nodes, with gradients based on messages arriving from adjacent variable nodes.

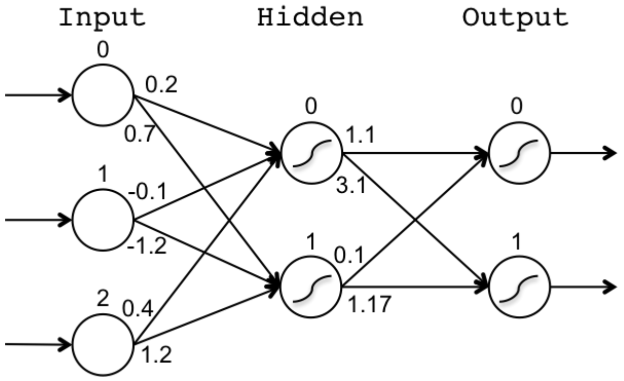


Figure 2: Two-layer neural network (adapted from http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture13.html).

Target Propagation

With target propagation, *targets* – that is, desired values – rather than errors are propagated backward over the network, with errors then computed locally at factor nodes based on subtracting computed outputs from desired outputs. To support this, the unidirectional rules in Figure 1 are converted to bidirectional constraints, with conditions and actions becoming condacts, as shown in Figure 3.

```
CONDITIONAL C-Layer1-TP
  Condacts: (Input arg:i)
            (Hidden s arg:h)
  Neural:h
  Function<i,h>: <Random in [-.1,.1]>

CONDITIONAL C-Layer2-TP
  Condacts: (Hidden arg:h)
            (Output s arg:o)
  Neural:o
  Function<h,o>: <Random in [-.1,.1]>
```

Figure 3: Target propagation conditionals for two-layer weight learning.

The weights in the functions are initialized randomly, and then learned online from training examples. The `Neural` attribute in the conditionals specifies that local discriminative learning is to be used here, with the gradient based on subtracting the output message for the specified variable (i.e., its computed value) from its input message (i.e., its desired/target value). Learning from this error-based gradient then follows the simplified additive form earlier developed for distributed vectors rather than the more complex form originally developed for distributions.

Starting with the targets for the network's output units, computing the targets and gradients for interior units leverages the bidirectionality of condacts to send messages backward in the graph. However, in contrast to backward messages in normal factor graphs, proper processing of these messages requires that the functions be inverted at factor nodes. This is straightforward for the logistic function, as its inverse is simply the *logit* function: *log*(*x*/[1-*x*]). However, this does raise a deeper problem, in that the domain of this function is (0,1), whereas there is no guarantee that a target – particularly one generated inside the network – will fall in this range. To work around this, backward messages at these nodes are truncated to [ε,1-ε].

A second problem arises at the factor nodes where the learned weight functions must be inverted. Rather than attempting to do this analytically, inversion is approximated empirically by gradient descent over the node's backward output. In particular, the product of the output error and the weight function is multiplied by a pseudo-learning rate (.05) and then added to the forward input message to yield the backward output message.

Aside from the nonstandard approach to computing backward messages, the result is a form of target propagation that otherwise fits cleanly into normal factor graphs, including respecting the constraint that all messages over a link are distributions over the link's variables.

Backpropagation

With backpropagation, a difference is computed only once, for the network's output units, and propagated backward successively from there. Sigma already supports an architectural *desirability* appraisal that calculates differences between distributions over goals and their associated states, and which is used in both guiding problem solving and directing attention. What is needed for backpropagation is an analogous *correctness* appraisal that operates over point values rather than distributions. The error is then simply the difference between the `output` predicate's specified target/goal and its computed value/state.

Unlike with target propagation, however, the error cannot just be propagated backward over a bidirectional network, as that would violate the constraint that all of the messages on a link should be distributions over the values of the link's variables. In the forward direction the messages are (unnormalized) distributions – effectively *activations* – over variables, each of which corresponds to the set of units at one layer of the network. Sending errors backward over these same links would be invalidly inhomogeneous.

Instead, what has been done is to complement each unidirectional forward network with a unidirectional backward network over which errors are sent, with the appraisal at the end of the forward network serving as the nexus connecting it to the backward network. Figure 4 shows an abstract graph for how this all works.

The left (green) path is the forward one, stretching from the perceptual buffer for the `Input` predicate up through two layers of weights to the `Output` predicate. The squares are factor nodes, where the weight functions are stored, whereas the circles are variable nodes. The two sigmoid transformations occur at additional factor nodes that are abstracted away in this figure. The output of the forward path joins with the target values for the outputs at the appraisal of correctness.

Figure 5 shows the forward conditionals for this. They are like those in Figure 1 in having conditions and actions, and like those in Figure 3 in using random initial weights, but they replace the `Neural` attribute with the `Vector` attribute to signal that distributed-vector gradients should be used in learning without target propagation's gradient-based approach to backward message passing.

The right (red) path in Figure 4 is the backward one. It includes its own factor and variable nodes, but with crucial linkages added to the forward path. The sigmoid nodes are also abstracted away here, but in the network compute the derivative of the logistic – *x*(1-*x*) – rather than its inverse.

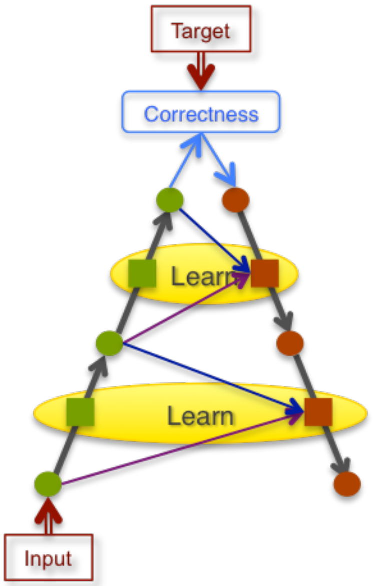


Figure 4: Structure of backpropagation for a two-layer network.

```
CONDITIONAL C-Layer1
  Conditions: (Input arg:i)
  Actions: (Hidden s arg:h)
  Function<i,h>: <Random in [-.1,.1]>
  Vector: T

CONDITIONAL C-Layer2
  Conditions: (Hidden arg:h)
  Actions: (Output s arg:o)
  Function<h,o>: <Random in [-.1,.1]>
  Vector: T
```

Figure 5: Forward conditionals for two-layer backpropagation network.

Figure 6 shows the corresponding backward conditionals. In general, backward predicates – such as `Hidden*Back` – are introduced that correspond to the forward ones, and conditions are swapped with actions. However, there are slight variations for the first and last layers. In the first layer, the backward propagation of information can stop at the weight function, rather than going all of the way back to the input units, so there is no action in `C-Layer1-B`. In the last layer, backward propagation starts with the appraisal for the `Output` predicate – `Output*Error` – so the error is used directly rather than a new backward predicate.

Learning occurs based on messages arriving at the factor nodes in the backward path, but the functions in these nodes are *tied* to those in the forward path – shown by the yellow ellipses in Figure 4 – so that any changes made to the former are directly reflected in the latter. This is enabled by the `Forward-Conditional` attributes in Figure 6, which specify the corresponding forward conditionals.

```
CONDITIONAL C-Layer1-B
  Conditions: (Hidden*Back arg:h)
              (Hidden s arg:h)
  Forward-Conditional: C-Layer1
  Exclude-Forward-Backward T
  Vector: T

CONDITIONAL C-Layer2-B
  Conditions: (Output*Error arg:o)
              (Output s arg:o)
  Actions: (Hidden*Back arg:h)
  Forward-Conditional: C-Layer2
  Exclude-Forward-Backward T
  Vector: T
```

Figure 6: Backward conditionals for two-layer backpropagation network.

In its simplest form, the gradient in backpropagation is the product of: (1) the learning rate; (2) the forward message at the weight function; (3) the output difference; and (4) the derivative of the sigmoid function. The forward message, as shown by the upward slanting (purple) unidirectional links from the forward path, is added automatically to the graph by the conditional compiler given the `Forward-Conditional` attribute. The output difference comes from above in the figure, as derived from the first condition in a backward conditional. The computation of the derivative of the sigmoid, although abstracted away in the graph, arrives at the backward factor node via the downward slanting (blue) links from the forward path, based on the second condition in a backward conditional. As with target propagation, the resulting gradient is handled in the simple additive manner developed earlier for distributed vectors.

In contrast to target propagation, here the backward message out of the factor node – that is, the propagated output difference – is computed simply by message/function multiplication and summarization, as is standard in factor graphs. There is one important caveat though. As specified by the `Exclude-Forward-Backward` attribute, the purple message from the forward path is not included in this product, so the backward output is just the product of the output difference, the derivative of the sigmoid function and the weight function in the node. This exception to the normal rule is motivated by backpropagation, but justified independently in factor graph terms by the fact that a message coming into a node on a bidirectional link should not be used in computing the reverse message on the same link. Here there are two unidirectional links, but they effectively comprise a single logical bidirectional path.

Basic Experiments

Regression and classification problems provide two forms of standard benchmarks for learning with neural networks. The network in Figure 2, for example, defines a regression problem, where two functions are to be learned from the inputs, one for each output. Small experiments with this network, starting with uniform weights, do show that both forms of propagation can learn weights in Sigma that yield outputs like those generated by the network in the figure. But what is really needed for verification is an investigation into how Sigma compares with standard packages.

For this, we have compared Sigma with PyBrain, a Python machine learning library (Schaul et al., 2010), via three standard machine learning datasets: (1) *Iris* – https://archive.ics.uci.edu/ml/datasets/Iris – a classification problem with 3 classes; (2) *Robot Arm* – http://mldata.org/repository/data/viewslug/uci-20070111-kin8nm/ – a regression problem that learns to predict the end effector position for an 8 link robot arm; and (3) *MNIST* – http://yann.lecun.com/exdb/mnist/ – a classification problem over the digits 0-9, based on 28x28 pixel images. Table 1 shows the static information for these datasets.

Table 1: Input, hidden and output units; training and test instances; learning rate; and training epochs.

	I	H	O	Train	Test	λ	Ep.
Iris	4	10	3	138	12	.1	100
Robot	8	100	1	6530	838	.01	100
MNIST	784	30	10	10K[1]	10K	.01	50

Our experiments so far with target propagation have not yet yielded reasonable results on these datasets, most likely because of the truncation required for the logit. So Table 2 only shows backpropagation results. The first and most critical result is that Sigma's accuracy is indistinguishable from that produced by PyBrain with the same settings. Second, Sigma is slower, by up to a factor of ~100. Although a slowdown with a general architecture is not surprising, it should actually be possible to close this gap with a more efficient message representation plus SIMD (as in PyBrain) and GPU hardware. It is also worth note though that these results are at most a factor of 2 slower than the human cognitive cycle time of ~50 ms, a factor that can be relevant when concerned with real-time cognitive models.

Table 2: Accuracy (% correct for Iris and MNIST, RMSE for Robot Arm); seconds per epoch; and ms per decision.

	Py A.	Σ A.	Py s/Ep.	Σ s/Ep	Σ ms/D
Iris	.917	.917	.082	.215	2
Robot	.173	.173	3.51	54.33	8
MNIST	.867	.867	9.8	1029.2	103

Neural Reinforcement Learning

Figure 7 shows a simple 1D grid in which reinforcement learning (RL) was initially explored in Sigma (Rosenbloom, 2012). The agent can move left or right in locations 1 through 6, with locations 0 and 7 being forbidden boundary

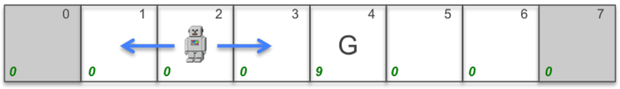


Figure 7: 1D grid with agent, goal location and rewards.

[1] Only the first 10K training examples are used for MNIST.

regions. When the goal (location 4, with a reward of 9) is reached the trial halts. This has since been extended to larger 2D grids and to other tasks, but its simplicity provides a good starting point for exploring neural RL.

Like neural learning, Sigma's RL capability is not a distinct architectural module. Instead it is deconstructed in terms of a set of conditionals plus learning of distributions over rewards, state utilities and action policies. Neural RL in Sigma is much like this – Figure 8 – with similar conditionals and learning of the same quantities. For example, the rightward arrow at the bottom of the figure indicates a conditional with a transition function (i.e., an action model) that predicts the location resulting from applying an operator, while the leftward arrow(s) at the top of the figure show the discounted backward propagation of the sum of the projected future utility and the reward for the predicted next state to the projected future utility of the current state and the policy for the current state and reward.

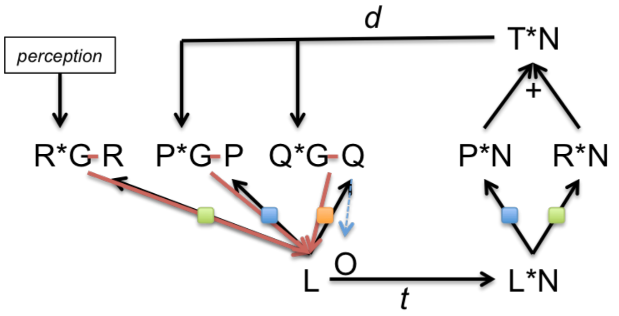


Figure 8: Diagram of neural RL in Sigma.

Still, there are several key differences implied by the top-level shift from distributional to neural learning that go beyond simply which form of learning is used. First, because backpropagation is used, there is a forward-backward arc of unidirectional conditionals (with functions) – as in Figure 4 – for each quantity to be learned, rather than a functional predicate or bidirectional conditional. In Figure 8, the forward paths are the upward black arrows from the location (L) to the reward (R), the projected future utility (P) and the policy (Q), whereas the backward paths are the downward red arrows back to L from correctness calculations (such as R*G - R). The tied functions are shown as path-spanning squares. A single-layer network – i.e., logistic regression – is used here due to the simplicity of the problem, but this can easily be extended to multiple layers.

Second, because neural learning structurally distinguishes input from output in the network, implying an asymmetry that need not exist in distributional learning, the arguments for these quantities must appear in different predicates. Semantically, distributions may be symmetric, as when they are joint, or they may be asymmetric, as when conditional; but both can appear identically in a graphical model. In distributional RL, conditional distributions are learned, but single symmetric predicates – such as `Reward(x:`*x*`, value:`*r*`)` – are used. For neural RL this must be split in two, to yield `Location(x:`*x*`)` and `Reward(value:`*r*`)`, as shown by L and R in the figure.

Third, instead of learning a distribution over all possible output values, with sums (of rewards and projected future utilities) and products (by discount factors) computed by *affine transforms*, in neural learning a single value is learned, with sums resulting from adding the effects of multiple actions (top-right of Figure 8) and products from multiplying the effects of multiple conditions (*d* at top of the figure). For example, in the distributional case the domain of the `value` argument for `Reward` includes all possible rewards, and the function over this and `x` is the conditional distribution over the value given the location. Summing two such values occurs by *translating* the distribution, and discounting by *scaling* it. In the neural case, there is instead only one domain element in the `value` argument, with the function over this element simply the learned reward, and computations on this occurring during pattern combination. Thus, not only is just a point value learned in the neural case, that value is implicit in the range of the learned function rather than explicit in the domain of the function, and computation with it occurs in a rather different manner.

Fourth, because with distributional learning the arguments all exist within one predicate, potentially providing a full cross-product among their elements, a table is effectively acquired from which multiple answers can be extracted simultaneously via conditions with appropriate constants and variables. With neural learning, extracting each answer requires either running the network once for each input, or including a distinct forward network for each possible input (but with tied functions across them). This latter approach has been used in neural RL to access in parallel the rewards and projected utilities of the current state and the predicted next state. In Figure 8, the separate paths for the next state are shown to the right, with function coloring indicating tying to the corresponding functions in the current path.

Despite these differences, Figure 9 shows that the point values learned for the neural policy are still appropriate, with rightward movement preferred when to the left of the goal location and leftward movement when to its right. This policy is averaged over ten runs of 500 trials each, with each trial starting at location 1 or 6, and all ending at location 4.

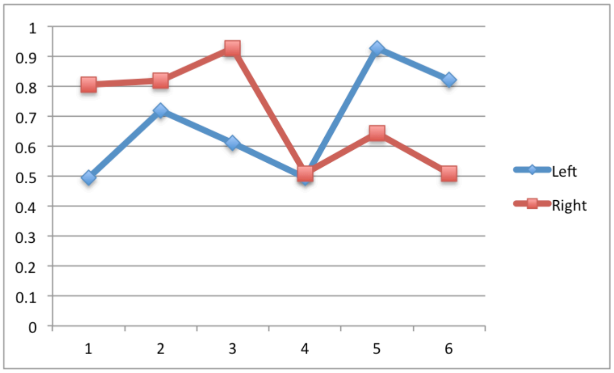


Figure 9: Policy (Q) function learned via neural RL.

Two other things are also worth noting from this simple neural RL experiment, which included an equivalent trial sequence for the distributional version. First, The neural version was approximately five times faster than the distributional one in terms of time per decision – 12 versus 62 ms/D – largely due to the smaller functions and messages possible when not using full distributions. Second, both versions learn models of their actions distributionally – in terms of a conditional probability distribution over the next location given the current location and action – while engaged in RL. The neural case thus illustrates the ease with which neural and distributional learning can combine in Sigma across subproblems in the same overall problem.

Conclusion

Building upon Sigma's feedforward neural-network inference capability and its distributed vector learning capability, two forms of neural network learning – target propagation and backpropagation – have been implemented via a combination of extensions to existing architectural mechanisms and knowledge expressed as predicates and conditionals. In both variations, the backward propagation of information occurs through message passing in Sigma's graphical architecture rather than via special purpose mechanisms; and in backpropagation, the initial error computation occurs via a form of appraisal.

For backpropagation we get results of comparable quality to, but slower than, a standard package; and we see the possibility of combining it with other capabilities, such as reinforcement learning and probabilistic action modeling. Neural network inference and learning are thus now becoming pervasively available within Sigma's central cognitive cycle, a major step toward a full neural-symbolic architecture that is based on a functionally elegant bridge to the biological band. In addition, although somewhat of a side point here, these extensions enable Sigma to perform discriminative learning over point values in general, whether for use in neural learning or not, to complement the existing ability of generative learning over full distributions.

Future work includes extension to the full power of deep learning and the handling of temporal sequences via techniques such as LSTMs (Hochreiter & Schmidhuber, 1997). Also planned is further optimization and integrations of neural networks with other critical cognitive capabilities.

Acknowledgments

The work described in this article was sponsored by the U.S. Army. Statements and opinions expressed may not reflect the position or policy of the United States Government, and no official endorsement should be inferred. We would also like to thank Sruthi Madapoosi Ravi for significant help with the regression and classification experiments.

References

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).

Eliasmith, C. (2013). *How to Build a Brain.* Oxford: Oxford University Press.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Jilk, D. J., Lebiere, C., O'Reilly, R. C. and Anderson, J. R. (2008). SAL: an explicitly pluralistic cognitive architecture. *Journal of Experimental & Theoretical Artificial Intelligence*, *20*, 197-218

Kschischang, F. R., Frey, B. J. & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*. *47*, 498-519.

Lee, D.-H., Zhang, S., Fischer, A., & Bengio, Y. (2015). Difference target propagation. In A. Appice, P. P. Rodrigues, V. S. Costa, C. Soares, J. Gama & A. Jorge (Eds.), *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015*. Switzerland: Springer International Publishing.

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

Rosenbloom, P. S. (2012). Deconstructing reinforcement learning in Sigma. *Proceedings of the 5th Conference on Artificial General Intelligence* (pp. 262-271).

Rosenbloom, P. S., Demski, A., Han, T., & Ustun, V. (2013). Learning via gradient descent in Sigma. *Proceedings of the 12th International Conference on Cognitive Modeling* (pp. 35-40).

Rosenbloom, P. S., Demski, A. & Ustun, V. (2016a). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, *7*, 1-103.

Rosenbloom, P. S., Demski, A. & Ustun, V. (2016b). Rethinking Sigma's graphical architecture: An extension to neural networks. *Proceedings of the 9th Conference on Artificial General Intelligence* (pp. 84-94).

Rosenbloom, P. S., Gratch, J. & Ustun, V. (2015). Towards emotion in Sigma: From appraisal to attention. *Proceedings of the 8th Conference on Artificial General Intelligence* (pp. 142-151).

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533-536.

Schaul, T., Bayer, J., Wierstra, D., Yi, S., Felder, M., Sehnke, F., Rückstieß, T. & Schmidhuber, J. (2010). PyBrain. Journal of Machine Learning Research, *11*, 743-746.

Sun, R. (2016). *Anatomy of the Mind: Exploring Psychological Mechanisms and Processes with the Clarion Cognitive Architecture*. New York, NY: Oxford University Press.

Ustun, V., Rosenbloom, P. S., Sagae, K., & Demski, A. (2014). Distributed vector representations of words in the Sigma cognitive architecture. *Proceedings of the 7th Annual Conference on Artificial General Intelligence* (pp. 196-207).

A causal role for right frontopolar cortex in directed, but not random, exploration

Wojciech Zajkowski (wzajkowski@st.swps.edu.pl)
University of Social Sciences and Humanities
Warsaw, Poland

Malgorzata Kossut
Departament of Psychology
University of Social Sciences and Humanities
Warsaw, Poland

Robert C. Wilson (bob@arizona.edu)
Department of Psychology and Cognitive Science Program
University of Arizona
Tucson AZ, USA

Abstract

The explore-exploit dilemma occurs anytime we must choose between exploring unknown options for information and exploiting known resources for reward. Previous work suggests that people use two different strategies to solve the explore-exploit dilemma: directed exploration, driven by information seeking, and random exploration, driven by decision noise. Here, we show that these two strategies rely on different neural systems. Using transcranial magnetic stimulation to inhibit the right frontopolar cortex, we were able to selectively inhibit directed exploration while leaving random exploration intact. This suggests a causal role for right frontopolar cortex in directed, but not random, exploration and that directed and random exploration rely on (at least partially) dissociable neural systems.

Keywords: Explore-exploit, decision making, transcranial magnetic stimulation, frontal pole

Introduction

In an uncertain world, adaptive behavior requires us to carefully balance the exploration of new opportunities with the exploitation of known resources. Finding the optimal balance between exploration and exploitation is a hard computational problem and there is considerable interest in how humans and animals strike this balance in practice (Hills et al.,2015). Recent work has suggested that humans use two distinct strategies to solve the explore-exploit dilemma: directed exploration, based on information seeking, and random exploration, based on decision noise (Wilson, Geana, White, Ludvig, & Cohen,2014). Even though both of these strategies serve the same purpose, i.e. balancing exploration and exploitation, it is likely they rely on different cognitive mechanisms. Directed exploration is driven by information and is thought to be computationally complex. On the other hand, random exploration can be implemented in a simpler fashion by using neural or environmental noise to randomize choice.

Of particular interest is the right frontopolar cortex (RFPC) – an area that has been associated with a number of functions, such as tracking alternate options (Boorman, Behrens, Woolrich, & Rushworth,2009), strategies (Domenech & Koechlin,2015) and goals (Pollmann,2016) that may be important for exploration. In addition, a number of studies have implicated the frontal pole in exploration itself (Badre, Doll, Long, & Frank,2012;Daw, O'Doherty, Dayan, Seymour, & Dolan,2006), although importantly, how exploration is defined varies from paper to paper. In one line of work, exploration is defined as information seeking. Understood this way, exploration correlates with FPC activity measured via fMRI (Badre et al.,2012), suggesting a role for FPC in directed exploration. However, in another line of work, exploration is operationalized differently, as choosing the low value option, not the most informative. Such a measure of exploration is more consistent with random exploration where decision noise drives the sampling of low value options by chance. Defined in this way, exploratory choice correlates with FPC activation (Daw et al.,2006) and stimulation and inhibition of RFPC with direct current (tDCS) can increase and decrease the frequency with which such exploratory choices occur (Raja Beharelle, Polania, Hare, & Ruff,2015).

Taken together, these two sets of findings suggest that lateral FPC plays a crucial role in both directed and random exploration. However, we believe that such a conclusion is premature because of a subtle confound that arises between reward and information in most explore-exploit tasks. This confound arises because participants only gain information from the options they choose, yet are incentivized to choose more rewarding options. Thus, over many trials, participants gain more information about more rewarding options such that the two ways of defining exploration, choosing high information or low reward options, become confounded (Wilson et al.,2014). This makes it impossible to tell whether the link between FPC and exploration is specific to directed exploration, random exploration, or whether it is general to both.

To distinguish these interpretations and investigate the causal role of RFPC in directed and random exploration, we used continuous theta-burst TMS (cTBS) (Huang, Edwards, Rounis, Bhatia, & Rothwell,2005) to selectively inhibit RFPC in participants performing the 'Horizon Task', an explore-exploit task specifically designed to separate directed and random exploration (Wilson et al.,2014). Using this task we find that RFPC inhibition selectively inhibits directed exploration while leaving random exploration intact.

Methods

Participants

31 healthy right-handed, adult volunteers (19 female, 12 male; ages 19-32) took part in the study. 6 participants were excluded from the analysis due to chance-level performance or for failure to to return for the second session leaving 25

participants (13 female, 12 male, ages 19-32) for the main analysis. All participants were informed about potential risks connected to TMS and signed a written consent. The study was approved by University of Social Sciences and Humanities ethics committee.

TMS procedure

All TMS was delivered in line with established safety guidelines (Rossi, Hallett, Rossini, Pascual-Leone, & Safety of TMS Consensus Group,2009). There were two experimental TMS sessions (targeting RFPC and vertex, as a control) and a preceding MRI session in which a T1 structural image was acquired in order to target frontal pole. During the TMS sessions, resting motor thresholds were obtained first and then the cTBS procedure took place. This involved 40 second of stimulation at 50Hz at 80% resting motor threshold, a protocol that is thought to decrease cortical excitability for up to 50 minutes (Wischnewski & Schutter,2015). Participants began the main task immediately after stimulation. The two experimental sessions were performed with an intersession interval of at least 5 days. All sessions took place at Nencki Institute of Experimental Biology in Warsaw. Based on previous fMRI work showing a link between FPC and exploration (Daw et al.,2006;Badre et al.,2012), RFPC stimulation was targeted at [x,y,z] = [35,50,15] in MNI (Montreal Neurological Institute) space. Vertex corresponded to the Cz position of the 10-20 EEG system.

Behavioral task

We used our previously published ‘Horizon Task’ (Figure 1) to measure the effects of TMS stimulation of RFPC on directed and random exploration. In this task, participants play a set of games in which they make choices between two slot machines (one-armed bandits) that pay out rewards from different Gaussian distributions. To maximize their rewards in each game, participants need to exploit the slot machine with the highest mean, but they cannot identify this best option without exploring both options first.

The Horizon Task has two key manipulations that allow us to measure directed and random exploration. The first manipulation is the horizon itself, i.e. the number of decisions remaining in each game. The idea behind this manipulation is that when the horizon is long (6 trials), participants should explore more frequently, because any information they acquire from exploring can be used to make better choices later on. In contrast, when the horizon is short (1 trial), participants should exploit the option they believe to be best. Thus, this task allows us to quantify directed and random exploration as changes in information seeking and behavioral variability that occur with horizon.

The second manipulation is the amount of information participants have about each option *before* making their first choice. This information manipulation is achieved by using four forced-choice trials, in which participants are told which option to pick, at the start of each game. We use these forced-choice trials to setup one of two information conditions: an unequal, or [1 3], condition, in which participants see 1 play from one option and 3 plays from the other option, and an unequal, or [2 2], condition, in which participants see two outcomes from both options. The two information conditions allow us to quantify directed and random exploration in a model-free manner (Figure 1). In particular, directed exploration, which involves information seeking, can be quantified as the probability of choosing the high information option, p(high info) in the [1 3] condition, while random exploration, which involves decision noise, can be quantified as the probability of making a mistake, or choosing the low mean reward option, p(low mean), in the [2 2] condition. Crucially, if p(high info) and p(low mean) increase with horizon, then we infer that participants are using directed and random exploration.

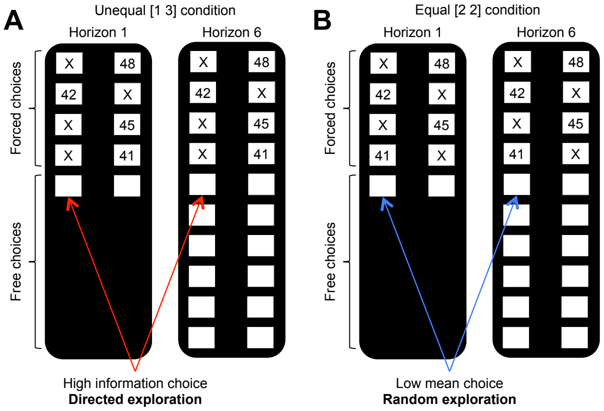


Figure 1: The Horizon Task. Participants make a series of decisions between two one-armed bandits that pay out probabilistic rewards with unknown means. At the start of each game, ‘forced-choice’ trials give participants partial information about the mean of each option. We use the forced-choice trials to set up one of two information conditions: (A) an unequal (or [1 3]) condition in which participants see 1 play from one option and 3 plays from the other and (B) an equal (or [2 2]) condition in which participants see 2 plays from both options. A model-free measure of directed exploration is then defined as the change in information seeking with horizon in the unequal condition (A). Likewise a model-free measure of random exploration is defined as the change choosing the low mean option in the equal condition (B).

Model-based analysis

While the model-free analyses are intuitive, the model-free statistics, p(high info) and p(low mean), are not pure reflections of information seeking and behavioral variability and could be influenced by other factors such as spatial bias and learning. To account for these possibilities we performed a model-based analysis using a model that extends our earlier work (Wilson et al.,2014;Somerville et al.,2017). In this model, the level of directed and random exploration is captured by two parameters: an information bonus for directed exploration, and decision noise for random exploration. In addition the model includes terms for the spatial bias and to describe learning. The model naturally decomposes into a learning component and a decision component and we consider each of these components in turn.

Learning component The learning component of the model assumes that participants use a Kalman filter (Kalman,1960) to learn a value for the mean reward of each option. In particular, we assume that participants use a generative model of the task in which the rewards from each bandit, r_t, are generated from Gaussian distribution with a fixed standard deviation, σ_r, and a mean, m_t^i, that is different for each bandit and can vary over time. The time dependence of the mean is determined by a Gaussian random walk with mean 0 and standard deviation σ_d. Note that this generative model, assumed by the Kalman filter, is slightly different to the true generative model used in the Horizon Task, which assumes that the mean of each bandit is constant over time, i.e. $\sigma_d = 0$. This mismatch between the assumed and actual generative models, is quite deliberate and allows us to account for the suboptimal learning of the subjects. In particular, this mismatch introduces the possibility of a recency bias (when $\sigma_d > 0$) whereby more recent rewards are over-weighted in the computation of R_t^i.

The actual equations of the Kalman filter model are straightforward. The model keeps track of an estimate of both the mean reward, R_t^i, of each option, i, and the uncertainty in that estimate, σ_t^i. When option i is played on trial t, these two parameters update according to

$$\begin{aligned} R_{t+1}^i &= R_t^i + \frac{(\sigma_{t+1}^i)^2}{\sigma_r^2}(r_t - R_t^i) \\ \frac{1}{(\sigma_{t+1}^i)^2} &= \frac{1}{(\sigma_t^i)^2 + \sigma_d^2} + \frac{1}{\sigma_r^2} \end{aligned} \tag{1}$$

When option i is not played on trial t we assume that the estimate of the mean stays the same, but that the uncertainty in this estimate grows as the generative model assumes the mean drifts over time. Thus for unchosen option j we have

$$R_{t+1}^j = R_t^j \qquad \text{and} \qquad (\sigma_{t+1}^j)^2 = (\sigma_t^j)^2 + \sigma_d^2 \tag{2}$$

When the option is played, the update equation for R_t^i is essentially just a 'delta rule' (Rescorla, Wagner, et al.,1972), with the estimate of the mean being updated in proportion to the prediction error, $r_t - R_t^i$. This relationship to the reinforcement learning literature is made more clear by rewriting the learning equations in terms of the time varying learning rate, $\alpha_t^i = (\sigma_{t+1}^i)^2/\sigma_r^2$ Written in terms of this learning rate, equations 1 become

$$R_{t+1}^i = R_t^i + \alpha_t^i(r_t - R_t^i) \quad \text{and} \quad \frac{1}{\alpha_t^i} = \frac{1}{\alpha_{t-1}^i + \alpha_d} + 1 \tag{3}$$

where $\alpha_d = \sigma_d^2/\sigma_r^2$. The learning model has four free parameters: the noise variance, σ_r^2, the drift variance, σ_d^2, and the initial values of the estimated reward, R_0, and uncertainty in that variance estimate, σ_0^2. In practice, only three of these parameters are identifiable from behavioral data, and we will find it useful to reparameterize the learning model in terms of R_0 and an initial, α_0, and asymptotic, α_∞, learning rate. In particular, the initial value of the learning rate relates to σ_0 and σ_r as $\alpha_0 = \sigma_0^2/\sigma_r^2$, while the asymptotic value of the learning rate, which corresponds to the steady state value of α_t^i if option i is played forever, relates to α_d (and hence σ_d and σ_r) as

$$\alpha_\infty = \frac{1}{2}\left(-\alpha_d + \sqrt{\alpha_d^2 + 4\alpha_d}\right) \tag{4}$$

Decision component Once the payoffs of each option, R_t^i, have been estimated from the outcomes of the forced-choice trials, the model makes a decision using a simple logistic choice rule:

$$p(\text{choose right}) = \frac{1}{1 + \exp\left(\frac{\Delta R + A\Delta I + B}{\sigma}\right)} \tag{5}$$

where ΔR ($= R_t^{left} - R_t^{right}$) is the difference in expected reward between left and right options and ΔI is the difference in information between left and right options (which we define as +1 when left is more informative, -1 when right is more informative, and 0 when both options convey equal information in the [2 2] condition). The three free parameters of the decision process are: the information bonus, A, the spatial bias, B, and the decision noise σ. We assume that these three decision parameters can take on different values in the different horizon and uncertainty conditions (with the proviso that A is undefined in the [2 2] information condition since $\Delta I = 0$). Thus the decision component of the model has 10 free parameters (A in the two horizon conditions, and B and σ in the 4 horizon x uncertainty conditions). Directed exploration is then quantified as the change in information bonus with horizon, while random exploration is quantified as the change in decision noise with horizon.

Model Fitting

Hierarchical Bayesian Model Between the learning and decision components of the model, each subject's behavior is described by 13 free parameters, all of which are allowed to vary between TMS conditions. These parameters are: the initial mean, R_0, the initial learning rate, α_0, the asymptotic learning rate, α_∞, the information bonus, A, in both horizon conditions, the spatial bias, B, in the four horizon x uncertainty conditions, and the decision noise, σ, in the four horizon x uncertainty conditions (Table 1, Figure 2).

We fit each of the free parameters to the behavior of each subject using a hierarchical Bayesian approach (Lee & Wagenmakers,2014). In this approach to model fitting, each parameter for each subject is assumed to be sampled from a group-level prior distribution whose parameters, the so-called 'hyperparameters', are estimated using a Markov Chain Monte Carlo (MCMC) sampling procedure.

Parameter	Prior	Hyperparameters	Hyperprior
prior mean, $R_0^{\tau s}$	$R_0^{\tau s} \sim \text{Gaussian}(\mu_{R_0}^{\tau}, \sigma_{R_0}^{\tau})$	$\theta_{R_0}^{\tau} = (\mu_{R_0}^{\tau}, \sigma_{R_0}^{\tau})$	$\mu_{R_0}^{\tau} \sim \text{Gaussian}(50, 14)$ $\sigma_{R_0}^{\tau} \sim \text{Gamma}(1, 0.001)$
initial learning rate, $\alpha_0^{\tau s}$	$\alpha_0^{\tau s} \sim \text{Beta}(a_{\alpha_0}^{\tau}, b_{\alpha_0}^{\tau})$	$\theta_{\alpha_0}^{\tau} = (a_{\alpha_0}^{\tau}, b_{\alpha_0}^{\tau})$	$a_{\alpha_0}^{\tau} \sim \text{Uniform}(0.1, 10)$ $b_{\alpha_0}^{\tau} \sim \text{Uniform}(0.5, 10)$
asymptotic learning rate, $\alpha_\infty^{\tau s}$	$\alpha_\infty^{\tau s} \sim \text{Beta}(a_{\alpha_\infty}^{\tau}, b_{\alpha_\infty}^{\tau})$	$\theta_{\alpha_\infty}^{\tau} = (a_{\alpha_\infty}^{\tau}, b_{\alpha_\infty}^{\tau})$	$a_{\alpha_\infty}^{\tau} \sim \text{Uniform}(0.1, 10)$ $b_{\alpha_\infty}^{\tau} \sim \text{Uniform}(0.1, 10)$
information bonus, $A^{\tau shu}$	$A^{\tau shu} \sim \text{Gaussian}(\mu_A^{\tau hu}, \sigma_A^{\tau hu})$	$\theta_A^{\tau hu} = (\mu_A^{\tau hu}, \sigma_A^{\tau hu})$	$\mu_A^{\tau hu} \sim \text{Gaussian}(0, 100)$ $\sigma_A^{\tau hu} \sim \text{Gamma}(1, 0.001)$
spatial bias, $B^{\tau shu}$	$B^{\tau shu} \sim \text{Gaussian}(\mu_B^{\tau hu}, \sigma_B^{\tau hu})$	$\theta_B^{\tau hu} = (\mu_B^{\tau hu}, \sigma_B^{\tau hu})$	$\mu_B^{\tau hu} \sim \text{Gaussian}(0, 100)$ $\sigma_B^{\tau hu} \sim \text{Gamma}(1, 0.001)$
decision noise, $\sigma^{\tau shu}$	$\sigma^{\tau shu} \sim \text{Gamma}(k_\sigma^{\tau hu}, \lambda_\sigma^{\tau hu})$	$\theta_\sigma^{\tau hu} = (k_\sigma^{\tau hu}, \lambda_\sigma^{\tau hu})$	$k_\sigma^{\tau hu} \sim \text{Exp}(0.1)$ $\lambda_\sigma^{\tau hu} \sim \text{Exp}(10)$

Table 1: Model parameters, priors, hyperparameters and hyperpriors.

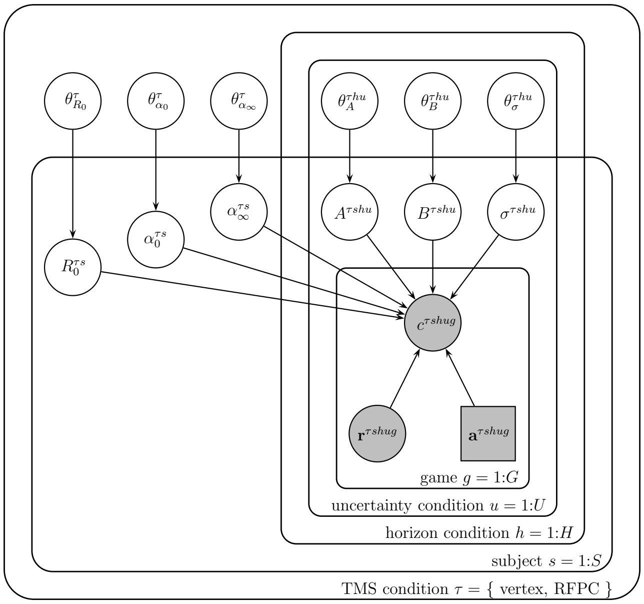


Figure 2: Graphical representation of the model. Each variable is represented by a node, with edges denoting the dependence between variables. Shaded nodes correspond to observed variables, i.e. the free choices $c^{\tau shug}$, forced-trial rewards, $\mathbf{r}^{\tau shug}$ and forced-trial choices $\mathbf{a}^{\tau shug}$. Unshaded nodes correspond to unobserved variables whose values are inferred by the model.

The hyper-parameters themselves are assumed to be sampled from 'hyperprior' distributions whose parameters are defined such that these hyperpriors are broad. For notational convenience, we refer to the hyperparameters that define the prior for variable X as θ^X. In addition we use superscripts to refer to the dependence of both parameters and hyperparameters on TMS stimulation condition, τ, horizon condition, h, uncertainty condition, u, subject, s, and game, g.

The particular priors and hyperpriors for each parameter are shown in Table 1. For example, we assume that the prior mean, $R_0^{\tau s}$, for each stimulation condition τ and horizon condition h, is sampled from a Gaussian prior with mean $\mu_{R_0}^{\tau}$ and standard deviation $\sigma_{R_0}^{\tau}$. These prior parameters are sampled in turn from their respective hyperpriors: $\mu_{R_0}^{\tau}$, from a Gaussian distribution with mean 50 and standard deviation 14, $\sigma_{R_0}^{\tau}$ from a Gamma distribution with shape parameter 1 and rate parameter 0.001.

Model fitting using MCMC The model was fit to the data using a Markov Chain Monte Carlo approach implemented in the JAGS package (Plummer et al.,2003) via the MATJAGS interface (psiexp.ss.uci.edu/research/programs_data/jags/). This package approximates the posterior distribution over model parameters by generating samples from this posterior distribution given the observed behavioral data. In particular we used 4 independent Markov chains to generate 4000 samples from the posterior distribution over parameters (1000 samples per chain). Each chain had a burn in period of 500 samples, which were discarded to reduce the effects of initial conditions, and posterior samples were acquired at a thin rate of 1. Convergence of the Markov chains was confirmed post hoc by eye.

Results

RFPC stimulation selectively inhibits directed exploration on the first free-choice

Model-free analysis Using the measures of directed and random exploration, $p(\text{high info})$ and $p(\text{low mean})$, we found that inhibiting the RFPC had a significant effect on directed exploration but not random exploration (Figure 3A, B). For directed exploration, a repeated measures ANOVA with horizon, TMS condition and order as factors revealed a significant interaction between stimulation condition and horizon ($F(1,24) = 4.96$, $p = 0.036$). Conversely, a similar analysis for random exploration revealed no effects of stimulation condition (main effect of stimulation condition, $F(1,24) = 0.88$, $p = 0.36$; interaction of stimulation condition with horizon, $F(1,24) = 1.24$, $p = 0.28$). Post hoc analyses revealed that the change in directed exploration was driven by changes in information seeking in horizon 6 (one-sided t-test, $t(24) = 2.62$, $p = 0.008$) and not in horizon 1 (two-sided t-test, $t(24) = -0.30$; $p = 0.77$).

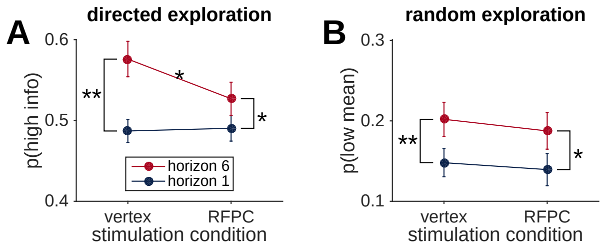


Figure 3: Model-free analysis of the first free-choice trial shows that RPFC stimulation affects directed, but not random, exploration. (A) In the control (vertex) condition, information seeking increases with horizon, consistent with directed exploration. When RFPC is stimulated, directed exploration is reduced, an effect that is entirely driven by changes in horizon 6 (* denotes $p < 0.02$ and ** denotes $p < 0.005$; error bars are ± s.e.m.). (B) Random exploration increases with horizon but is not affected by RFPC stimulation.

Model-based analysis Posterior distributions over the group-level means of all 13 parameters in the model are shown in Figure 4. The left column of Figure 4 shows the posteriors over each parameter while the right column shows the posteriors over the TMS-related change in each parameter. Both columns suggest a selective effect of RFPC stimulation on the information bonus in horizon 6.

Focussing on the left column first, overall the parameter values seem reasonable. The prior mean is close to the generative mean of 50 used in the actual experiment, and the decision parameters are comparable to those found in our previous work (Wilson et al.,2014). The learning rate parameters, α_0 and α_∞, were not included in our previous models and are worth discussing in more detail. As expected for Bayesian learning (Kalman,1960), the initial learning rate is higher than the asymptotic learning rate (95% of samples in the vertex condition, 94% in the RFPC condition). However, the actual values of the learning rates are quite far from their 'optimal' settings of $\alpha_0 = 1$ and $\alpha_\infty = 0$ that would correspond to perfectly computing the mean reward. This suggests a greater than optimal reliance on the prior ($\alpha_0 < 1$) and a pronounced recency bias ($\alpha_\infty > 0$) such that the most recent rewards are weighted more heavily in the computation of expected reward, R^i_t. Both of these findings are likely due to the fact that the version of the task we employed did not keep the outcomes of the forced trials on screen and instead relied on people's memories to compute the expected value.

Turning to the right hand column of Figure 4, we can see that the model-based analysis yields similar result to the model-free analysis. In particular we see a reduction (of about 4.8 points) in the information bonus in horizon 6 (with 99% of samples showing a reduced information bonus in the RFPC stimulation condition) and no effect on decision noise in either horizon in either the [2 2] or [1 3] uncertainty conditions (with between 40% and 63% of samples below zero).

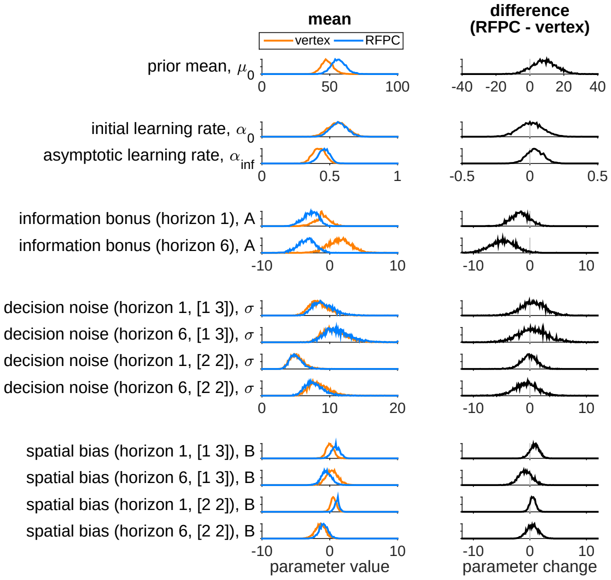


Figure 4: Model-based analysis of the first free-choice trial showing the effect of RFPC stimulation on each of the 13 parameters. Left column: Posterior distributions over each parameter value for RFPC and vertex stimulation condition. Right column: posterior distributions over the change in each parameter between stimulation conditions. Note that, because information bonus, decision noise and spatial bias are all in units of points, we plot them on the same scale to facilitate comparison of effect size.

Discussion

In this work we used continuous theta-burst transcranial magnetic stimulation (cTBS) to investigate whether right frontopolar cortex (RFPC) is causally involved in directed and random exploration. Using a task that is able to behaviorally dissociate these two types of exploration, we found that inhibition of RFPC caused a selective reduction in directed, but not random exploration. To the best of our knowledge, this finding represents the first causal evidence that directed and random exploration rely on dissociable neural systems and is consistent with our recent findings showing that directed and random exploration have different developmental profiles (Somerville et al.,2017). This suggests that, contrary to the assumption underlying many contemporary studies (Daw et al.,2006;Badre et al.,2012), exploration is not a unitary process, but a dual process in which the distinct strategies of information seeking and choice randomization are implemented via distinct neural systems.

Such a dual-process view of exploration is consistent with the classical idea that there are multiple types of exploration (Berlyne,1966). In particular Berlyne's constructs of 'specific exploration', involving a drive for information, and 'diversive exploration', involving a drive for variety, bear a striking resemblance to our definitions of directed and random exploration. Despite the importance of Berlyne's work, more modern views of exploration tend not to make the distinction between different types of exploration, considering instead a single exploratory state or exploratory drive that controls information seeking across a wide range of tasks (Hills et al.,2015;Kidd & Hayden,2015). At face value, such unitary accounts seem at odds with a dual-process view of exploration. However, these two viewpoints can be reconciled if we allow for the possibility that, while directed and random exploration are implemented by different systems, their levels are set by a common exploratory drive. More work will be required to determine whether this is the case.

While the present study does allow us to conclude that directed and random exploration rely on different neural systems, the limited spatial specificity of TMS limits our ability to say exactly what those systems are. In particular, because the spatial extent of TMS is quite large, stimulation aimed at frontal pole may directly affect activity in nearby areas such as ventromedial prefrontal cortex (vmPFC) and orbitofrontal cortex (OFC), both areas that have been implicated in exploratory decision making and that may be contributing to our effect (Daw et al.,2006). In addition to such direct effects of TMS on nearby regions, indirect changes in areas that are connected to the frontal pole could also be driving our effect. For example, cTBS of left frontal pole has been associated with changes in blood perfusion in areas such as amygdala, fusiform gyrus and posterior parietal cortex (Volman, Roelofs, Koch, Verhagen, & Toni,2011). In addition the same study showed that unilateral cTBS of left frontal pole is associated with changes in blood perfusion to the right frontal pole. Indeed, such a bilateral effect of cTBS may explain why our intervention was effective at all given that a number of neuroimaging studies have shown bilateral activation of the frontal pole associated with exploration (Daw et al.,2006;Badre et al.,2012). Future work combining cTBS with neuroimaging will be necessary to shed light on these issues.

References

Badre, D., Doll, B. B., Long, N. M., & Frank, M. J. (2012, 9 February). Rostrolateral prefrontal cortex and individual differences in uncertainty-driven exploration. *Neuron*, *73*(3), 595–607.

Berlyne, D. E. (1966, 1 July). Curiosity and exploration. *Science*, *153*(3731), 25–33.

Boorman, E. D., Behrens, T. E. J., Woolrich, M. W., & Rushworth, M. F. S. (2009, 11 June). How green is the grass on the other side? frontopolar cortex and the evidence in favor of alternative courses of action. *Neuron*, *62*(5), 733–743.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006, 15 June). Cortical substrates for exploratory decisions in humans. *Nature*, *441*(7095), 876–879.

Domenech, P., & Koechlin, E. (2015). Executive control and decision-making in the prefrontal cortex. *Current Opinion in Behavioral Sciences*, *1*, 101–106.

Hills, T. T., Todd, P. M., Lazer, D., Redish, A. D., Couzin, I. D., & Cognitive Search Research Group. (2015, January). Exploration versus exploitation in space, mind, and society. *Trends Cogn. Sci.*, *19*(1), 46–54.

Huang, Y.-Z., Edwards, M. J., Rounis, E., Bhatia, K. P., & Rothwell, J. C. (2005, 20 January). Theta burst stimulation of the human motor cortex. *Neuron*, *45*(2), 201–206.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Int. J. Eng. Trans. A*, *82*(1), 35.

Kidd, C., & Hayden, B. Y. (2015, 4 November). The psychology and neuroscience of curiosity. *Neuron*, *88*(3), 449–460.

Lee, M. D., & Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge University Press.

Plummer, M., et al. (2003). Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing* (Vol. 124, p. 125).

Pollmann, S. (2016). Frontopolar resource allocation in human and nonhuman primates. *Trends Cogn. Sci.*, *20*(2), 84–86.

Raja Beharelle, A., Polania, R., Hare, T. A., & Ruff, C. C. (2015). Transcranial stimulation over frontopolar cortex elucidates the choice attributes and neural mechanisms used to resolve Exploration-Exploitation Trade-Offs. *Journal of Neuroscience*, *35*(43), 14544–14556.

Rescorla, R. A., Wagner, A. R., et al. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, *2*, 64–99.

Rossi, S., Hallett, M., Rossini, P. M., Pascual-Leone, A., & Safety of TMS Consensus Group. (2009). Safety, ethical considerations, and application guidelines for the use of transcranial magnetic stimulation in clinical practice and research. *Clin. Neurophysiol.*, *120*(12), 2008–2039.

Somerville, L. H., Sasse, S. F., Garrad, M. C., Drysdale, A. T., Abi Akar, N., Insel, C., et al. (2017, February). Charting the expansion of strategic exploratory behavior during adolescence. *J. Exp. Psychol. Gen.*, *146*(2), 155–164.

Volman, I., Roelofs, K., Koch, S., Verhagen, L., & Toni, I. (2011, 25 October). Anterior prefrontal cortex inhibition impairs control over social emotional actions. *Curr. Biol.*, *21*(20), 1766–1770.

Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., & Cohen, J. D. (2014, December). Humans use directed and random exploration to solve the explore-exploit dilemma. *J. Exp. Psychol. Gen.*, *143*(6), 2074–2081.

Wischnewski, M., & Schutter, D. J. L. G. (2015, July). Efficacy and time course of theta burst stimulation in healthy humans. *Brain Stimul.*, *8*(4), 685–692.

A Neural Accumulator Model of Antisaccade Performance of Healthy Controls and Obsessive-Compulsive Disorder Patients

Vassilis Cutsuridis (vcutsuridis@gmail.com)
School of Computer Science, University of Lincoln,
Brayford Pool, Lincoln, LN6 7TS, U.K.

Abstract

Antisaccade performance in obsessive-compulsive disorder (OCD) is related to a dysfunctional network of brain structures including the (pre)frontal and posterior parietal cortices, basal ganglia, and superior colliculus. Previously recorded antisaccade performance of healthy and OCD subjects is re-analyzed to show greater variability in mean latency and variance of corrected antisaccades as well as in shape of antisaccade and corrected antisaccade latency distributions and increased error rates of OCD patients relative to healthy participants. Then a well-established neural accumulator model of antisaccade performance is employed to uncover the mechanisms giving rise to these observed OCD deficits. The model shows: i) increased variability in latency distributions of OCD patients is due to a more noisy accumulation of information by both correct and erroneous decision signals; (ii) OCD patients are *almost* as confident about their decisions as healthy controls; iii) competition via local lateral inhibition between the correct and erroneous decision processes, and *not* a third top-down STOP signal of the erroneous response, accounts for both the antisaccade performance of healthy controls and OCD patients.

Keywords: Eye movements; superior colliculus; computer model; response inhibition; OCD.

Introduction

In the antisaccade paradigm participats suppress a reflexive saccade (error prosaccade) in favor of a saccade to a position in the opposite hemifield (correct antisaccade) (Hallett, 1978). At least two processes take place during this paradigm: (1) suppression (or inhibition) of an error prosaccade towards the peripheral stimulus, and (2) generation of a volitional saccade to the opposite direction (antisaccade) (Everling and Fischer, 1998; Munoz and Everling, 2004). The reaction times (RT) of error prosaccades, antisaccades and corrected antisaccades, the error rate, the percentage of corrected errors, the amplitude of antisaccades and error prosaccades, and the final eye position of correct responses are some of the measures of antisaccade performance (Hutton and Ettinger, 2006) with the error rate being the most reliable measure of it. A large study of healthy young males has reported that error prosaccade and antisaccade RTs are highly variable and the error rate is about 20-25% (Smyrnis et al., 2002; Evdokimidis et al., 2002).

A recent experimental study reported an increase in error rates and in latency of corrected antisaccades in OCD patients (Damilou et al., 2016). The antisaccade performance deficit in OCD was speculated to be due a common dysfunctional network of brain structures including the (pre)frontal and posterior parietal cortices and superior colliculus. In this network there is a reported deficit in erroneous response inhibition control (Chamberlain et al., 2005).

Models of decision making involves a gradual accumulation of information concerning the various potential responses (Cutsuridis et al., 2007; Cutsuridis, 2010; Noorani and Carpenter, 2013, 2014, Cutsuridis et al., 2014; Cutsuridis, 2015, 2017). As soon as the target appears, a decision process starting at some baseline level T_0 representing the prior expectation, begins to rise at a constant rate r until it reaches a threshold T_h representing the confidence level required before the commitment to a particular course of action. Once T_h is crossed, then a response towards the target is initiated. Response time (RT) is the time from the onset of the decision process till when the decision signal crosses T_h. The rate of rise is sometimes assumed to vary randomly from trial to trial, with a mean μ and variance σ^2 (Reddi and Carpenter, 2000). Changes in the baseline level of activity, the rate of rise or the threshold often result in changes in response latency. Prior expectation and level of activation of intention influence the baseline levels of activation. Carpenter (1981) proposed if the cumulative RT distribution is plotted against 1/RT on reciprobit scale, then the resulting straight line can be used as a diagnostic tool to assess the contribution of different factors influencing the experimental results. In a choice reaction time task such as the antisaccade paradigm, the various choices are represented by different straight lines. If the lines swivel by the threshold T_h, then the mean and variances of the lines are unequal (Reddi and Carpenter, 2000). If the lines are shifted by μ, then the slopes ($1/\sigma$) of the lines are equal, but their latency medians are not (Reddi et al., 2003). If the lines cross, then the slopes are not equal, but their medians are (Nakahara et al., 2006).

In the present study, the Cutsuridis and colleagues (2014) model of antisaccade performance was used and extended into the realm of OCD. Previously recorded error rates and latencies of healthy and OCD participants (Damilou et al., 2016; Evdokimidis et al., 2002) were re-analyzed to show that OCD patients display higher error rates, increases in mean latency and variance of corrected antisaccades, and greater variability in shape of antisaccade and corrected antisaccade latency distributions relative to healthy participants. The Cutsuridis and colleagues (2014) neural model was then employed to decipher the biophysical mechanisms that gave rise to these antisaccade performance deficits in OCD. The model showed that i) increased variability in latency distributions of OCD patients was due to a more noisy accumulation of information by both (pre)frontal and posterior parietal centers representing the volitional (correct antisaccade) and reactive (erroneous prosaccade) decision signals, respectively, (ii) OCD patients were *almost as confident* about their decisions as healthy controls (i.e. the decision threshold level T_h value is almost the same in healthy controls and OCD patients), and iii) competition between the correct and erroneous decision processes, and *not* a third top-down STOP of the erroneous response, accounted for the antisaccade performance of both healthy controls and OCD patients.

Methods

Experimental data

Participants

Two groups of individuals participated in the study: healthy controls and OCD patients. Both participant groups were extensively described in two previously published studies (Evdokimidis et al., 2002; Damilou et al., 2016).

Antisaccade task

The antisaccade task for the healthy controls and OCD groups was identical to the experimental protocol used in the Evdokimidis and colleagues (2002) study. Stimuli were delivered through a 17-inch computer screen (LCD) located 1m away from the level of their eyes. Their head was immobilized using a chin rest. Subjects were informed about the requirements of the antisaccade task prior to its initiation. A calibration procedure was performed using a sequence of four saccadic eye movements, two to the left and two to the right of a central fixation target at an eccentricity of 10 deg. This process was then repeated with eye movements performed at 5 deg from the fixation point. During each antisaccade trial participants were instructed to fixate on a central fixation stimulus (white cross 0.3^o x 0.3^o of visual angle). After a variable period of 1–2 s, the central stimulus would disappear and a peripheral cue (the same white cross) would appear randomly at one of five positions ($2–10^o$ at 2^o intervals), either on the left or on the right hand side of the central fixation stimulus. The subject was instructed to make a saccade in the opposite direction from the peripheral target. Each subject performed 90 antisaccade trials (5 trials for each cue position) in a randomized order.

Eye movement recordings and analysis

For the control and OCD groups, eye movements were recorded from the right eye using the IRIS SKALAR infrared device. Stimulus presentation and recording of the responses was accomplished with a program written in Turbo Pascal 7.0 for DOS. A 12-bit A/D converter was used for data acquisition (Advantech PC-Lab Card 818L). Eye movement data were sampled at 600 Hz and stored in a PC for off-line data processing. Data pre-processing of all recordings was conducted using an interactive PC program (created using the Test-Point CEC Software). Trials with artifacts (blinks, etc.) in the analysis period or with any type of eye movement in the period of 100ms before the appearance of the peripheral stimulus were excluded from the analysis (Evdokimidis, et al., 2002). In addition, only the trials with response latency within the window of 80–600ms were included in the analysis. Based on these criteria, individuals who performed at least 30 valid antisaccade trials were only retained.

Metrics

The experimental control and patient saccade reaction times (RTs) were divided into three behavioral categories: (1) error prosaccades, (2) antisaccades, and (3) corrected antisaccades. Saccade reaction time (RT) was defined as the time interval from the onset of peripheral stimulus till the time of the first detectable eye movement. Corrected antisaccade RT was as the time between an error prosaccade and the subsequent corrected antisaccade.

Neural model

The model and its mathematical formalism were extensively described in Cutsuridis et al. (2014) study. Briefly, the model was a one-layer neural network of the superior colliculus (SC) with firing rate nodes (Fig. 1A). The total number of nodes in the network was assumed to be 100. Short-range lateral excitation and long distance lateral inhibition was also assumed between all nodes in model. The lateral interaction kernel w_{ij}, which allowed for lateral interactions between model nodes, was a shifted Gaussian, which depended only on the spatial distance between nodes and it was positive for nearby nodes to the node activated by the input and negative for distant nodes (Fig. 1B).

Model inputs were of two types: (1) a reactive input (I_r), which represented the error prosaccade decision signal and it was hypothesized to originate from the posterior parietal cortices (Munoz and Everling, 2004) and (2) a planned input (I_p), which represented the correct antisaccade decision signal and it was originated in the model from the frontal cortical areas (Munoz and Everling, 2004). In the model, each input was integrated in opposite model half according to the following way: if the reactive input activated a node and two of each nearest neighbors on each side in the left model half, then the planned input activated the mirror node and its two nearest neighbor nodes on each side in the right model half, and vice versa. The strengths of the external inputs were not equal ($I_p > I_r$).

In the model, the reactive input was presented first at time t = 50 ms, followed by the planned input, which was presented 50 ms later (t = 100 ms). Experimental evidence (Becker, 1989) reported that the difference in the afferent delays of the reactive and planned decision signals (inputs) is close to 50 ms. Both inputs remained active for 600 ms.

Results

As in the Cutsuridis and colleagues (2014) modeling study, to fit the experimental OCD data two model parameters were varied: the integration constant (τ) and the threshold (T_h). In the model, the integration constant was a parameter which indicated how fast or how slow the neuron integrated information. A large value of τ allowed the neuron to integrate information slowly. A small value of τ allowed the neuron to integrate information fast.

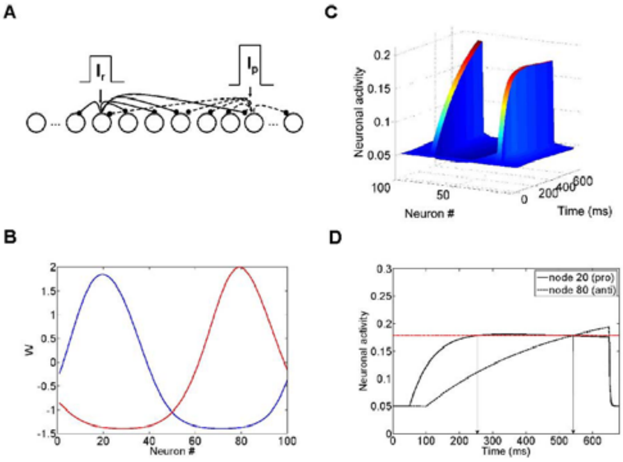


Figure 1. (A) Neural network model (reprinted with permission from Cutsuridis et al. (2014) study). (B) Lateral interaction kernels W for nodes 20 and 80 modelled as a shifted Gaussians (reprinted with permission from Cutsuridis et al. (2014) study). The kernels for nodes 20 and 80 were excitatory for the nearby nodes and inhibitory for the distant ones. (C) Neuronal activities of all nodes in the network as a function of time (ms) (reprinted with permission from Cutsuridis et al. (2014) study). (D) Neuronal activity of nodes 20 and 80 as a function of time (reprinted with permission from Cutsuridis et al. (2014) study). Node 20 encoded the reactive input (error prosaccade) and node 80 encoded the planned input (antisaccade). When both activities crossed the threshold (dotted horizontal line), then an eye movement decision was made. In this case, an error prosaccade followed by a corrected antisaccade.

Threshold was a model parameter that indicated how confident the model was to make a decision. When the neuronal activity crossed the threshold (see Fig. 1C), then a decision was made (i.e. an eye movement was generated).

In each trial run the integration constant τ values of the two nodes that encoded the erroneous prosaccade and the antisaccade decision signals took values from two normal distributions with different means and standard deviations. The model was then run for 5000 trials. In each trial the error prosaccade, antisaccade and corrected antisaccade latencies were recorded. In the model the error prosaccade reaction time was estimated as the time interval from the onset of the reactive input until the time the activity of the node encoding the reactive input reached a preset threshold (T_h) plus an additional 30 ms (Fig. 1D). The antisaccade reaction time was estimated as the time interval from the onset of the reactive input until the time the activity of the node encoding the planned input reached the threshold plus 30 ms (Fig. 1D). The corrected antisaccade reaction time was the time interval from threshold crossing of the error node activity until the threshold crossing of the correct node activity.

Table 1: Model parameters

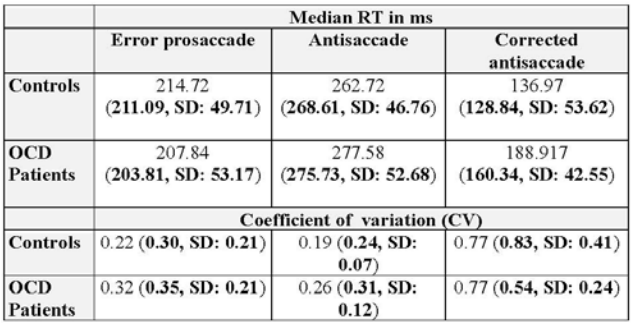
Symbol	Value		Symbol	Value	
	Controls	OCD		Controls	OCD
T_h	0.1767	0.177	σ	2π/10	2π/10
C	0.35	0.35	Δx	2π/N	2π/N
I_r	1	1	A	1	1
I_p	1.5	1.5	N	100	100
μ_1	0.01787	0.0165	β	0.5	0.5
σ_1	0.003	0.005	θ	0.5	0.5
μ_2	0.0056	0.0047	μ_n	0	0
σ_2	0.0016	0.002	σ_n	0.05	0.05
T	50 ms, unless mentioned otherwise		ntrials	5000	5000

To simulate the error prosaccade, antisaccade and corrected antisaccade RT distributions as well as the error rates of both healthy controls and OCD participant groups, the integration constants τ (μ and σ) for both nodes that integrated the reactive (μ_1 and σ_1) and planned (μ_2 and σ_2) inputs were varied (see Table 1 for parameter values). In both conditions, the threshold value at which as a decision was reached (parameter T_h in Table 1) was slightly higher in OCD patients than in healthy controls. The parameter values (μ_1, σ_1, μ_2, σ_2 and T_h) that best fitted the experimental data were found via exhaustive search of the parameter value space. The remaining model parameter values were the same as in Cutsuridis et al. (2014) study. The simulated median RTs for the error prosaccades, antisaccades and corrective antisaccades were 214.72 ms, 262.72 ms and 136.97 ms, respectively for the model controls and 207.84 ms, 277.58 ms and 188.917 ms, respectively for the model patients. The simulated median RT values are very close to the experimental ones (see Table 2). The simulated coefficients of variation (CVs) for the error prosaccades, antisaccades and corrected antisaccades were 0.22, 0.19 and 0.77, respectively for the controls and 0.32, 0.26 and 0.77, respectively for the patients. The simulated CV values are very close to the experimental ones (see Table 2).

To compare the experimental and simulated error prosaccade, antisaccade and corrected antisaccade RT distributions for both groups (healthy controls vs OCD patients) I replicated the measures reported in Cutsuridis and colleagues (2014) study. First, I estimated the experimental average cumulative distribution for error prosaccades, antisaccades and corrected antisaccades for both healthy controls and OCD patients by organizing the RTs for each subject (control subject or OCD patient) in ascending order and calculating the percentile values in increments of 5% (at 5, 10,15,20,...,95, 100%). The calculated percentile values from each subject were then averaged across each subject group (healthy controls or OCD patients) to give the experimental average group percentile values for error prosaccades, antisaccades and corrected antisaccades, which were then plotted in the average cumulative distribution (controls vs. patients) (see left plots of Figs 2A, 2B, and 2C). Ratcliff (1977) showed that the average distribution retains the basic shape characteristics of the individual distributions. Second, I repeated the same procedure for the error prosaccade, antisaccade and corrected antisaccade RTs of the virtual control and OCD subjects. The percentile values were then averaged across trial runs (5000 trial runs) for each subject group (virtual control subject vs virtual OCD patient) to give average subject group percentile values.

Carpenter and Williams (1995) showed that if the cumulative RT distribution is plotted using 1/RT in a reciprobit plot, then the RTs will fall on a straight line. Thus, the average cumulative distribution data of RT (error prosaccade, antisaccade and corrected antisaccade) for the experimental and simulated controls and patients in a reciprobit plot were transformed (see left plots of Figs 2A, 2B and 2C). A best-fitting regression line was computed for each behavioural category (error prosaccade, antisaccade and corrected antisaccade) in each simulated subject group (simulated controls vs simulated patients). An R correlation coefficient was estimated to assess how good fit was the regression line (simulated data) to the experimental data (open circles and dark squares). The model fit for each behavioural category and for subject group was excellent (correlation coefficient R was 0.99 for error prosaccades and antisaccades and 0.96 for corrected antisaccades in the healthy control group and 0.99 for error prosaccades and antisaccades and 0.97 for corrected antisaccades in the OCD group).

Table 2: Simulated median saccade reaction times, their standard deviations and coefficients of variation (CV) for healthy controls and OCD patients. Bold values in parentheses correspond to experimentally estimated means of medians of saccade RTs, their standard deviations and CVs for controls and patients.



	Median RT in ms		
	Error prosaccade	**Antisaccade**	**Corrected antisaccade**
Controls	214.72 (**211.09, SD: 49.71**)	262.72 (**268.61, SD: 46.76**)	136.97 (**128.84, SD: 53.62**)
OCD Patients	207.84 (**203.81, SD: 53.17**)	277.58 (**275.73, SD: 52.68**)	188.917 (**160.34, SD: 42.55**)
	Coefficient of variation (CV)		
Controls	0.22 (**0.30, SD: 0.21**)	0.19 (**0.24, SD: 0.07**)	0.77 (**0.83, SD: 0.41**)
OCD Patients	0.32 (**0.35, SD: 0.21**)	0.26 (**0.31, SD: 0.12**)	0.77 (**0.54, SD: 0.24**)

Discussion

What have learned from the model

Previously recorded antisaccade performance of healthy and OCD subjects (Damilou et al., 2016) was re-analyzed to show greater variability in mean latency and variance of corrected antisaccades as well as variability in shape of antisaccade and corrected antisaccade latency distributions and increased error rates of OCD patients relative to healthy participants. A neural accumulator model of antisaccade performance is then employed to uncover the biophysical mechanisms giving rise to these observed OCD deficits. The major finding of this study is that the brains of OCD participants when they performing the antisaccade task are noisier than the brains of healthy controls. This noise is reflected *mostly* in the rate of accumulation of information (μ and σ) and *less on* the threshold level T_h (confidence level required before commitment to a particular course of action). As we can see from Table 1 parameters μ_1 and μ_2 (see Table 1 for values) are greater in control condition than in the OCD condition meaning that error prosaccades, antisaccades and corrected antisaccades are slower in OCD patients than in healthy controls. Similarly, σ_1 and σ_2 (see Table 1 for values) are smaller in healthy control condition than in the patient one, which means that error prosaccade, antisaccade and corrected antisaccade latencies are more variable in OCD patients than in healthy participants. A physiological interpretation of the variability in the rate of accumulation of information (variability in parameter τ) is variability of NMDA based rate of evidence integration (Cutsuridis et al., 2007b). Experimental (Lewis, 2012) and computational (Kahramanoglou et al., 2008) studies have shown that NMDA hypofunction is implicated in neurodegenerative disorders such schizophrenia and OCD.

On the other hand, the value of T_h (threshold level) is almost the same in the OCD patient case as in healthy control one meaning that the OCD patients are as confident about their decisions as the healthy controls.

Comparison with other models

An important finding of this study is the absence of a third signal, inhibitory in nature, necessary to prevent the error prosaccade from being expressed when the antisaccade reached the threshold first. Such a third inhibitory signal has been speculated to exist by Noorani and Carpenter (2013, 2014) in the form of a "stop-and-restart" mechanism that partially captures the antisaccade performance of healthy participants (see the Cutsuridis (2015, 2017) studies for constructive critiques of Noorani and Carpenter (2013, 2014) models). In favor of the major finding of the current study that "competition via local lateral inhibition between the correct and erroneous decision processes, and *not* a third top-down STOP signal of the erroneous response, accounts for both the antisaccade performance of healthy controls and OCD patients" recent experimental evidence has demonstrated that lateral interactions within SC intermediate segment are more suitable for faithfully accumulating subthreshold signals for saccadic decision-making (Phongphanphanee et al., 2014). Another experimental study by Everling and colleagues (2013) challenges the idea of a third suppressive/inhibitory influence (STOP signal in the Noorani and Carpenter model) of prefrontal cortical areas on reflexive, erroneous prosaccade generation in the antisaccade paradigm.

Reciprobit plot as an insights tool of antisaccade performance

It has been suggested that when data are plotted on the reciprobit plot, then the resulting straight line on the reciprobit plot could be used a diagnostic tool to assess the contribution of different factors influencing the experimental results (Carpenter, 1981). When straight lines swivel (Reddi and Carpenter, 2000), then the mean and variances of the lines are unequal. When the lines are parallel and shifted by μ, then the slopes (1/ σ) of the lines are equal, but their latency medians are not (Reddi et al., 2003). When the lines cross, then the slopes are not equal, but their medians are (Nakahara et al., 2006). Along these lines we observed from the simulations that when the lines crossed (error prosaccade (right plot of Fig. 2A) and antisaccade (right plot of Fig. 2B)), then the median values of error prosaccade and antisaccade latencies are not significantly equal. When the lines are parallel and shifted (corrected antisaccades; right plot of Fig. 2C), then the median latencies are significantly different.

Acknowledgments

Author would like to thank Nikolaos Smyrnis for graciously sharing his control and OCD antisaccade datasets. The author declares that he has no competing financial interests.

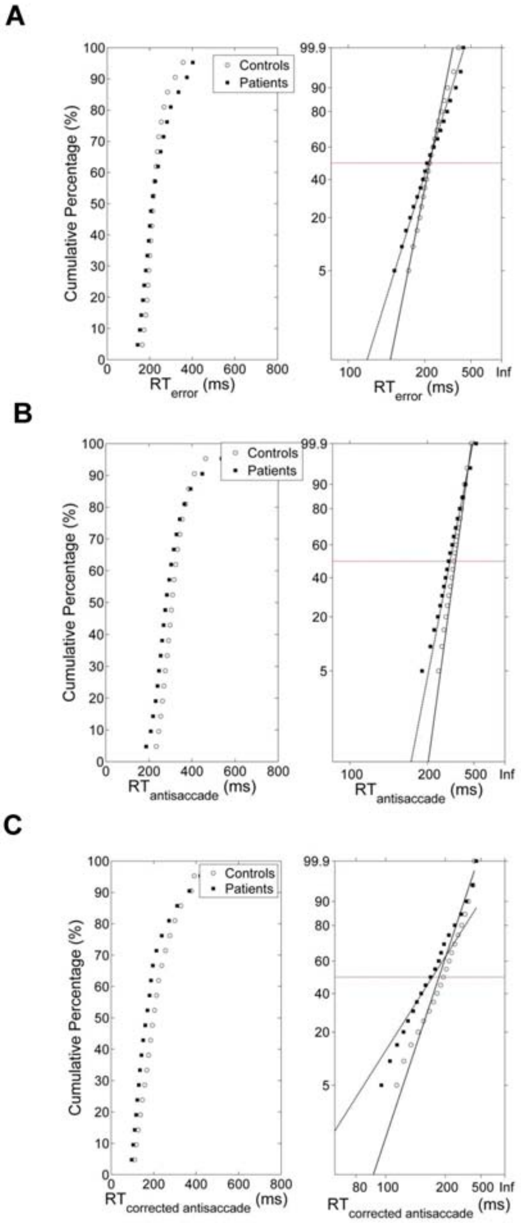


Figure 2. (*Left*) Experimental average cumulative RT distribution for controls (white empty circles) and patients (black squares). (*Right*) Reciprobit plots of the experimental (white empty circles and black squares) and simulated (solid lines) average cumulative RT distributions. The x-axis represents 1/RT and it has been reversed so that RTs increase to the right. Instead of 1/RT values the axis is marked with the corresponding RT values. The fitted lines correspond to linear regression (simulated data) on the experimental data (white circles and black squares) of each distribution (controls vs. patients). (A) Error prosaccades. (B) Antisaccades. (C) Corrected antisaccades.

References

Becker, W. (1989). Metrics. In: Wurtz R, Goldberg M, editors, *Neurobiology of saccadic eye movements*, Elsevier, NY, 12-67

Chamberlain, S. R., Blackwell, A. D., Fineberg, N. A., Robbins, T. W., Sahakian, B. J. (2005). The neuropsychology of obsessive compulsive disorder: the importance of failures in cognitive and behavioural inhibition as candidate endophenotypic markers. *Neurosci Biobehav Rev*, *29*, 399–419

Carpenter, R. H. S. (1981). Oculomotor procrastination. In D.F. Fisher, R.A. Monty, J.W. Senders, editors. *Eye movements: Cognition and visual perception.* Hillsdale, New Jersey: Lawrence Erlbaum, pp 237-246

Carpenter, R. H. S., Williams, M. L. L. (1995). Neural computation of log likelihood in the control of saccadic eye movements. *Nature, 377*, 59–62

Cutsuridis, V. (2015). Neural Competition via Lateral Inhibition between Decision Processes and Not a STOP Signal Accounts for the Antisaccade Performance in Healthy and Schizophrenia Subjects. *Front. Neurosci.*, *9*, 5.

Cutsuridis, V. (2010). Neural accumulator models of decision making in eye movements. *Adv Exp Med Biol, 657*, 61-72

Cutsuridis, V. (2017). Behavioral and computational varieties of response inhibition in eye movements. *Phil. Trans. R. Soc. B*, 372, 20160196.

Cutsuridis, V., Kahramanoglou, I., Smyrnis, N., Evdokimidis, I., Perantonis, S. (2007b). A Neural Variable Integrator Model of Decision Making in an Antisaccade Task. *Neurocomputing, 70(7-9)*, 1390-1402

Cutsuridis, V., Kumari, V., Ettinger, U. (2014). Antisaccade performance in schizophrenia: A Neural Model of Decision Making in the Superior Colliculus. Front. Neurosci., *8*, 13.

Cutsuridis, V., Smyrnis, N., Evdokimidis, I., Perantonis, S. (2007). A neural model of decision making by the superior colliculus in an antisaccade task. *Neural Netw.*, *20*, 690-704

Damilou, A., Apostolakis, S., Thrapsanioti, E., Theleritis, C., Smyrnis, N. (2016). Shared and distinct oculomotor function deficits in schizophrenia and obsessive compulsive disorder. Psychophysiology, *53(6)*, 796-805.

Evdokimidis, I., Smyrnis, N., Constantinidis, T. S., Stefanis, N. C., Avramopoulos, D., Paximadis, C., *et al.* (2002). The antisaccade task in a sample of 2006 young men. I. Normal population characteristics. *Exp. Brain Res.*, *147*, 45-52

Everling, S., & Fischer, B. (1998). The antisaccade: a review of basic research and clinical studies. *Neuropsychologia*, *36*, 885-899

Everling, S., & Johnston, K. (2013). Control of the superior colliculus by the lateral pre frontal cortex. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, *368*, 20130068.

Hallett, P. E. (1978). Primary and secondary saccades to goals defined by instructions. *Vis Res.*, *18*, 1279-96

Hutton, S. B., & Ettinger, U. (2006). The antisaccade task as a research tool in psychopathology: A critical review. *Psychophysiology*, *43*, 302-313

Kahramanoglou, I., Perantonis, S., Smyrnis, N., Evdokimidis, I., Cutsuridis, V. (2008). Modeling the effects of dopamine on the antisaccade reaction times (aSRT) of schizophrenia patients. Lecture Notes in Computer Science (LNCS) 5164, pp. 290–299. Berlin, Heidelberg: Springer-Verlag.

Lewis, D. A. (2012). Cortical circuit dysfunction and cognitive deficits in schizophrenia-implications for preemptive interventions. *Eur J Neurosci*, *35(12)*, 1871-8

Munoz, D. P., & Everling, S. (2004). Look away: The antisaccade task and the voluntary control of eye movement. *Nat. Rev. Neurosci.*, *5*, 218-228

Nakahara, H., Nakamura, K., Hikosaka, O. (2006). Extended later model can account for trial-by-trial variability of both pre- and post-processes. *Neural Networks*, *19*, 1027-46

Noorani, I., Carpenter, R. H. S. (2013). Antisaccades as decisions: LATER model predicts latency distributions and error responses. *Eur. J. Neurosci.*, *37*, 330–338.

Noorani, I., Carpenter, R. H. S. (2014). Re-starting a neural-race: antisaccade correction. *Eur. J. Neurosci.*, *39*, 159–164.

Phongphanphanee, P., Marino, R. A., Kaneda, K., Yanagawa, Y., Munoz, D. P., Isa, T. (2014). Distinct local circuit properties of the superficial and intermediate layers of the rodent superior colliculus. *Eur. J. Neurosci.*, *40*, 2329–2343.

Ratcliff, R. (1977). Group reaction time distributions and analysis of distribution statistics. *Psychol. Bull*, 86, 446–461.doi:10.1037/0033-2909.86.3.446

Reddi, B. A. J, Asrress, K. N., Carpenter, R. H. S. (2003). Accuracy, info.rmation and response time in a saccadic decision task. *J Neurophys.*, *90*, 3538-3546

Reddi, B. A., Carpenter, R. H. S. (2000). The influence of urgency on decision time. *Nat. Neurosci.*, *3*, 827-30

Smyrnis N, Evdokimidis I, Stefanis NC, Constantinidis TS, Avramopoulos D, Theleritis C, *et al.* (2002): The antisaccade task in a sample of 2006 young men. II. Effects of task parameters. *Exp. Brain Res.*, *147*, 53-63

Takahashi, M., Sugiuchi, Y., Izawa, Y., Shinoda, Y. (2005). Commissural excitation and inhibition by the superior colliculus in tectoreticular neurons projecting to omnipause neuron and inhibitory burst neuron regions. *J. Neurophysiol.*, *94*, 1707–1726.

A Neurocomputational Model of Learning to Select Actions

Andrea Caso
(andrea@andreacaso.com)

Richard P. Cooper
(R.Cooper@bbk.ac.uk)

Centre for Cognition, Computation and Modelling
Department of Psychological Sciences, Birkbeck, University of London
Malet Street, London, WC1E 7HX, United Kingdom

Abstract

We present an extension of a schema-based architecture for action selection, where competition between schemas is resolved using a variation of a neuroanatomically detailed model of the basal ganglia. The extended model implements distinct learning mechanisms for cortical schemas and for units within the basal ganglia. We demonstrate the functionality of the proposed mechanisms by applying the model to two classic neuropsychological tasks, the Wisconsin Card Sorting Task (WCST) and the Probabilistic Reversal Learning Task (PRLT). We discuss how the model captures existing behavioural data in neurologically healthy subjects and PD patients and how to overcome its shortcomings.

Keywords: schema theory; basal ganglia; Wisconsin Card Sorting Test, Probabilistic Reversal Task

Introduction

Schema theory is a framework based on the idea that behaviour in many areas depends on abstractions over instances, i.e., schemas. In these abstract terms, schema theory is very general and has been applied to different domains such as memory and motor control. Norman and Shallice (1980) applied the theory in the domain of routine sequential action. Their theory proposes that action schemas work in a cooperative or sequential fashion, but also that they compete with each other for activation.

While schema theory is helpful in representing functional interactions in the action-perception cycle, it is not committed to a specific neural implementation. However, at the neural level the basal ganglia have been proposed as a candidate for resolving competition between schemas in order to carry out action selection (Redgrave et al., 2001). In part this is because of their recurrent connections with the cortex.

In the first part of the paper we present a schema-theoretic model of action selection where competition between motor and/or cognitive schemas is resolved using a variation of a neuroanatomically detailed model of the basal ganglia. We assume that schemas are cortically represented but that schema selection (i.e., selecting one from a set of competing schemas) is facilitated by the basal ganglia. The latter receive multiple signals from the cortex but they are presumably 'content-free'. In other words, unlike their corresponding cortical structures, they are not directly related to the stimulus features. Following the description of the model we propose how learning may occur in the model subsequent to reward, introducing two parameters that drive separate learning mechanisms. Then, we proceed to present two examples of the model applied to two tasks: Wisconsin Card Sorting Task (WCST) and a variant of the Probabilistic Reversal Learning Task (PRLTv). We discuss computational results, the model fit with existing empirical data, and experiments that could further validate the model.

The Extended Schema-Theory Model

At a general level, the model can be understood as two systems or layers of computational units that feed signals to each other – a cortical system and a basal ganglia system. Each unit within the cortical system corresponds to to a schema, and represent a meaningful action or thought. Cortical units are connected with other cortical units and to the basal ganglia (BG) units (Fig. 1). These BG units take input from all cortical units at the same level of abstraction, generate an output signal, and feed it back to the same cortical units. The BG units serve to resolve competition between same-level schemas via the feedback loop between cortical and BG layers. Below, we will introduce two applications of the model – to the Wisconsin Card Sorting Test (WCST), which makes use of two distinct sets of schemas (cognitive and motor schemas) each with their own BG layer, and to a variation of the Probabilistic Reversal Learning task (PRLTv), which makes use of motor schemas only. First we describe the general model more fully.

Computation

Computation is carried out in both the cortical units and in the five nuclei which make up the basal ganglia (Fig. 2; for a complete description of the basal ganglia functional units see Alexander, 1990) according to the equations given below. In all cases, u_i represents the entry signal to the unit, a_i is the result of integration along the time domain, and o_i represents the output of the unit. The function σ computes the sigmoid function of the input, ensuring output values are bounded between 0 and 1. Sigmoid functions have a fixed slope but variable threshold. Varying the threshold of cortical or striatal units alters the way competition between units is carried out, and can be considered a function of phasic dopamine present in the circuit.[1]

[1] In a separate simulation it has been shown that the level of external dopamine from the substantia nigra pars compacta (SNpc) unit can be simulated by varying the threshold of the saturation curve in the striatum (β_{sma}), without making use of an additional unit.

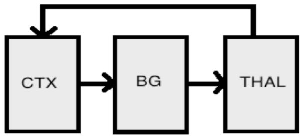


Figure 1: Schematic of the basal ganglia. Legend: Cortex-Thalamic complex (CTX-THAL), Striatum (STR), Subthalamic nucleus (STN), Globus Pallidus Internal/External Segment (GPi and GPe)

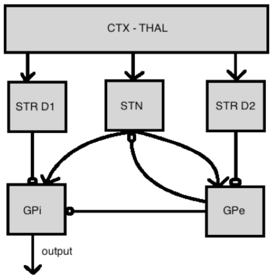


Figure 2: Schematic of the subunits that compose the basal ganglia. Legend: Cortex-Thalamic complex (CTX-THAL), Striatum (STR), Subthalamic nucleus (STN), Globus Pallidus Internal/External Segment (GPi and GPe)

Action selection is not the product of higher order schemas alone. Environmental features directly excite lower order schemas and can lead to selection of those schemas in the absence of higher order control. An excessive ratio or difference between bottom-up and top-down excitation of the lower-level schemas produces behaviours akin to those seen in some frontal patients (Cooper & Shallice, 2000).

Cortical Units (Motor or Cognitive)

$$\mathrm{u_i} \longmapsfrom \mathrm{o_{ext,i}} + \mathrm{o_{thal,i}}$$

$$\mathrm{a_i(t)} \longmapsfrom \partial \cdot \mathrm{a_i(t-1)} + (1-\partial)\mathrm{u_i(t-1)}$$

$$\mathrm{o_i} \longmapsfrom \sigma\,(\mathrm{a_i})$$

Striatum (D1 and D2):

$$\mathrm{u_i} \longmapsfrom \mathrm{o_{sma,i}}$$

$$\mathrm{a_i(t)} \longmapsfrom \partial \cdot \mathrm{a_i(t-1)} + (1-\partial)\mathrm{u_i(t-1)}$$

$$\mathrm{o_i} \longmapsfrom \sigma\left(\mathrm{a_{strD1/D2,i}}\right)$$

Subthalamic nucleus:

$$\mathrm{u_{stn,i}(t)} \longmapsfrom \mathrm{w_{stn}o_{sma,i}} + \mathrm{w_{gpe\_stn}o_{gpe,i}\,(t-1)}$$

$$\mathrm{a_{stn,i}(t)} \longmapsfrom \partial \cdot \mathrm{a_{stn,i}(t-1)} + (1-\partial)\mathrm{u_{stn,i}(t-1)}$$

$$\mathrm{o_{stn,i}} \Longleftarrow \sigma\left(\mathrm{a_{stn,i}}\right)$$

Globus Pallidus (External Segment):

$$\mathrm{u_{gpe,i}} \Longleftarrow \mathrm{w_{stn\_gpe}} \sum_{\mathrm{i}} \mathrm{o_{stn,i}} + \mathrm{w_{strD2\_gpe}o_{strD2,i}}$$

$$\mathrm{a_{gpe,i}(t)} \Leftarrow \partial \cdot \mathrm{a_{gpe,i}(t-1)} + (1-\partial)\mathrm{u_{gpe,i}(t-1)}$$

$$\mathrm{o_{gpe,i}} \Longleftarrow \sigma\left(\mathrm{a_{gpe,i}}\right)$$

Globus Pallidus (Internal Segment):

$$\mathrm{u_{gpi,i}(t)} \longmapsfrom \mathrm{w_{stn\_gpi}} \sum_{\mathrm{i}} \mathrm{o_{stn,i}} + \mathrm{w_{gpe\_gpi}o_{gpe,i}\,(t-1)} + \mathrm{w_{strD1\_gpi}o_{strD1,i}\,(t-1)}$$

$$\mathrm{a_{gpi,i}(t)} \longmapsfrom \partial \cdot \mathrm{a_{gpi,i}(t-1)} + (1-\partial)\mathrm{u_{gpi,i}(t-1)}$$

$$\mathrm{o_{gpi,i}} \longmapsfrom \sigma\left(\mathrm{a_{gpi,i}}\right)$$

Thalamus:

$$\mathrm{u_i} \longmapsfrom \mathrm{o_{gpi,i}}$$

$$\mathrm{a_i(t)} \longmapsfrom \partial \cdot \mathrm{a_i(t-1)} + (1-\partial)\mathrm{u_i(t-1)}$$

$$\mathrm{o_i} \longmapsfrom -\sigma\,(\mathrm{a_i})$$

Cortical and Basal Learning Mechanisms

The general model includes weighted connections from cortical schema units to basal ganglia units, and weighted connections from basal ganglia units back to cortical units. We assume that the weights are learned by separate reward-based mechanisms (for reasons given below). When the system is provided with positive ($r_i = +1$) or negative ($r_i = -1$) feedback after a response, two separate mechanisms control how the system adapts to new stimuli. We assume the following teaching signals are produced by rewards and activations:

$$R_i = (r_i - a_i) \tag{1}$$

$$S_i = r_i - \sum_{t=1}^{T-1} 2^{i-T+1} \cdot r_t \tag{2}$$

In Eq. 1 r_i represents the reward assigned to the ith schema and a_i represents the activation of the ith schema, t represents the trial and T is the total number of trials. Eq. 2 encodes the 'surprise' of the reward and assigns a greater value to the most recent trials, effectively implementing a form of 'memory'.

The teaching signals produce a variation in the threshold of the schema and basal ganglia unit saturation curves, β_{ctx} and β_{str}, respectively, as given by Eq. 4 and 5. Uniformly distributed noise ζ in the range [-0.1,0.1] is also added to prevent deadlock.

$$\beta_{str,i} \longmapsfrom \eta_0(\beta_{str,i} - \epsilon_{str}R_i \ + \ \zeta) \qquad (4)$$

$$\beta_{ctx,i} \longmapsfrom \eta_0(\beta_{ctx,0} - \epsilon_{ctx}S_i \ + \ \zeta) \qquad (5)$$

$$\eta_0(x) = \begin{cases} 1, & x > 1 \\ x, & 0 < x < 1 \\ 0, & x < 0 \end{cases} \qquad (6)$$

The left arrow indicates assignment[2]. Eq. 4 describes the change of threshold of the saturation curve of BG units following reward. Decreasing β_{str} augments the probability of the i^{th} schema being selected. Eq. 5 describes the change of threshold of the saturation curve of cortical units following reward. Unlike Eq. 4, the value of the β_{ctx} is centred around $\beta_{ctx,0}$ (set to 0.5 in all simulations). Eq. 6 is a limiting function which ensures that the thresholds remain within range.

Overall, this set of equations attempts to capture the division of labour between cortical structures and the basal ganglia. The two distinct learning signals that drive the overall model behaviour represent the direct (mesocortical, through the ventral tegmental area) and indirect (nigrostriatal, from the substantia nigra pars compacta) influence of dopamine to the task representation in the frontal circuits. Both equations are a function of reward, but while Eq. 4 slowly alters the probability of a channel to being selected, Eq. 5 energises schemas when surprise (the difference between expected and given reward) is high and therefore promotes fast dishabituation. Cognitive control emerges from the interaction between the two mechanisms

Theoretical Commitments

The core theoretical commitments of the model are the presence of cortical schemas, the presence of the basal ganglia that act as a content-free action selection device, and two different learning mechanisms for cortical schemas and the basal ganglia. Provided that the learning functions are both based on reward, the analytical form of the functions constitute peripheral hypotheses. Other peripheral hypotheses include the value of the threshold above which a schema is considered selected and the task-dependent number of schemas. The model can also be extended to accommodate other kinds of computation, such as that carried out in the cerebellum.

Model Applied to the WCST

Task and model description

In the Wisconsin Card Sorting Task (WCST), participants are required to sort a series of cards into four categories based on binary (i.e., correct/incorrect) feedback (Heaton, 1981). Each card shows one, two, three or four shapes, printed in one of four colours, and there are four shapes (triangle, star, cross, circle). It is therefore possible to sort cards according to colour, number or shape. To succeed, participants must match each successive card with one of four target cards (which show One Red Triangle, Two Green Stars, Three Yellow Crosses, Four Blue Circles), and use the subsequent feedback to discover the appropriate rule. However, once they have discovered the rule (as indicated by a succession of 10 correct sorts), the experimenter changes the rule without notice. The task yields a number of dependent measures, including the number of rules obtained (with a deck of fixed size – typically 64 or 128 cards), the number of cards correctly sorted, the number of perseverative errors (i.e., errors where the participant persists in using a rule despite having received negative feedback) and the number of Set Loss errors (i.e., errors where the participant fails to stick with a rule despite positive feedback).

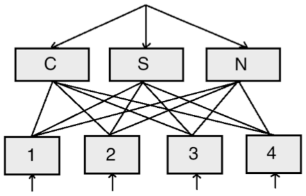


Figure 3: Schematic of the model, not showing competition between schemas. Cognitive schemas (top row) send signals to the motor schemas (bottom row).

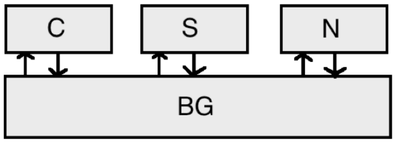


Figure 4: Schematic of the competition between schemas. The basal ganglia units compute the amount of inhibition that each schema receives given the activation of the others. Only cognitive schemas are shown here.

The model comprises three cognitive schemas and four motor schemas (see Fig. 3).[3] Cognitive schemas represent the selection rules (Sort by Colour, Sort by Number, Sort by Shape) while the four motor schemas represent the acts of putting the stimulus card below each of the four target cards. All schemas send signals to the basal ganglia units at the same level of hierarchy (Fig. 4), but only cognitive schemas implement the learning mechanisms outlined in Eq.4-6. Each schema has an activation level that varies over time as a function of input from various sources. Motor schemas are fed by cognitive schemas, and the signal from the cognitive layer to motor layer is rule-dependent. If, for instance, the stimulus card displays three red circles, the shape schema

[2] In assignment the value at the current trial is equal to a function of the same variable in the previous trial. Initial values are 0.5 plus a minimal amount of noise to randomise the first response.

[3] Source code for the simulation, including a complete list of parameters and their values, is available from the first author on request.

will excite the fourth motor schema (Four Blue Circles), the number schema will excite the third motor schema (Three Yellow Crosses), and the colour schema will excite the first motor schema (One Red Triangle).

Motor schemas are also fed by environmental cues which depend on the stimulus card features. Thus, when cognitive schemas are not strong enough to influence motor schemas, stimulus features alone may drive action selection. Feedback is given after each trial, and it drives learning within the cognitive schemas and their BG units as outlined in the previous section (Eq. 4 and 5). Learning in the motor schemas and their associated BG units is unnecessary in the WCST because randomisation of stimuli prevents a preference for a card position from being formed. A typical run of the task is shown in Fig. 5.

Simulation and results

We simulated 20 subjects for each value of the learning rates ε_{ctx} and ε_{str}, for a total of 560 subjects and recorded the relevant dependant variables (Fig. 6). Total Errors (TE), Perseverative Errors (PE) and Non Perseverative Errors (NPE) are all monotonic functions of ε_{str} and ε_{ctx} while Set Loss (SL) errors show a more erratic pattern. The value of the analysed dependent variables is a function of both ε_{str} and ε_{ctx}, but also of external activation of cognitive and motor schemas. These signals act as modulators between internal and external attentional process. An excessively low/high value of cognitive/motor external activation signals produces a general increase in all kind of errors. Varying these parameters produces performance more similar to behaviour exhibited by some frontal patients, where environmental cues drive action selection (Cooper & Shallice, 2000). Once baseline values for external excitations are set, we observe how the values of dependent variables fit data from young, older adults, and Parkinson's Disease (PD) patients. PD results from reduced dopaminergic input to the striatum (Siegelbaum et al., 2000) and it is therefore appropriately modelled by lower values of ε_{str}.

Total and Perseverative Errors Empirical data from PD patients (Paolo et al., 1996) performing the WCST show that perseverative errors are significantly greater in non-demented PD patients than in older controls, while the difference between older controls and younger subjects is not significant. The model successfully simulates this pattern of Total Errors and Perseverative Errors in healthy and PD patients with a set of values for (ε_{str} , ε_{ctx}) of (0.15, 0.08) and (0.05, 0.01), respectively. Thus, consistent with the neurophysiological hypothesis, PD patient performance may be accounted for by lower values of ε_{str}.

Set Loss and Non-Perseverative Errors Set loss errors have a different profile from all the other errors, suggesting the presence of distinct cognitive mechanisms underlying these and other errors. Empirical data from young, older controls and PD patients (Paolo et al., 1996) show that SL errors are not significantly greater in non-demented PD patients than in older controls. Paolo et al. (1996) also report that older controls tend to produce more SL errors than younger participants but the difference does not reach significance (*t(89)=1.89, p =.062*).

The model does not adequately capture the prevalence of set loss errors, but this limitation might be overcome by choosing parameters more carefully. In addition, it is necessary to further analyse how these errors arise in both the model and in experimental data. SL errors are relatively rare, and do not occur in all attempts at the task (either in human participants or in the model). Further work is required to see whether a more sensitive measurement of SL errors is needed.

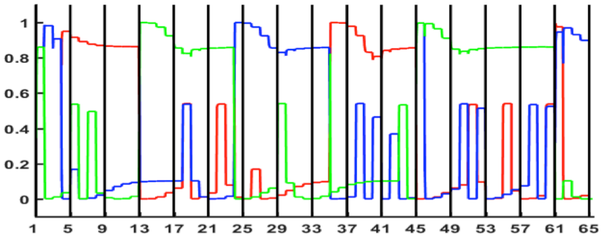


Figure 5: Cognitive schema activation in a typical run of the WCST. The red, green and blue lines represent the colour shape and number schemas, respectively. Black vertical lines have been plotted every 4 trials.

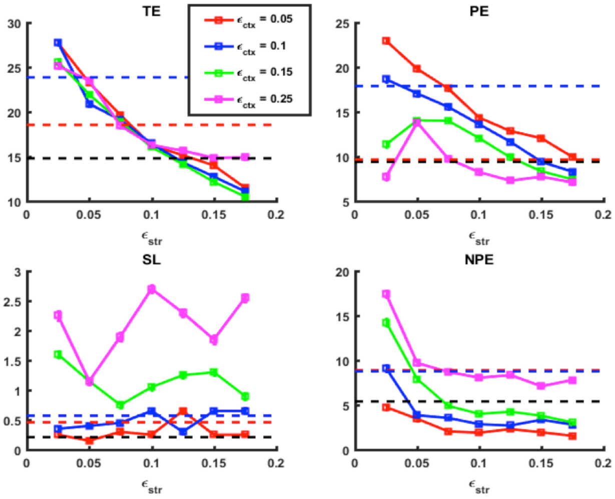


Figure 6: Plot of WSCT simulation results. Dependent variables shown are Total Errors (TE), Perseverative Errors (PE), Set Loss Errors (SL) and Non-Perseverative Errors (NPE). The dashed horizontal black lines, the red lines, and the blue lines represent the mean values of the dependent variables for young participants, older participants, and PD patients, respectively.

Discussion

Simulating the WCST yields an adequate fit with empirical data from healthy young controls and PD patients and it explains how perseveration errors might arise from an

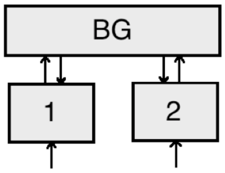


Figure 7: Model diagram for the PRLTv. Unlike the WCST, there are no higher order schemas that control the two lower order schemas.

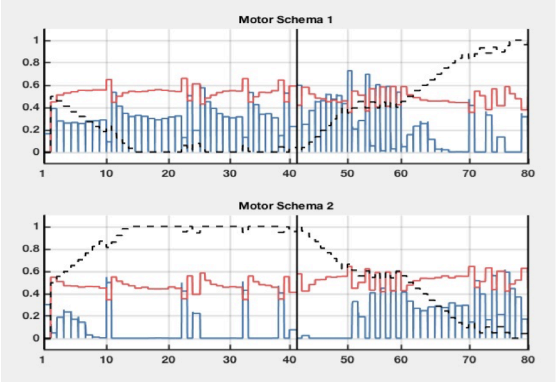


Figure 8: Typical run of the PRLTv. The blue line represents the schema activation while the red line and the dashed black line represent β_{ctx} and β_{str}, respectively.

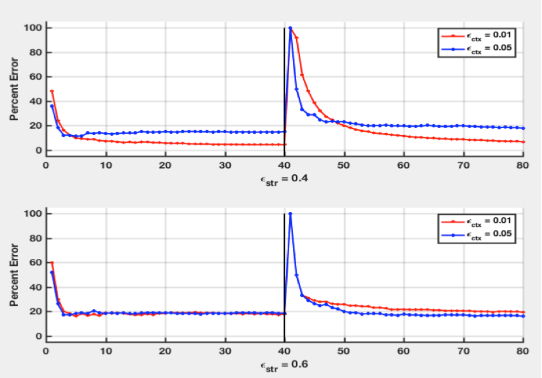


Figure 9: Plots of the model performance in PRLTv for different values of ε_{str} and ε_{ctx} across all trials. Points represent the error percentage for each stage of the task (acquisition and reversal)

impaired selection mechanism, in which rewards do not update quickly enough, or from an impaired schema activation mechanism, where surprising results are not powerful enough to trigger quick selection of a new rule. The dissociation between Set Loss and Perseverative Errors, which reflects the dissociation between distractibility and perseveration (Kaplan et al., 2006), is also replicated. Nevertheless, the model fails to fully explain the difference in Set Loss and Non-Perseverative Errors in healthy and PD populations. It is also unclear whether the difference between young and older control can be modeled with the two learning parameters alone (on the assumption that the trend reported by Paolo et al., 1996, indicates a real effect).

Model Applied to the PRLTv

Task and model description

Here, we apply the general model to a variant of the Probabilistic Reversal Learning Task (PRLTv; Cools et al., 2002). In this task, two stimuli are presented on each trial, but only one is the correct one. However, feedback is unreliable – the subject receives feedback that is correct only 80% of the time. After 40 trials the stimulus that receives the reward (i.e., positive feedback) is reversed. Again, feedback is correct 80% of the time. In the version of the task modelled here (unlike the standard experimental task), we assume that the subject is not told that feedback will be probabilistic. This allows us to test only stimulus-reward contingencies in absence of any super-ordinate rule.

To succeed at the task, subjects have to be able to stick to the first rewarded stimulus despite spurious feedback, but they also have to be able to reverse the choice and not perseverate when the contingency changes. The task is modelled as a simple stimulus-reward association, without higher order rules controlling the selection of lower schemas. The structure of the PRLTv thus is simpler than the one used for the WCST, and consists of only two cortical schemas with their associated basal ganglia units (Fig. 7). A typical run of the model is shown in Fig 8.

Simulation and results

We simulated 25 subjects for two values each of ε_{ctx} and ε_{str} for a total of 100 subjects and display the percentage error across the 80 trials (Fig. 9). Two performance measures are calculated: Errors to Criterion (ETC) are evaluated by counting the number of trials the subject takes to score 8 consecutive correct responses (ignoring spurious feedback). Consecutive-Perseverative (CP) errors are evaluated by counting how many trials from the reversal trial (41st trial) the subject takes to select the correct new response. Both variables are non-normally distributed, and therefore the Kruskal-Wallis H statistic has been used to test differences among the groups.

Errors To Criterion In the acquisition stage, ETCs are not significantly different, irrespective of the parameters (Fig. 10.). On the reversal stage, increasing ε_{str} from 0.4 to 0.6 inverts the ETC trend in the function of ε_{ctx}. The difference in ECT is significant in both the low ε_{str} value *($H(1) = 4.10$, $p = 0.043$)* and the high ε_{str} value (*$H(1) = 5.56$, $p = 0.018$).*

Consecutive Perseverative A low value of ε_{str} generally impairs the model by increasing perseveration (CP = 2), but only for lower values of ε_{str} (*$H(1) = 11.68$, $p < 0.001$*) (Fig.11).

Discussion

In the standard version of the Probabilistic Reversal Learning Task (e.g., Swainston et al., 2000), for which data from PD and age-matched controls is available, subjects are encouraged to stick with a rule even if it is occasionally

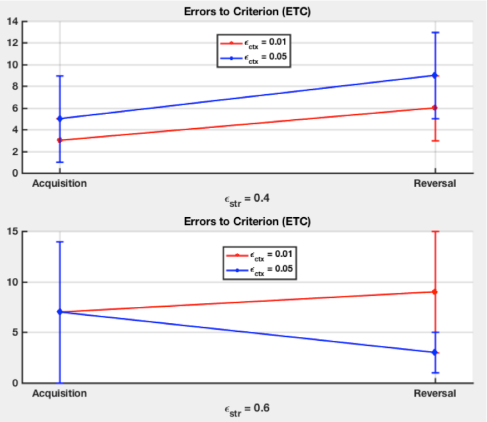


Figure 10: Errors to Criterion (ETCs) are shown here. Points and error bars represent medians and median absolute deviations.

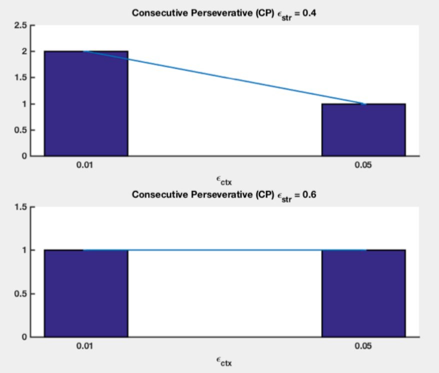


Figure 11: Consecutive Perseverative errors with four different settings (ε_{ctx} ,ε_{str}). Points and error bars represent medians and median absolute deviations.

wrong. This effectively creates a high-level schema. The variant of the task considered here deliberately avoids this and constitutes the lower-level version of the WCST, where only low-level schemas (those schemas that receive direct excitation from the environment) are activated and acted upon by the learning mechanisms. However, because of this difference in task instructions the model cannot be evaluated against the available data. The above results are therefore predictions that remain to be evaluated by contrasting the performance of PD patients and age-matched controls). Our model aims to capture computationally how a simple stimulus-reward association changes in terms of learning mechanisms that act directly on lower level schemas. Therefore the model needs to be experimentally validated with the adjusted behavioural task.

General Discussion

The general model is successful in replicating several empirical results and in reflecting the dissociation between distractibility (exemplified by SL errors in the WSCT and ETC in the PRLTv) and perseveration (exemplified by PE in the WSCT and ETC in the PRLTv). Limitations in accounting for experimental data in the WCST may be overcome by studying how subjects produce NPE and SL errors and whether the model accurately reflects this. Conversely, matching experimental data in the PRLTv requires running new experiments where instructions are reduced to a minimum. Ultimately, the model's purpose is to bridge the concept of neurotransmission, that acts as a medium to increase computational power, and the meaningful unit of action or thought. Thus, while the theoretical core assumptions seem to be capable of reproducing at least two tasks adequately, peripheral hypotheses on the learning mechanisms may require revision to achieve a better fit and to strengthen the link with the neurobiology.

References

Alexander, G. E., & Crutcher, M. D. (1990). Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends in Neurosciences*, *13*(7), 266-271.

Caso, A., & Cooper, R. (2017). A model of cognitive control in the Wisconsin card sorting test: integrating schema theory and basal ganglia function. *Cognitive Science Society*.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, *17*(4), 297-338.

Cools, R., Clark, L., Owen, A. M., & Robbins, T. W. (2002). Defining the neural mechanisms of probabilistic reversal learning using even t-related functional magnetic resonance imaging. *Journal of Neuroscience*, *22*(11), 4563-4567.

Heaton, R. K. (1981). *A manual for the Wisconsin card sorting test*. Western Psychological Services.

Kaplan, G. B., Şengör, N. S., Gürvit, H., Genç, İ., & Güzeliş, C. (2006). A composite neural network model for perseveration and distractibility in the Wisconsin card sorting test. *Neural Networks*, *19*(4), 375-387.

Norman, D. A., & Shallice, T. (1986). Attention to action: Willed and automatic control of behavior: In RJ Davidson, GE Schwartz, & D. Shapiro (Eds.), Consciousness and self-regulation (Vol. 4; pp. 1-18). *New York: Plenum Press*.

Paolo, A. M., Tröster, A. I., Blackwell, K. T., Koller, W. C., & Axelrod, B. N. (1996). Utility of a Wisconsin Card Sorting Test short form in persons with Alzheimer's and Parkinson's disease. *Journal of Clinical and Experimental Neuropsychology*, *18*(6), 892-897.

Siegelbaum, S. A., & Hudspeth, A. J. (2000). *Principles of neural science* (Vol. 4, pp. 1227-1246). E. R. Kandel, J. H. Schwartz, & T. M. Jessell (Eds.). New York: McGraw-hill.

Swainson, R., Rogers, R. D., Sahakian, B. J., Summers, B. A., Polkey, C. E., & Robbins, T. W. (2000). Probabilistic learning and reversal deficits in patients with Parkinson's disease or frontal or temporal lobe lesions: possible adverse effects of dopaminergic medication. *Neuropsychologia*, *38*(5), 596-612.

Gaps Between Human and Artificial Mathematics

Aaron Sloman (a.sloman@cs.bham.ac.uk)
School of Computer Science, University of Birmingham,
Birmingham B15 2TT, UK

Abstract

The Turing-inspired Meta-morphogenesis project begun in 2011 was partly motivated by deep gaps in our understanding of mathematical cognition and other aspects of human and non-human intelligence and our inability to model them. The project attempts to identify previously unnoticed evolutionary transitions in biological information processing related to gaps in our current understanding of cognition. Analysis of such transitions may also shed light on gaps in current AI. This is very different from attempts to study human mathematical cognition directly, e.g. via observation, experiment, neural imaging, etc. Fashionable ideas about "embodied cognition", "enactivism", and "situated cognition", focus on shallow products of evolution, ignoring pressures to evolve increasingly *dis*embodied forms of cognition to meet increasingly complex and varied challenges produced by articulated physical forms, multiple sensory capabilities, geographical and temporal spread of important information and other resources, and "other-related meta-cognition" concerning mental states, processes and capabilities of other individuals. Computers are normally thought of as good at mathematics: they perform logical, arithmetical and statistical calculations and manipulate formulas, at enormous speeds, but still lack abilities in humans and other animals to perceive and understand geometrical and topological possibilities and constraints that (a) are required for perception and use of affordances, and (b) play roles in mathematical, and proto-mathematical, discoveries made by ancient mathematicians, human toddlers and other intelligent animals. Neurally inspired, statistics-based (e.g."deep learning") models cannot explain recognition and understanding of mathematical *necessity* or *impossibility*. A partial (neo-Kantian) analysis of types of evolved biological information processing capability still missing from our models may inspire new kinds of research helping to fill the gaps. Had Turing lived long enough to develop his ideas on morphogenesis, he might have done this.

Keywords: Archimedes; Euclid; Kant; geometry; topology; vision; evolution; biological information processing; limitations of current computational models evolution as a blind mathematician.

Introduction

There are deep gaps in current AI models, related to gaps in theories of cognition, especially mathematical cognition (despite impressive mathematical powers of computers). The Turing-inspired Meta-Morphogenesis project, proposed in 2011 asks new questions about evolution of biological information processing, identifying what needs to be explained and possible types of explanation corresponding to different evolutionary stages.[1] Large sums are being spent in the hope that more training on more data can diminish, and eventually remove, those gaps, guided by research on how humans acquire the relevant competences and on brain mechanisms involved, but the research focuses on a *subset* of the relevant competences and mechanisms, leaving much unexplained. E.g. research that focuses on numerical competences, ignores geometric and topological competences, that are arguably more fundamental, in ways that I'll explain later. Moreover research on statistics based learning cannot explain discoveries of necessary truths, e.g. geometrical, topological and arithmetic truths.

Many psychologists also ignore important mathematical features of competences being investigated, because they don't clearly distinguish empirical from non-empirical learning. For example, not all psychologists studying number cognition seem to realise that full understanding of cardinal and ordinal numbers depends on grasping that one-to-one correspondence (bijection) is a transitive and symmetric relation (and therefore also reflexive), and moreover those properties are *necessary* (i.e. non-contingent) features of bijection, but not logical or definitional features. This was pointed out by Kant in 1781, though he knew of no explanatory mechanisms. My 1962 thesis (now online) defended Kant against common criticisms, but I had never heard of AI then and I lacked the opportunity to base a defence on computational modelling, a gap I began trying to fill in my 1978 book. Four decades later there still seem to be no working AI systems able to replicate the discoveries in topology, geometry and arithmetic, made by ancient mathematicians such as Archimedes, Euclid, Zeno and others, nor the closely related, hard to observe, discoveries unwittingly made by pre-verbal human toddlers,[2] or even squirrels and nest-building birds.

A rich sample of approaches to the problems of characterising and explaining numerical competences can be found in a BBS survey by Rips *et.al.*, including commentaries and responses. Unfortunately influences on individual mathematical development now are so diverse, including biological, physical, cultural, educational and individual differences, and so little attention is paid to the problem of specifying *implementable mechanisms*, as opposed to verbal descriptions of what brains or minds do, that the research is inevitably fragmentary and inconclusive and proposed theories lack the precision required to guide designs for testable working models.

Piaget drew attention to many combinations of competence and incompetence displayed by children, and produced evidence that most did not understand that 1-1 correspondence is a transitive relation until they are five or six years old. It is also symmetric, unlike many transitive relations children learn about (e.g. "taller than", "heavier than"). Unfortunately, calling this learning about "conservation" misleadingly sug-

[1] References have been deleted in this version for lack of space, but can be found, with links to online papers, at http://www.cs.bham.ac.uk/research/projects/cogaff/sloman-iccm17.pdf

[2] Like the 17.5 month old child apparently testing a conjecture in 3D topology here http://www.cs.bham.ac.uk/research/projects/cogaff/misc/toddler-theorems.html#pencil

gests that understanding preservation of numerosity across spatial rearrangement is related to understanding that matter is conserved when rearranged. One-to-one correspondences can hold between completely abstract entities that have no matter to conserve. This is obvious to mathematicians, but perhaps not all developmental researchers.

One common Piagetian test for understanding numerosity tends to use examples of two types (e.g. apples, bananas) and supertype (e.g. fruit) in situations where there are (e.g.) more apples than bananas and children are asked whether there are more apples or more fruit. At a certain age they tend to say "more apples". However, there is usually no attempt to check that they have understood the question as intended. I found that if a child is asked to count the apples, then to count the fruit, then asked the same question, the correct answer is given. Some then generalise, without help, to other cases, e.g. giving the right answer to the question "Are there more open windows or more windows?" asked of a building with far more windows open than shut. This suggests that some children interpret the original question wrongly. I don't know if any psychologist has tried tampering with Piaget's experiment in this way. However, Margaret Donaldson showed in 1978 that slight variants of some of Piaget's other experiments, produced significantly different results.

My aim is not to criticise Piaget or his (often less well informed) followers but to draw attention to problems of empirical research not based on deep theories. Is there a deep theory in neuroscience capable of explaining what sort of late developing neural mechanism can change the powers of a child's brain so that the necessary transitivity and symmetry of one-to-one correspondence is grasped? This can be viewed as a topological problem about two networks of connections, e.g. a network formed by setting up a one-to-one correspondence between elements of sets A and B, and one between elements of B and C. We can see (How?) that if A, B and C are disjoint sets, the two sets of links can always be concatenated to form one-to-one relationships between A and C. Does anyone have a theory as to how brain mechanisms can detect, or even represent, the impossibility of any counter-example – i.e. the fact that the transitivity is a *necessary* truth? The work of mathematical logicians (e.g. Frege, Russell and others) allows the transitivity to be proved (tediously) in a formal logical system, but it was understood by ancient mathematicians (and young learners), centuries before those formal proof methods had been discovered. What happened in their brains when the *necessity* of transitivity of bijection, i.e. the *impossibility* of counter examples, was grasped?

Mathematical discoveries are not concerned with empirical or contingent regularities but with necessary connections and impossibilities (e.g. internal angles of a planar triangle *necessarily* sum to half a rotation, and it is *impossible* for any number to be the largest prime). How could we check that a brain mechanism is able to represent and use these notions of necessity and impossibility, which are features of mathematical discoveries, but not empirical discoveries? The answer will depend in part on a good theory of the semantics of modal concepts – often taken nowadays to be "possible world" semantics.[3] However, ancient mathematicians did not need this notion of a possible world: they were exploring compatible and incompatible collections of relationships in *this* world, often represented diagrammatically (Sloman 1962). .

So mathematical (as opposed to empirical) discoveries about numbers, lines, angles, etc. require use of (alethic) *modal* concepts (e.g. "possible", "impossible", "necessarily true", "necessarily false"). Standard ways of acquiring general information by observing instances and collecting statistics, cannot yield such mathematical knowledge, since that requires more than observed regularities. Perhaps many badly taught learners never get beyond memorising what they have been taught, but that's not what needs explaining.

I am not aware of any computational model that is able to replicate not only those arithmetical and geometrical discoveries but also other topological impossibilities that children seem to understand without mathematical training, for example that two solid rings cannot become linked and unlinked simply by being moved continuously, or that a shoe-lace cannot be pulled out of lace-holes twice as fast by pulling both ends at once. Nor does any AI model that I know of explain this. There is no evidence that AI theorem provers that draw conclusions from logical axioms can model what a young child, or an intelligent squirrel or crow does, or what ancient mathematicians did over 23 centuries ago, long before discovery of modern logic and algebra, and Descartes' use of arithmetic to model geometry.

Kant pointed out that ancient mathematical discoveries are characterised by being (a) non-empirical, (b) non-analytic (i.e. not derivable from definitions using only logic) and (c) non-contingent – the truths and falsehoods are instances of *necessity* and *impossibility* as explained in my thesis. This does not imply that mathematicians are *infallible*: they can and do make mistakes of various sorts, though they often discover and correct their mistakes, as demonstrated in *Proofs and Refutations* by Lakatos.

The 20th Century discovery that physical space is non-Euclidean is often regarded as demonstrating that Kant was wrong about mathematical knowledge, whereas it merely shows that some of his examples were wrong. He could have used the discovery that a subset of Euclidean geometry can be extended in different ways, yielding Euclidean and non-Euclidean geometries, as an example of a mathematical truth that is synthetic, necessarily true and not empirically based. Non-Euclidean geometries had been discovered before the 1919 eclipse showed that physical space was not Euclidean. Such discoveries add to what needs to be explained by neuroscience and modelled by AI.

Regarding arithmetic: is there a neural theory explaining how brains generate and control parallel sequences of actions required in counting operations of various sorts, with different stopping conditions depending on the task and various

[3] https://en.wikipedia.org/wiki/Possible_world

ways in which counting errors can be detected and be corrected, as described in Ch. 8 of *CRP*?[4] An explanatory mechanism should explain how counting can be applied, via different senses and movable body parts, to events, continuous processes (e.g. rotations, changes of direction, skin strokes, or sound oscillations), to static objects, and to abstract entities (e.g. numbers, words), along with self-monitoring to detect departures from strict one-to-one correspondence. Moreover, some mathematical discoveries can be made by noticing novel features of such thinking processes, e.g. repeated patterns. I suspect no known *neural* mechanism explains how reflection on processes produced by number generating mechanisms can lead to the concept of a non-terminating sequence, and then to an understanding that there are infinitely many numbers. What allows a child to understand "never stops"?

Another discovery that I believe is beyond current AI theorem provers was known to Archimedes and others: namely adding the *neusis* construction to Euclidean geometry, allowing motion of a straight-edge with two marks, makes it easy to trisect an arbitrary angle, which is impossible in standard Euclidean geometry.[5] What would a neural explanation of such a discovery process look like? Finding brain regions that are active during such discoveries does not tell us how brains encode universally quantified semantic content, or how they derive new semantic contents. It cannot be assumed that such discoveries are based on applying rules of modern logic (e.g. predicate calculus) to logical axioms, in part because modern logic was not available to ancient mathematics: it was mostly created recently by thinkers like Boole, Peano, Frege, Russell and others. Moreover, Euclidean geometry was not axiomatised using modern logic until 1899, by David Hilbert. Trisection was proved impossible in that system, so discovery of a construction that trisects an arbitrary angle must have used a different mode of spatial reasoning. I suspect ancient discoveries in geometry and topology were closely related to the need to identify positive and negative affordances, shared with other intelligent species. But evolution added some additional, unknown(?) discovery or reasoning mechanism in humans.

Meta-cognitive mechanisms, allowing internal processes based on previous competences to become objects of reflection during their performance seem to be required for some new mathematical insights. Many practical tasks can make use of multiplication and division, e.g. making sure that every member of a group has two shoes, or dividing N tasks between M people. Reflecting on this leads to the discovery that some sets with N members can be divided into M equal sets, but not into M+1 equal sets, and eventually that some numbers *cannot* be divided into any number of equal sets: they are primes, already familiar to Euclid. It is not clear how the *impossibility* is recognized, as opposed to mere *repeated failure*. Statistics-based learning mechanisms could not discover impossibilities and necessary truths: those are not degrees of probability. (However mathematical theorems about probabilities are necessary truths, not probabilistic assertions!)

Piaget (who had studied Kant, Frege and Russell) understood some of the problems. His two posthumous books were on *possibility* and *necessity*, though he lacked the tools required to solve our problems.

Mathematical meta-cognition

Metacognitive reasoning processes seem to have enabled Euclid (or a predecessor) to discover and prove that there cannot be a largest prime number, so there must be infinitely many prime numbers. How did evolution produce mechanisms with such capabilities, and how do they work? Perhaps a "duplicate then differentiate" transition in our evolutionary history somehow produced meta-cognitive capabilities, allowing comparisons of modes of thinking on different occasions, leading to important insights concerning differences between reliable and unreliable reasoning, enabling introspected reasoning processes to be described and modified while they were being performed, and allowing mistakes of reasoning to be discovered and eliminated, or successful modes to be combined to form more complex modes.

Such meta-cognitive abilities would also have social consequences, e.g. allowing strategies discovered during self-debugging to be later taught to others.[6] Every good mathematics teacher knows that learning to detect mistakes in reasoning is a deep part of mathematical education. More generally, the extension of meta-cognition from direct self-observation to indirect other-observation can help with effective other-debugging processes. I don't know if anyone has an appropriately deep theory of how brains encode and manipulate self- and other- directed meta-cognitive information. (Could Barnden's *ATT-Meta* system be a start?)

Can we get clues from biological evolution?

If a bird is seen to be flying around in an elliptical orbit, it will not be because the bird's motion is caused by elliptical physical motion of something outside the bird, as rotary motion of a leaf in a river whirlpool is caused by the motion of the water. The bird will have information about its environment (e.g. about possible prey, a possible nesting site, or a predator approaching its nestlings) and identified needs (e.g. to get food, to find a good place for a nest, to find nesting material, to distract a predator, etc.) It will also need the ability to increase or decrease speed and change direction. Depending on the circumstances, the bird's motion will use energy (either in its muscles, or in wind or updrafts, or gravity), controlled on the basis of constantly changing information, to produce motion with intended results. There may or may not be additional meta-cognition (self-awareness). Instead of being moved solely by external physical forces, as planets and

[4] Revised edition online at http://www.cs.bham.ac.uk/research/projects/cogaff/crp#chap8

[5] For more detail see http://www.cs.bham.ac.uk/research/projects/cogaff/misc/trisect.html

[6] However, it's a fashionable mistake in some circles to assume that mathematical discovery necessarily requires social uses of language, just as its a fashionable mistake to assume physical embodiment plays a role in all mathematical reasoning.

clouds are, the bird has information-processing mechanisms that control its motion. E.g. it can select some information items rather than others then select and execute an action, then switch to a different goal and different action. Evolution changes the amount and variety of information that can be acquired, manipulated, stored and used, and the variety of types of needs and goals that can drive such processes.

Long before humans existed, various mathematical structures and relationships, some but not all numerical, were involved in control processes, including increasing or decreasing turn angles, speed, height, joint angles, forces applied, etc. At some stage humans developed meta-informational (meta-cognitive) abilities to reflect on, reason about, increasingly complex examples of such structures and relationships, including possible future structures; e.g. shelters not yet built, clothing not yet made from an animal skin, a meal whose ingredients are not yet assembled.

As yet unknown evolutionary changes must have supported new proto-mathematical abilities for manipulating and using information about structures, processes, actions, forces, etc. including future possible (intended) cases. A large subset is shared with other intelligent species. The mathematics that we teach and do research on is just a small subset, and almost certainly cannot be understood independently of the less obvious mathematical competences we share with many other species, especially topological and geometrical competences. Different mathematical structures occur in percepts, in intentions, in plans and, later on, in linguistic communications.

Evolutionary pressures for mathematical minds

Increasingly complex forms of life need to use increasingly complex and varied information structures including motives: information states concerned not with what *is* the case but with what *should be* the case, i.e. not just belief-like but also desire-like information contents of increasingly complex kinds. I am not claiming that ALL intelligent behaviour is based on current biological needs, or expected rewards, since some motives are triggered as "internal reflexes" by opportunities without any expected benefit, as can be seen in much playful activity in young children, kittens, apes, and others. What is learnt in such contexts can have consequences that are later useful in ways that the individual could not possibly predict. So although the mechanisms do not involve *expected* rewards, the indirect benefits they previously produced in ancestors may explain the survival of the goal generating mechanisms in their descendants, though not how they formed in the first place (using specially evolved construction kits).

There may be "branch points" during development where different lineages take different branches, under control of genome and environment. But at later stages of development evolution can support greater environmental variation, so that genetically programmed developmental choices may use information previously acquired during development. The fact that common gene-based language potential can support development of thousands of different languages in different contexts illustrates this.

That requires the genome to have a mathematically abstract language specification with very rich generative power, as Chomsky pointed out long ago. I suggest that that is a common feature of biological intelligence, which began with evolution of intelligent control systems in many species that have never been able to use human languages. But they must have rich internal languages for specifying percepts, goals, actions, and environmental structures, including structures that were never encountered by earlier members of the species. A special case is ability to represent entirely new affordances–not unique to humans. .

One of the deep discoveries of evolution was the need for reflexes: actions triggered without the agent having any idea what the benefits are. We need to generalise this to include reflex triggering of new internal motivational states that join other current motivational states, and may or may not lead to action, depending on what else is going on. I call this "Architecture Based Motivation" (ABM) in contrast with "Reward Based Motivation" (RBM) which requires every selected motive to be associated with some measurable expected utility.

ABM seems to be the basis of much exploratory and playful behaviour, including developing linguistic abilities of different sorts, e.g. early babbling and later uses of increasingly complex syntactic forms and growing vocabularies. This may be a source of mathematical development and discovery in young humans (with much individual variation). It also depends on prior, presumably genome-derived, mathematical competences required for exploring novel semantic contents.

As evolution produced increasingly complex organisms, with increasingly complex time-varying needs, and complex articulated bodies capable of rich and varied interactions with the environment, the requirements for mathematical abstraction in information processing increased, including use of geometrical and topological information about spatial structures and both observed and desired changes in spatial relationships, unlike organisms that simply depend on physical influences such as wind or water or the intervention of other organisms to produce the changes they need, e.g. use of other organisms for seed dispersal.

Simple types of information-based control are *online*: information is used as it is acquired and immediately overwritten by new information, e.g. if an animal moves continuously towards a fixed or moving edible target. More sophisticated organisms combine information fragments acquired at different times to produce richer information-structures concerning the environment, e.g. a human (or urban animal) storing and integrating information about the layout of a town and later using the information to work out a route that will reach a new target. This uses *offline* information processing, and *offline* control: actions may be selected long before they are performed, unlike *online* homeostatic control. The richer the environment, the more varied its structures, routes, materials, and other resources, the more powerful the organism's mathematical resources will need be to be able to create and reason about novel possibilities for achieving goals, avoiding dangers, etc. Because of the need to cope with novelty by getting things right first time, empirical learning from repeated trials will be of limited use. This is where mathematical competences are so biologically useful: solutions can be evaluated in advance by reasoning, using structural relationships, instead of having to be evaluated only by repeated testing.

There are differences between a planetary system in which mathematical relationships restrict motions resulting from forces and what goes on in the majority of biological control systems: where, instead of physical processes directly producing or modifying behaviour, there are intervening information processing mechanisms. E.g. sensory systems acquire information and motor control mechanisms use that information in selecting between control alternatives. The control actions may be influenced by information from several sources: e.g. information about an internal need (e.g. for energy-rich food, or for water) can be combined with information about opportunities and obstacles in the environment, or lurking predators. These are unlike processes combining physical forces.

In many cases physical attractive forces increase as distance is diminished, which in the case of physical control leads to increasing acceleration. That could be disastrous for an organism approaching a target: so it is useful be able to detect closeness to the target and use that information to produce deceleration (using stored energy for braking). Where the target is a prey animal that is likely to attempt escape, acceleration right up to contact may be useful, but that requires additional control mechanisms, e.g. producing appropriate motion of claws, or beak or jaws, to capture (or perhaps kill) the prey while avoiding a dangerous impact for the predator.

Even in a very simple single-celled organism, mathematical relationships play a role in control of osmotic pressure, which can be altered by absorption of nutrients or secretion of waste products. One of the important differences between forces and information contents is that forces remain active in the presence of other forces, and their effects combine to produce "resultant" forces, whereas an information item can be temporarily disabled by being ignored, until some urgent task has been completed. So it is essential in organisms to be able to use information to control which other information items have causal powers at which times.

Mathematical competences required for use of such information in selecting and controlling actions are found in many non-human species. These are important aspects of perception and use of what James Gibson called "affordances" in the environment . However Gibson focused on a subset of affordances, mainly those that are relevant to *online* control of actions by the perceiver, whereas humans can perceive and make use of positive and negative affordances for other individuals, and "proto-affordances" – that involve possibilities for change in many aspects of the environment that are not produced by the perceiver and which may be irrelevant to the needs of the perceiver, for example, perceiving that if a certain apple drops off the tree it will not hit the ground because it will land on a rock, whereas if the rock is moved the result will be different. Humans, (and some other organisms?) can also deal with negative affordances that are impossibilities.[7]

Moreover, control relationships can change as an animal grows: genetic mechanisms must somehow enable controlling forces to be varied as sizes, weights, moments of inertia, geometrical relationships and muscular strength change in a growing animal, as D'Arcy Thompson and others have noted.

Besides control based on quantitative relationships, evolution also uses information about *structures* and *structural relationships*, insofar as genetic information plays a role in specifying parts and relationships between parts of developing chemical and physical structures. In humans, another kind of mathematical power is involved in the ability of individuals to develop linguistic competences that make use of complex and varied grammatical structures for information-bearing utterances, and competences that build complex semantic interpretations based on structural relationships (compositional semantics).

Evolution: the blind mathematician

In all these cases the evolutionary and developmental control mechanisms seem to make use of repeated discovery of new structures that can be abstracted from particular instances and later combined with different information in new contexts, while performing complex controlled actions, and while interpreting complex structured perceptual input. Some information about newly discovered abstractions is somehow encoded in genetic mechanisms that allow the information gained to be used in later products of evolution. And in many cases it is crucial that the replication is not a matter of repeated blind copying of the same structure: what is passed on is at a level of abstraction that can be instantiated in different instances, for example (a) when used for continued control of organisms or parts of organisms while growth produces different sizes, weights, size-ratios, moments of inertia, etc. during development and (b) when used in newly emerging species with different details caused by changes in other parts of the genome.

So evolution can be described as "discovering" that new mathematical structures are possible, and that they can be used for new control functions, during reproduction, during development, and during particular actions. Moreover, the evolutionary and developmental histories can be regarded as *proofs* of those mathematical possibilities, even though there is no mathematical mind at work in discovering the theorems or creating the proofs. In that sense biological evolution can be regarded as a "blind theorem prover", rather than Dawkins' "blind watch-maker".

Computers are much faster and more accurate than humans at performing certain kinds of mathematical operations, including numerical and statistical operations, and using arith-

[7] http://www.cs.bham.ac.uk/research/projects/cogaff/misc/impossible.html

metic, algebra and logic to derive conclusions, solve problems and make plans. But not all mathematical discoveries made by humans are based on arithmetic, algebra and logic. Examples include the ancient geometrical and topological discoveries leading up to Euclid's *Elements*[8] made by ancient mathematicians, e.g. Euclid, Archimedes, Pythagoras, Zeno and others; and also the implicit mathematical discoveries regarding syntactic and semantic structures used in human languages.

Even pre-verbal toddlers, and other animals, such as crows, elephants, weaver-birds and squirrels, seem to have spatial (e.g. topological) reasoning competences unmatched by current automated theorem provers and highly trained robots.[9] However, non-human mathematical reasoners and very young humans lack meta-cognitive abilities to reflect on their mathematical discoveries or to explain and defend them against criticism. That limitation may also have afflicted our adult ancestors who first started to make unreflective and unsystematic use of some of their practical reasoning abilities.

I suspect the variety of evolved mathematical competences is far larger and deeper than anyone has noticed. Researchers are currently struggling to sort them out. E.g. there is a notion of *density* (of grains of salt or sand, of leaves, or flocking birds) and a notion of an area or volume occupied with uniform density, which leads to a notion of amount or numerosity that varies both in proportion with the density and with the area or volume, because *total* amount, or numerosity, as opposed to (cardinal) number increases or decreases as either the density, or the area/volume increases or decreases. Understanding that can lead to inferences about increasing numerosity as density remains constant and area or volume increases, or as density increases while area or volume remains constant. This can support judgements of partial orderings of amount or numerosity. But it does not provide a basis for comparing two regions A and B where area or volume of A is greater than that of B, but density of occupancy of B is greater than that of A. Understanding the tradeoff between change in total space and change in density requires a kind of mathematical sophistication that is a pre-cursor to the understanding of integral calculus. I don't know whether anyone understands the mechanisms used in such cases, nor how they produce new competences during development.

Still more mechanism is required for comparisons of areas, volumes, lengths and amounts of stuff occupying areas or volumes. Those require understanding of new kinds of number that occupy spaces between the natural numbers. Ratios, or fractions may seem at first to suffice, so that we can talk of a jug being half, three quarters, five sixths, full etc., but ancient mathematicians discovered (to their horror) that those ratios do not suffice. In particular something more is needed if the side of a square and its diagonal are to be thought of both having definite lengths, as was understood by the time of Euclid's *Elements*.

Limited mathematical abilities of AI systems

Computers are generally thought of as good at doing mathematics. But that is based on a limited view of the scope of mathematics. Computers can perform logical, arithmetical (and therefore statistical) calculations, and operations on text strings, at enormous speeds, because those processes are readily mapped onto operations on bit patterns – especially in combination with random access memory (RAM) operations that allow contents of memory locations to be checked or modified at very high speed (unlike operations on the tape of a Turing machine). Moreover developments in AI, software engineering, theoretical computer science, networking technology, and increasingly sophisticated fabrication processes have expanded the abilities of (networks of) computers so that they increasingly form interfaces to a host of everyday functions, and outperform humans in many activities.

Yet there are many aspects of human (and non-human) intelligence that are not yet modelled on computers, and seem to be particularly hard to model. Many cases go unnoticed by researchers because because they involve not just abilities to act (e.g. catching, throwing, assembling, stacking, etc.) but also abilities to understand possibilities, necessities and impossibilities, which abound in both mathematics and everyday life. These aspects of human and animal intelligence cannot be derived from statistics based learning, nor expressed in probabilistic frameworks, because they are concerned with what is possible, impossible, or necessarily the case, not probabilities. And many are about structures, not measures.

The Turing-inspired Meta-Morphogenesis project includes trying to understand many intermediate forms of information processing between the very simplest organisms and current highly intelligent animals, in the hope that we may stumble across cases that we have never previously thought of that provide new clues regarding mechanisms required and used in brains. The project has already identified a need for evolution to make use of both the fundamental construction kit (FCK) provided by the physical universe and also many derived construction kits (DCKs) produced by biological evolution. Some are concrete construction kits for producing physical and chemical structures and processes. Others are abstract construction kits for producing information structures and information processing mechanisms. It is hoped that eventually we'll understand the sorts of construction kit required to replicate human mathematical intelligence in machines, so that we'll know how to make a baby Kantian robot that can grow up to make discoveries like Euclid.

For more on the meta-morphogenesis project see: http://www.cs.bham.ac.uk/research/projects/cogaff/misc/meta-morphogenesis.html

FOR MISSING REFERENCES SEE:
http://www.cs.bham.ac.uk/research/projects/cogaff/sloman-iccm17.pdf

[8] http://www.gutenberg.org/ebooks/21076

[9] Examples involving human toddlers can be found here http://www.cs.bham.ac.uk/research/projects/cogaff/misc/toddler-theorems.html

Noisy Reasoning: a Model of Probability Estimation and Inferential Judgment

Fintan Costello,[1*] Paul Watts[2]

[1]School of Computer Science and Informatics
University College Dublin, Belfield, Dublin 4, Ireland
[2]Department of Mathematical Physics
National University of Ireland, Maynooth, Co Kildare, Ireland

*To whom correspondence should be addressed; E-mail: fintan.costello@ucd.ie.

Abstract

We describe a computational model of two central aspects of people's probabilistic reasoning: descriptive probability estimation and inferential probability judgment. This model assumes that people's reasoning follows standard frequentist probability theory, but is subject to random noise. This random noise has a regressive effect in probability estimation, moving probability estimates away from normative probabilities and towards the center of the probability scale. This regressive effect explains various reliable and systematic biases seen in people's probability estimation. This random noise has an anti-regressive effect in inferential judgment, however. This model predicts that these contrary effects will tend to cancel out in tasks that involve both descriptive probability estimation and inferential probability judgment, leading to unbiased responses in those tasks. We test this model by applying it to one such task, described by Gallistel et al. (2014). Participants' median responses in this task were unbiased, agreeing with normative probability theory over the full range of responses. Our model captures the pattern of unbiased responses in this task, while simultaneously explaining systematic biases away from normatively correct probabilities seen in other tasks.

We live in a world of nonstationary stochastic processes, where events occur with some associated probability, and this probability itself changes unpredictably over time. To make successful predictions about event occurrence in such a world we must use two distinct types of probabilistic reasoning: descriptive probability estimation (given the events we have seen recently, what is the current underlying probability of A?) and inferential probability judgment (given our current estimate for the probability of A, is the current sample of events consistent with that probability? Or should we infer that the underlying probability of A has changed?). Our aim in this paper is to present a computational model of these two interacting components of probabilistic reasoning.

One revealing aspect of human probabilistic reasoning is the reliable occurrence of a number of systematic biases; biases such as conservatism (Erev et al., 1994), subadditivity (Tversky and Koehler, 1994) and the conjunction fallacy (Tversky and Kahneman, 1983). The model we present here was originally developed to explain these biases in terms of the regressive effect of random noise in reasoning (see Costello and Watts, 2014). Here we extend this model to inferential probability judgment, and show that this model explains patterns of bias seen in such judgment. This model predicts that, in situations that involve both forms of reasoning, these regressive effects will tend to cancel out, leaving subjective probability estimates that tend to agree with the normatively correct values with no systematic bias. Such agreement is seen in recent studies of probability estimation for nonstationary stochastic processes by Gallistel et al. (2014). We demonstrate the model by applying it to Gallistel et al.'s study in detail.

The probability theory plus noise model

Our model assumes that people's probability judgments are produced by a mechanism that is fundamentally rational, but is perturbed in various ways by purely random noise or error, which causes systematic regressive effects. We take $P(A)$ to represent the 'true' probability of event A (that is, the proportion of items in memory that represent A). We take $p_*(A)$ to represent an individual estimate of the probability of event A, and take $\langle p_*(A) \rangle$ to represent the expectation value or mean of these estimates for A: this is the value we would expect to get if we averaged an infinite number of individual estimates for $p_*(A)$. In standard probability theory, the probability of some event A is estimated by drawing a random sample of events, counting the number of those events that are instances of A, and dividing by the sample size. The expected value of these estimates is $P(A)$, the probability of A. We assume that people estimate the probability of some event A in exactly this way: randomly sampling events from memory, counting the number of instances of A, and dividing by the sample size.

If this counting process was error-free, people's estimates would have an expected value of $P(A)$. Human memory, however, is subject to various forms of random error or noise. To reflect this we assume events have some chance $d < 0.5$ of randomly being counted incorrectly: there is a chance d that a $\neg A$ (*not A*) event will be incorrectly counted as A, and the same chance d that an A event will be incorrectly counted as $\neg A$. Given this form of noise, a randomly sampled event will be counted as A if the event truly is A and is counted correctly (with a probability $(1-d)P(A)$, since $P(A)$ events are truly A and events have a $1-d$ chance of being counted correctly), or if the event is truly $\neg A$ and is counted incorrectly as A (with a probability $(1-P(A))d$, since $1-P(A)$ events are truly $\neg A$, and events have a d chance of being counted incorrectly). Summing the probabilities of these two mutually exclusive situations, we get an expected value for a noisy probability estimate of

$$\langle p_*(A) \rangle = (1-2d)P(A) + d \qquad (1)$$

with individual estimates varying independently around this expected value. This average is systematically biased away from the 'true' probability $P(A)$, such that estimates will tend to be greater than $P(A)$ when $P(A) < 0.5$, and will tend to be less than $P(A)$ when $P(A) > 0.5$: a pattern of systematic regression towards 0.5, the center of the probability scale.

Regression, in this model, explains a number of observed patterns of bias in people's probability estimates, such as conservatism, subadditivity, and the conjunction fallacy (see Costello and Watts, 2016a, 2014). This model also makes a number of novel predictions about patterns of bias and agreement with probability theory for various probabilistic expressions; for example, this model predicts that

$$p_*(A) + p_*(B) - p_*(A \wedge B) - p_*(A \vee B) = 0$$

will hold, on average, in people's probability estimates for any events A and B (because in this expression the regressive effects of noise on individual probability estimates $p_*(A)$, $p_*(B)$ $p_*(A \wedge B)$ and $p_*(A \vee B)$ will tend to cancel out). These predictions are strongly supported by experimental results (see Costello and Watts, 2014, 2016b).

Inferential probability judgment

Equation 1 describes the expected value for a probability estimate in one type of probabilistic reasoning task: one where the reasoner sees a sample containing some instances for the event of interest, A, and produces an estimate of the underlying probability $P(A)$. This type of task involves the estimation of a descriptive probability: a probability that summarises the observed sample. We now consider a probabilistic reasoning task where the reasoner is given an explicit probability value p and a sample of n events containing x instances of event A , and judges whether the number of A's seen in the sample is consistent with the given probability. This type of task involves the estimation of an inferential probability $P(x,n|P(A) = p)$: the probability of seeing x A's in a sample of n items, given that $P(A) = p$. Frequentist probability theory provides a normative mechanism for estimating such inferential probabilities: to estimate $P(x,n|P(A) = p)$, draw a series of random samples, each of size n, from a population where $P(A) = p$ and count the proportion of samples that contain exactly x instances of A. This proportion gives an estimate of the probability of the observed sample occurring in a population with $P(A) = p$: the lower this estimate, the less likely it is that the observed sample came from such a population. The expected value of this estimate is given by the binomial probability function

$$P(x,n|p) = \binom{n}{x} p^x (1-p)^{n-x} \tag{2}$$

In our model we assume that people estimate inferential probabilities just as in frequentist probability theory: by drawing a series of random samples of size n from a (simulated) population where $P(A) = p$, and counting the proportion of samples that contain exactly x instances of A. We assume that this counting process is subject to random error; that the count of occurrences of A in a sample is subject to random noise at a rate d (there is d chance that an instance of A in a given sample will be counted as $\neg A$, and d chance that an instance of $\neg A$ in a given sample will be counted as A). Given this random error, with $P(A) = p$ the chance of an instance in a sample being counted as A is equal to $(1-2d)p + d$ (from Equation 1), and so the expected value for this noisy estimate is given by the binomial probability

$$\langle p_*(x,n|p) \rangle = \binom{n}{x} ((1-2d)p + d)^x ((1-2d)(1-p) + d)^{n-x} \tag{3}$$

Note that the probabilities given in Equation 2 and Equation 3 are both binomially distributed with common terms x and n. If we take p_e to be our current estimate of the probablity of A in the population in question, this means that, for any given values of x and n, the associated noisy inferential probability $\langle p_*(x,n|p_e) \rangle$ is exactly equal to another normatively correct inferential probability $P(x,n|p)$ when

$$((1-2d)p_e + d)^x ((1-2d)(1-p_e) + d)^{n-x} = p^x (1-p)^{n-x}$$

When $d \leq p \leq 1-d$, this equality holds for all values of n and x when

$$(1-2d)p_e + d = p$$

or equivalently when

$$p_e = \frac{p-d}{1-2d}$$

This expression is 'anti-regressive', giving values for p_e that are closer to the boundaries 0 and 1 than values of p: p_e is greater than p when $p > 0.5$, and less than p when $p < 0.5$.

Properties of the model

In this section we apply the above model to two sets of experimental results: on conservatism in inferential probability judgment, and on probability estimation in tasks that mix probability estimation and inferential judgment.

Conservatism in inferential judgment

Experimental studies typically investigate inferential probability estimation indirectly, using the related concept of relative probability. These studies involve describing two populations containing complementary proportions of two different types of event. Participants are told that a population has been picked at random, and are then shown a sample of events drawn from the selected population and asked to assess the probability that the sample came from one population rather than the other. Typically these populations are 'bookbags' containing poker chips, with one bag containing, for example, 70% red chips and 30% black (this is the 'red bag'), and the other bag containing the complementary proportions: 30% red chips and 70% black (this is the 'black bag'). Participants are told the distribution of chips in each bag. They are then shown a sequence of n chips and asked, after seeing each chip, to estimate the probability that the sample came from the red bag rather than the black bag, or vice versa (the relative probability of one bag over the other; see Peterson and Beach, 1967, for examples).

Having seen a sample of n events containing x red chips, the normatively correct relative probability that the sample came from the red bag rather than the black bag is given by

$$R(x,n,p) = \frac{P(x,n|p)}{P(x,n|p)+P(x,n|1-p)} = \frac{1}{1+\left[\frac{1-p}{p}\right]^{x}\left[\frac{p}{1-p}\right]^{n-x}} \tag{4}$$

(since the proportion of red chips is p in the red bag, and $1-p$ the black bag). As participants proceed through these tasks they give relative probability estimates that follow the direction required by normative probability theory, but with values of these estimates being ‘conservative’: less extreme than the normatively correct values. This means that if participants see $x > n/2$ red chips in their sample, they give estimates for the probability that the sample came from the red bag that are greater than 0.5 but less than the normatively correct value, while if participants see $x > n/2$ black chips in their sample, they give estimates for the probability that the sample came from the black bag that are greater than 0.5 but less than the normatively correct value. In applying our model to this task we assume, without loss of generality, that red chips are most frequent in the sample and take $x > n/2$ to be the number of red chips in the sample of n events that have been seen, and assume $p > 0.5$ to be the proportion of red chips in the red bag (the bag that participants associate with the sample).

The estimated relative probability, in our model, of a seeing a sample of size n with x red chips coming from the red bag rather than the black bag is given by

$$R_E(x,n,p) = \frac{p_*(x,n|p)}{p_*(x,n|p)+p_*(x,n|1-p)}$$

Note that, since by assumption $p > 0.5$ and $x > n/2$, from Equation 3 we see that $p_*(x,n|p) > p_*(x,n|1-p)$ will tend to hold (subject, of course, to random error: more specifically, the higher the values of x and p the more likely it is that this inequality will hold). This means that $R_E(x,n,p)$ will be greater than 0.5, and these noisy relative probability estimates will follow the direction required by normative probability theory, just as seen in experiments.

For $p > .5$ this function $R_E(x,n,p)$ will be concave for all $x > n/2$ (since as x increases from $n/2$ the probability that the sample came from the red bag increases while the probability that the sample came from the black bag simultaneously falls). Since from Jensen's Inequality we have $\langle f(x)\rangle \leq f(\langle x\rangle)$ for concave functions (the expected value of a concave function is less than that function of the expected value of its argument), we get

$$\left\langle \frac{p_*(x,n|p)}{p_*(x,n|p)+p_*(x,n|1-p)} \right\rangle \leq \frac{\langle p_*(x,n|p)\rangle}{\langle p_*(x,n|p)\rangle + \langle p_*(x,n|1-p)\rangle}$$

and so, rearranging and substituting, we get

$$\langle R_E(x,n,p)\rangle \leq \frac{1}{1+\left[\frac{(1-2d)(1-p)+d}{(1-2d)p+d}\right]^{x}\left[\frac{(1-2d)p+d}{(1-2d)(1-p)+d}\right]^{n-x}} \tag{5}$$

Comparing Equations 4 and 5 we see that $\langle R_E(x,n)\rangle < R(x,n,p)$ when

$$\left[\frac{1+d\left(\frac{1}{p}-2\right)}{1+d\left(\frac{1}{1-p}-2\right)}\right]^{x} < \left[\frac{1+d\left(\frac{1}{p}-2\right)}{1+d\left(\frac{1}{1-p}-2\right)}\right]^{n-x} \tag{6}$$

Since by assumption we have $p > 0.5$ and $x > n/2$ we see that the inequality in equation 6 always holds, and so $0.5 < \langle R_E(x,n,p)\rangle < R(x,n,p)$: estimated relative probability follows the direction required by probability theory, but is conservative, just as observed in people's relative probability judgments. In other words, even though the expected values for the individual inferential probability judgments $\langle p_*(x,n|p)\rangle$ and $\langle p_*(x,n|1-p)\rangle$ are each anti-regressive relative to their corresponding normative values in this model, when combined to produce an overall estimate of relative probability, this estimate is regressive and so reproduces the pattern of conservatism seen in inferential judgment.

Combined estimation and judgment tasks

We finally describe how this model applies to tasks that involve both descriptive and inferential probability estimation. We consider an iterative task that involves the repeated updating of an estimate for the hidden probablity parameter (which may itself randomly change), given a sample of events presented outcome by outcome. People's performance in such tasks were investigated in an experiment by Gallistel et al. (2014), where participants gave repeated estimates of the hidden parameter, p, of a stepwise non-stationary Bernoulli process that controlled the colour of a circle being drawn from a concealed box. On each trial participants clicked a button to draw a new circle from the box. After being drawn, the circle evaporated, and participants could update their estimate for the hidden probability p. Participants were told that the box would sometimes be silently replaced by another box with a different value of p. Participants could update their estimates by either clicking a ”The box has changed!” button (and then picking a new probability estimate), or by adjusting their current probability estimate, or by making no change.

There were two main results from this experiment. First, people's probability estimates were characterised by rapid changes in the estimated value in response to changes in the underlying hidden probability, separated by periods of small adjustments in the estimate (see Figure 1, left side). The speed of detection of a change in the underlying probability p depended on the degree of change: large changes in the underlying probability were detected more rapidly than smaller changes. The median latency for detection of a change in probability estimate in response to a change in the underlying probability was around 12 events in Gallistel et al. (2014).

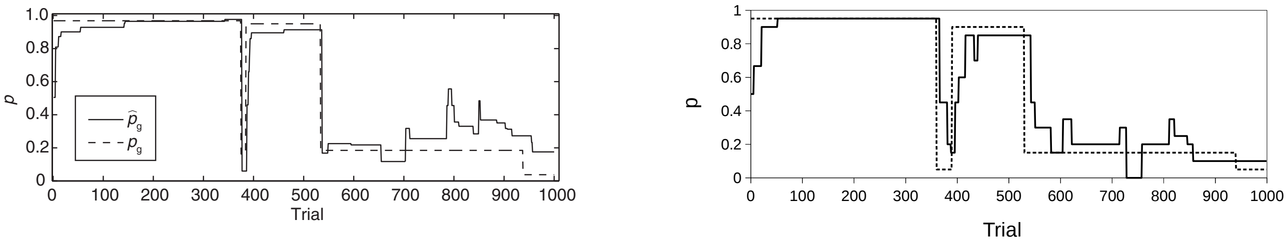


Figure 1. **(Left)** Trial-by-trial true probability (dashed line) and trial-by-trial probability estimate (solid line) for Subject 4, Session 8 in Gallistel et al.'s task (From Fig. 5 in Gallistel et al., 2014, page 102; p_g and $\hat{p}_g$ represent true and estimated probabilities respectively). **(Right)** Trial-by-trial probability estimates produced by our model for the same set of true probabilities. These graphs illustrate the step-hold pattern seen in Gallistel et al.'s task, and show that the model reproduces this pattern.

The second main result was that the relationship between the true probability p and participants estimated probability was essentially that of identity: the median trial-by-trial probability estimates closely tracked the true hidden probability with no systematic bias.

This pattern of agreement with the true probability arises, in our model, due to the cancellation of regressive effects in probability estimation against those in inferential judgment. Suppose we see a series of random samples drawn from a population with a parameter $p = P(A)$, and take p_e to represent our estimate of p (which we repeatedly update as outcomes are presented in the task). This estimate p_e will be subject to random noise, and so will have a regressive average value as in Equation 1. Individual estimates p_e will be adjusted (in a quasi-random walk) in response to inferential probability judgment of the chance of obtaining the currently-seen sequence of outcomes, given our current estimate. This inferential probability judgment will also be subject to random noise, and so will be anti-regressive. This estimate p_e will be least likely to be adjusted when it reaches a value maximally consistent with the average number of counted occurrences of A in the presented sample, and so will tend to fix at that value. Due to random noise, the average number of counted occurrences of A in a sample is equal to $[(1-2d)p+d]n$, and so p_e will fix at the value for which the inferential probability $\langle p_*([(1-2d)p+d]n, n|p_e)\rangle$ is maximised. Since from Equation 3 this inferential probability has a binomial distribution with probability $(1-2d)p_e+d$, it has its maximum value when

$$(1-2d)p_e+d = (1-2d)p+d$$

or equivalently, when $p_e = p$; when our estimate p_e for the underlying population probability equals the true value. In other words, even though descriptive probability estimates are regressive in this model (due to random noise), and inferential probability estimates are anti-regressive (also due to random noise), when these two types of probability judgment are combined these regressive and anti-regressive effects should on average cancel out, leaving estimates that on average agree with the hidden probability parameter p; just as seen in mixed estimation and inferential judgment tasks such as Gallistel et al.'s.

Computational simulation

We apply the model to Gallistel et al.'s continuous probability perception task by assuming that a continuous probability estimate p_e is assessed by counting the frequency of A in n just-observed events (subject to random noise). The parameter n here represents the size of short-term memory available to store just-seen events: we assume n is small, but beyond that make no assumptions about the value n (in our simulations, below, we chose n randomly for each simulated participant, uniformly in the range $5\ldots20$).

We take x to represent the number of occurrences of A in the n most recently observed events and take x_e to represent the noisy count of that number (the count of occurrences obtained with a chance d of randomly miscounting). The expected value of x_e equals $(1-2d)x+nd$, and so the immediately observed probability of A in that sample has the expected value

$$q = (1-2d)\frac{x}{n}+d \qquad (7)$$

On each event occurrence the model makes one of three choices, corresponding to the 3 choices available to participants in Gallistel et al.'s experiment. First, the model may reject the current value of p_e as inconsistent with the number of A's just observed, and update to a new estimate by setting $p_e = q$ (this choice corresponds to clicking "The box has changed!" in Gallistel et al.'s experiment). Second, the model may decide that the underlying distribution has *not* changed but that q is more consistent with the observed number of A's than p_e. In this case the model again updates to a new estimate by setting $p_e = q$: this choice corresponds to a small adjustment of the current probability estimate. Third, the model may decide not to modify p_e.

To decide whether the current estimate p_e needs to be rejected, the model considers the chance of seeing x_e occurrences of A in n samples where the probability of seeing A in those samples is actually p_e. If this chance is too low p_e is rejected. The model assesses this chance in a simple way: by generating 100 random samples (each of size n, with A occurring randomly with probability p_e) and counting the number of A's in each sample. This counting process is subject to random error, with some probability $d < 0.5$ that an occurrence of A will be counted as $\neg A$, or an occurrence of $\neg A$ will be counted as A. The proportion of these samples that contain exactly x_e occurrences of A represents an estimate of the inferential probability $P_E(x_e,n|p_e)$. If this inferential probability is less than some decision criterion T_1 the model concludes that p_e should be rejected because the underlying distribution has changed. The model then changes the new estimate to q.[1].

If the current estimate is not rejected, the model next considers making an estimate adjustment. To decide whether the current estimate p_e needs to be adjusted, the model considers the inferential probability $P_E(x_e,n|q)$: the chance of seeing x_e occurrences of A in n samples drawn from a population where $P(A) = q$. As above, the model assesses this chance by generating 100 random samples (each of size n, with A occurring randomly with probability q) and counting the number of A's in each sample (subject to a rate d of random error in counting). The proportion of these samples that contain exactly x_e occurrences of A represents an estimate of the inferential probability $P_E(x_e,n|q)$. If the difference between this inferential probability and the previous inferential probability is greater than some decision criterion T_2 (that is, if $P_E(x_e,n|q) - P_E(x_e,n|p_e) > T_2$) the model decides that q is a better estimate and changes to a new estimate by setting $p_e = q$. Otherwise the current estimate p_e is left unchanged.

Results

We wrote a computer program implementing this model and tested it by simulating Gallistel et al.'s experiment. On each run of this simulation the model was shown a consecutive sequence of 1000 randomly generated A or $\neg A$ events. After seeing each event, the model either rejected its current probability estimate and changed to the new estimate q; or adjusted its estimate to the new estimate q; or else left its estimate unchanged. Events were generated randomly, with a hidden probability p. The value of p itself changed randomly over the sequence of 1000 events, with the probability that p would change after a given event being set at a constant value of 0.005 (just as in Gallistel et al.'s experiment). The size and direction of a change in the hidden probability were determined by a random choice of the next value from a uniform distribution between 0 and 1, subject to the restriction that $p/(1-p)$, the resulting change in the odds, was no less than fourfold, just as in Gallistel et al. (2014).

To investigate the role of error in descriptive probability estimation and in inferential judgment, we designed the program so that we could set one error rate d for descriptive estimation, and another rate d_s for inferential judgment. We simulated Gallistel et al.'s experiment for 4 different pairs of values for these parameters: Sim A ($d = 0.0, d_s = 0.0$), Sim B ($d = 0.1, d_s = 0.0$), Sim C ($d = 0.0, d_s = 0.1$), and Sim D ($d = 0.1, d_s = 0.1$). We set the criterion parameters T_1 and T_2 to 0.01 and 0.1 respectively in all simulations, since initial tests suggested that these values produced a reasonable rate of adjustment in the model's probability estimates. Each simulation involved 500 'participants' (runs of the model), all with the same values for parameters d and d_s, and each with a value of n (the size of short-term memory) selected randomly from the range $5\ldots20$. Each 'participant' saw a different randomly generated sequence of 1000 events, produced according to a different randomly generated sequence of values of p (as in Gallistel et al., 2014).

Rapid detection of changes The median latency between a change in the hidden probability p and the recognition of that change by the model (via rejection of the current probability estimate) was 10 in simulations A and B, 13 in simulation C and 12 in simulation D. These values agree with the median latency of reported change detection of 12 seen in Gallistel et al. (2014).

High hit rates and low false alarm rates Gallistel et al. (2014) describe a method for computing hit rates and false-alarm rates in participant's responses in their experiment: they found that nine out of ten participants had hit rates in the range $0.77\ldots1$ and false-alarm rates in the range $0.004\ldots0.02$. We used the same method to compute hit rates and false alarm rates across all 'participants' in our simulations. Average hit rates were $0.87, 0.79, 0.81, 0.76$ and false-alarm rates were $0.006, 0.005, 0.005, 0.005$ in simulations A, B, C and D respectively. These agree with the rates seen by Gallistel et al. (2014).

Precision We assess the precision of the model's probability estimates by computing the RMSD between the model's estimate at a given event against the true probability p at that event. These RMSD's between estimated and true probabilities were $0.15, 0.17, 0.17, 0.17$ for simulations A, B, C and D respectively. These were consistent with the corresponding RMSD's for participants in Gallistel et al.'s experiment, which ranged between 0.15 and 0.21.

These three aspects of the model are illustrated in the right of Figure 1. This figure shows trial-by-trial probability estimates produced by the model for one run, with parameter values $d = 0.1, d_s = 0.1, n = 20$. Values of the true probability p were controlled match those in Gallistel et al.'s example. Individual event occurrences in this run, however, were random, and did not follow the precise sequence of event occurrences in Gallistel et al. (2014). This figure shows that the model produces the step-hold pattern seen in Gallistel et al.'s task, with large changes in the estimate when the hidden probability changes, and small adjustments, or no changes, otherwise.

Identity between true probability and median estimates Recall that the noisy frequentist model predicts that noise will have different effects in different probability judgment tasks: when estimating a probability from a sample (descriptive probability estimation), noise will produce regressive effects; when estimating the likelihood of a sample given a probabilty (inferential probability judgment), noise will produce anti-regressive effects; and in tasks that involve both

[1] Note that our decision to use 100 random samples when estimating inferential probabilities here is essentially arbitrary: this number was chosen to allow us to use values for the decision criteria T_1 and T_2 that correspond to standard significance level values such as 0.01 and 0.05. Versions of the simulation that make use of much smaller numbers of samples give essentially the same results as seen here.

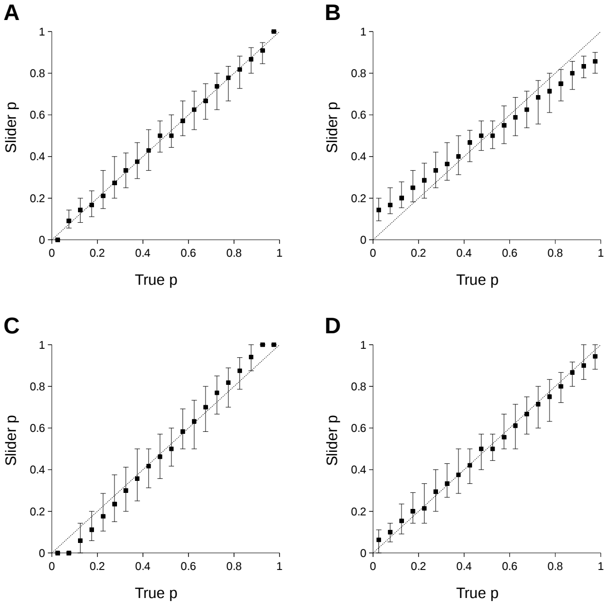


Figure 2. Median (squares) and interquartile intervals (vertical lines) of model's probability estimates plotted against corresponding true probabilities, for different values of the noise parameters: $d = 0.0, d_s = 0.0$(graph A), $d = 0.1, d_s = 0.0$ (graph B) $d = 0.0, d_s = 0.1$ (graph C) and $d = 0.1, d_s = 0.1$ (graph D). The dashed line represents identity.

forms of estimation, these contrasting effects of noise cancel out, producing agreement with the true probability. To test these predictions, for each simulation we calculated the median estimate produced by the model for a given hidden probability value p. The results are shown in the 4 graphs in Figure 2. Graph A gives the results obtained when there is no noise in either descriptive estimation or inferential judgment ($d = 0.0, d_s = 0.0$); the relationship between median estimates and the true probability is one of identity here. Graph B gives the results with noise in descriptive estimation but not inferential judgment ($d = 0.1, d_s = 0.0$), and shows a clear pattern of regression. Graph C gives the results with no noise in descriptive estimation but noise in inferential judgment ($d = 0.0, d_s = 0.1$), and shows a clear anti-regressive pattern. Finally, graph D shows the results obtained when there is the same rate of noise in both components ($d = 0.1, d_s = 0.1$). The relationship between median estimates and the true probability in graph D is one of identity: the effects of noise in the two components have cancelled each other out.

These results show that, if we assume a constant rate of error $d = 0.1$ in both descriptive probability estimation and inferential probability judgment, the probability theory plus noise model produces results that agree closely with those seen in Gallistel et al. (2014). Similar agreement holds for a range of other values of d. These same values of d, however, also produce regressive effects; in our model these regressive effects produce patterns of bias such as conservatism, subadditivity and the conjunction fallacy. In other words, this model may provide a single unified account for systematic bias away from the true probabilies (in some tasks) and for agreement with the true probabilities (in other tasks): an account that depends on a single factor - noise in reasoning.

Conclusions

Our aim in this paper is to present a general model of descriptive probability estimation, of inferential probability judgment, and of the interation between these two processes. This model assumes that people estimate (descriptive and inferential) probabilities using a mechanism that follows standard frequentist probability theory, but is subject to the biasing effects of random noise in the reasoning process. In other work we've shown that this model makes a number of novel predictions about patterns of bias and agreement with probability theory for various probabilistic expressions: predictions which are strongly supported by experimental results (see Costello and Watts, 2016a, 2014, 2016b). Here we show that this model can simultaneously explain the observed patterns of bias seen in people's descriptive probability estimation and inferential probability judgment (which arise in the model due to the regressive effects of random noise), and the observed agreement with the underlying true probability in tasks such as that of Gallistel et al.'s (where the regressive effect of noise in descriptive probability estimation is counteracted by the anti-regressive effect of noise in inferential probability judgment).

References

Brunswik, E. (1955). In defense of probabilistic functionalism: A reply. *Psychological Review*, 62:236–242.

Costello, F. and Watts, P. (2014). Surprisingly rational: Probability theory plus noise explains biases in judgment. *Psychological Review*, 121(3):463–480.

Costello, F. and Watts, P. (2016a). Explaining high conjunction fallacy rates: the probability theory plus noise account. *Journal of Behavioral Decision Making*. In press, available at http://dx.doi.org/10.1002/bdm.1936.

Costello, F. and Watts, P. (2016b). People's conditional probability judgments follow probability theory (plus noise). *Cognitive Psychology*, 89:106–133.

Erev, I., Wallsten, T. S., and Budescu, D. V. (1994). Simultaneous over- and underconfidence: The role of error in judgment processes. *Psychological Review*, 101(3):519–527.

Gallistel, C. R., Krishan, M., Liu, Y., Miller, R., and Latham, P. E. (2014). The perception of probability. *Psychological Review*, 121(1):96.

Peterson, C. and Beach, L. (1967). Man as an intuitive statistician. *Psychonomic Bulletin*, 68(1):29–46.

Tversky, A. and Kahneman, D. (1983). Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review*, 90(4):293–315.

Tversky, A. and Koehler, D. J. (1994). Support theory: a nonextensional representation of subjective probability. *Psychological Review*, 101(4):547.

Cognitive Computational Models for Conditional Reasoning

Marco Ragni (ragni@cs.uni-freiburg.de)
Department of Computer Science, Georges-Köhler-Allee 52,
79110 Freiburg, Germany

Alice Ping Ping Tse (tse@tf.uni-freiburg.de)
Department of Computer Science, Georges-Köhler-Allee 52,
79110 Freiburg, Germany

Abstract

Premises in conditional reasoning consist of an "if" statement (e.g., "if I can catch the bus, I won't be late") and a fact (e.g., I can catch the bus). Such types of simple inference have been studied empirically and formally for about a century. In the past five decades, several cognitive theories have been proposed to explain why humans deviate from predictions of conditional logic. In this article, we (i) describe existing theories, (ii) develop multinomial processing tree (MPT) models for these theories and systematically extend the theories with guessing subtrees to test the predictive power of the cognitive models. The models are evaluated with G^2, Akaike's (AIC) and Bayesian Information Criteria (BIC), and Fisher's Information Approximation (FIA). Mental model theory with directionality for indicative conditionals while the independence model for counterfactuals provide the best fits to data from psychological studies.

Keywords: Human conditional reasoning; multinomial process trees; cognitive theories

Introduction

Suppositional and hypothetical thinking are one of the major cognitive abilities distinguishing humans from other animals. This form of thinking is essential to reflect on past events, hypothesize alternative outcomes, and partially prevent future mistakes. It also facilitates us to make and test assumptions about future outcomes to select actions, responses, precautions or/and procedures. This kind of thought is usually presented as conditional statements in natural language. A conditional statement is usually in the form of "if p then q", expressing a relationship between the *antecedent* p and a *consequent* q. Classical studies of reasoning always use sets of arguments consisting of a conditional and an additional categorical information ("a fact"), i.e., p, $\neg p$, q, $\neg q$. Consider the following problem:

> If I can catch the bus, I won't be late. (*conditional*)
> I can catch the bus. (*categorical*)
> What, if anything, follows?

Almost all reasoners draw "I won't be late" as a conclusion of the two statements. This is an example of a *modus ponens* (MP for short) inference, i.e., to conclude the consequent q (*I won't be late*) from the conditional and the categorical statement p (*I can catch the bus*). Other inference schemas are *modus tollens* (MT for short), i.e., to conclude $\neg p$ (*I cannot catch the bus*) from the conditional and an additional categorical statement $\neg q$ (*I will be late*).

Both schemas MP and MT are classically logically valid. The other two schemas, namely *denial of the antecedent* (DA, to conclude q when $\neg p$ is given as the additional categorical statement) and *affirmation of the consequent* (AC, to conclude p when q is given), are logically invalid but commonly drawn by humans. We focus on deductive reasoning in this article. While the classical logical interpretation is the so-called material implication (if the antecedent is true, the consequence cannot be false) and is easy to define, many psychological experiments have demonstrated that humans deviate from this interpretation. For example, conditional statements in subjunctive grammatical mood (i.e., counterfactual statements) can trigger a different endorsement pattern of the inferences (Byrne & Tasso, 1999; Thompson & Byrne, 2002), compared to statements in indicative mood, i.e., factual statements. It was found that people make inferences from counterfactual conditionals that are less frequently made, for example, when they are asked to reason from the two conditionals: 'If George kept his stock in Company B, then it earned \$1,200 (Byrne & McEleney, 2000)' (factual) and 'If George had kept his stock in Company B, then he would have been better off by \$1,200' (counterfactual). The two negative inferences, namely Modus Tollens (MT) and Denial of Antecedent (DA), had higher endorsement rates in the counterfactual than in the factual condition. We analyze different psychological theories while combining them with an idea from signal detection theory (Macmillan & Creelman, 2004). In visual perception (or memory recognition), the application is to test if humans can correctly identify or not the presence or absence of stimulus in an environment with background noise. We apply this idea to conditional reasoning as follows:

Inference does logically	Response of Ss "not follow"	Response of Ss "follow"
follow	miss	hit
not follow	correct rejection	false alarm

Both inference rules MP and MT are in the category of *logically follows*. Hence if they are applied, we have a *hit* (otherwise, we have a *miss*). If the inference rules AC and DA are not applied, we have a *correct rejection* (but a *false alarm* if applied). Oberauer (2006) has already formalized some theories with multinomial process trees (MPTs) for all the 16 possible answer patterns that are subsets of the four inferences. Hence, his tree included all cognitive processes altogether that led from an input to the 16 leaves which represent the responses. A single fixed guessing tree was inserted to each tree. The models were evaluated by G^2 (see later section for details).

Inspired by the aforementioned idea, we have systematically developed trees for each of the four inference patterns combined with parametrized guessing trees, determining different modes of guessing. That means instead of one tree for all the four inferences, 4 separate trees were constructed for each of MP MT AC and DA according to different cognitive theories. The remainder of this paper is structured as follows: In the next section, we will briefly review current existing theories for conditional reasoning. Then, we will represent these theories as multinomial process trees and systematically vary the amount of guessing for different theories. Then, we will review and report the model fitting results of 45 behavioral experiments (total number of participants $N = 2530$, datasets with the endorsement percentages and N provided for all the four inference rules) on conditional reasoning for simple/classical/indicative conditionals and 12 experimental datasets for counterfactual reasoning, $N = 577$. The cognitive theories formulated as multinomial process theories are then evaluated based on model selection criteria measures – the information criteria AIC and BIC which take additionally the model size into account. A discussion of the best cognitive theory in terms of predictive power concludes the paper.

Cognitive Models of Conditional Reasoning

We introduce some formal notations that we will use in the following sections. A conditional ("if p then q") is written as $p \rightarrow q$ or $(q \mid p)$. Negating a fact p is represented as $\neg p$, the same applies for q. Theories of conditional reasoning can be vastly classified into model-based, e.g., the theory of mental models (Johnson-Laird & Byrne, 1991), rule-based, e.g., the theory of mental logic (Rips, 1994), and theories that build on the idea of Bayesian modeling (Oaksford et al., 2000).

Theory of mental models

The mental model theory (MMT) of conditional reasoning (Byrne & Johnson-Laird, 2009) assumes that for a conditional $p \rightarrow q$, the semantic information of each premise is represented in an initial mental model akin to:

$$\begin{array}{c} p\ q \\ \dots \end{array}$$

Hence both the antecedent and consequent are true in the initial mental model. If p is given, a modus ponens inference, can be drawn and q is derived. In cases where other information is given, e.g., $\neg q$, the model needs to be fleshed out, i.e., other true interpretations of the conditional need to be generated. This leads to the construction of three models eventually:

$$\begin{array}{rrl} p & q & \text{(initial mental model)} \\ \neg p & q & \text{(alternative mental model 1)} \\ \neg p & \neg q & \text{(alternative mental model 2)} \end{array}$$

Hence, an MP-inference is easy, while MT requires more cognitive effort to generate alternative models. MMT explains deviation of human reasoners from the normative logically correct performance by inaction or failure in the search of counterexamples and fleshing out of the initial mental model. The mental model theory does not make any assumption about the directionality of the antecedent and consequent. However, several studies have shown that the directionality of conditionals plays a role in the reasoning process (Evans & Beck, 1981; Barrouillet et al., 2000). We thus include both the classical and extended mental model theory by introducing the assumption about directionality (Oberauer, 2006).

The theory of mental logic

The mental logic theory suggests that humans translate the premises into logical form and use formal rules to draw or prove the conclusion (Rips, 1994; Braine & O'Brien, 1998). However, only MP and MT can be proved by formal rules. MP can be drawn directly with the formal rule of inference but the proof of MT requires several more steps, with reductio ad absurdum (finding a contradiction to the supposition of p). That means, reasoners firstly suppose p after reading the two premises and then find that the conclusion q (by applying the MP inference rule on the supposition) and $\neg q$ (the second/minor premise) are incompatible and thus reject the supposition of p using *reductio ad absurdum* and finally conclude $\neg p$. Errors in reasoning performance are due to misunderstanding of the conditional statements or the application of a wrong rule. As the endorsement of AC and DA rules cannot be explained by mental logic, we use guessing trees for these two inferences. It follows that implementing the mental logic as an MPT is not possible without any guessing trees.

Probabilistic approach: The independence model

Oaksford & Chater (1994) proposed a Bayesian understanding and modeling about how people interpret a conditional and reason about it. Instead of interpreting $p \rightarrow q$ in the classical logical sense – as material implication – human reasoners and their reasoning processes can be modeled as the conditional probability of q given p, i.e., $P(q \mid p)$. In their classical work, they proposed a dependence and an independence model. We focus on the later: the classical independence model (Oaksford & Chater, 1994) consists of two parameters a for $P(p)$ and b for $P(q \mid \neg p)$. To fit experiments, the best parameter values were determined by iterating through the values 0.1, 0.3, 0.5, 0.7, and 0.9 for both a and b as in Table 1 of Oaksford & Chater (1994). The model accepts a specific conditional probability only if the computed value is above a given threshold. We present here an updated version (Oaksford et al., 2000; Singmann et al., 2016). The model assumes that reasoning is done through assessing the probability values of conclusions based on the reasoner's background knowledge. More precisely, when asked to evaluate an inference such as MP, "Given 'If p then q' and 'p', how likely is q?", individuals consult their background knowledge regarding p and q and assess the conditional probability of the conclusion q given the minor premise p. Thus, endorsement E is modeled as $E(\text{MP}) = P(q \mid p)$. The joint probability distribution of p and q, and their negations $\neg p$ and $\neg q$ can be parameterized in terms of three parameters, $a = P(p)$; $b = P(q)$,

Table 1: Oaksford et al. (2000) model of probabilistic conditional reasoning (see, Singmann et al., 2016).

	q	$\neg q$
p	$a \cdot (1-e)$	$a \cdot e$
$\neg p$	$b - a(1-e)$	$(1-b) - ae$

Note. The table represents the joint probability distribution for a conditional, "if p then q" by three parameters: $a = P(p)$, $b = P(q)$, and $e = P(\neg q \mid p)$.

and $e = P(\neg q \mid p)$ as shown in Table 1, which leads to the following model predictions (cp. Singmann et al., 2016):

$$E(\mathsf{MP}) = P(q \mid p) = (1-e) \tag{1}$$

$$E(\mathsf{MT}) = P(\neg p \mid \neg q) = \frac{1-b-ae}{1-b} \tag{2}$$

$$E(\mathsf{AC}) = P(p \mid q) = \frac{a(1-e)}{b} \tag{3}$$

$$E(\mathsf{DA}) = P(\neg q \mid \neg p) = \frac{1-b-ae}{1-a} \tag{4}$$

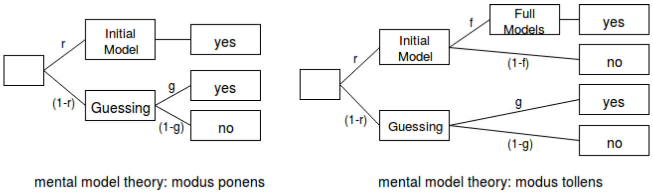
Many experiments, however, provided reasoners with premise information that they were asked to consider as *true*. These formulae can thus be reduced for our problems – the probability of the given fact in the experiments can be assigned as 1. Hence, it holds for the example $P(\text{"I catch the bus"}) = 1$. We can then represent the simplified independence model and transform it into an MPT. Consider modus ponens, with $P(p) = 1$. As $a = P(p)$, we have $a = 1$. Hence, the formula is reduced to $1-e$. For MT, $P(\neg q) = 1$, it follows $P(q) = 0$, and hence $b = 0$. Thus, (2) above is reduced to $(1-ae)$. For $E(\mathsf{AC})$ for AC with $b = P(q) = 1$, Equation (3) is reduced to $a(1-e)$. For $E(\mathsf{DA})$ holds, $a = P(p) = 0$, and (4) above is reduced to $(1-b)$.

The suppositional theory

The suppositional theory proposed by Evans & Over (2004) is a hybrid theory with the application of probability assumption akin to the independence theory and dual process theory, together with some rules in the field of pragmatics. Similar to the independence theory, it emphasizes the cases where we have a high probability of the consequent given the probability of the antecedent. Contextual effect found in a vast amount of studies in conditional reasoning can be explained by pragmatic inferences. Finally, the theory has a dual system incorporated: While immediate inferences (System 1) are solely drawn by the probability account, System 2 inferences are possible and lead to deductively valid answers (cp. Oberauer, 2006).

Theories of Conditional Reasoning as MPTs

Our main goal is to assess the empirical adequacy of the aforementioned cognitive theories. Towards this goal, we need to represent the theories formally. Following a similar approach by Oberauer (2006), we formalize the theories as *multinomial processing tree* (MPT) models (Riefer & Batchelder, 1988).



mental model theory: modus ponens

mental model theory: modus tollens

Figure 1: Two examples for MPTs for the mental model theory (without directionality). The left tree represents the model for the modus ponens; while the right represents the model for the modus tollens, where an additional flesh-out process from the initial mental model is necessary.

MPT models are a class of cognitive models for categorical data that describe observed response frequencies resulting from latent cognitive states. The probabilities that are represented at the edges in the graph for transitioning a cognitive states are estimated from data. At the same time, the aforementioned cognitive theories must explain why the answers of the participants often deviate from the logically correct solution as well. There are two ways of how responses are generated by a reasoner: a *reasoning process*, the process is described and/or predicted by a cognitive theory and a *guessing process*, a process that is not explained in a cognitive theory and, in principle, any possible response can be given.

We represent the reasoning and guessing parts of the theories by multinomial process trees as we outline in the following. For each of the four inference schemas (MP, MT, DA, AC) we develop separate MPT trees. As the theories assume different cognitive processes for the four inferences, we model them as four MPTs. An additional advantage is that this enables us to investigate the differences in processing of the four inferences, see Fig. 1 for an example for MP and MT for the mental model theory. Each root node contains a reasoning parameter r that is responsible for estimating responses that are generated by the reasoning subtree and consequently $(1-r)$ as generated by the guessing subtree. The guessing tree is identical for all theories, with a parameter g to guess a yes-answer and $(1-g)$ a no-answer. In the reasoning tree, we have theory specific nodes that provide specific answers by transitioning through them. In the case of modus ponens (the most simple case), the correct answer can be read out via going along the reasoning branch, where an initial model is built (the model $p \quad q$). In the case of modus tollens, a full explicit model needs to be built (parameter f) for a correct answer and if it is not built $(1-f)$, the decision is solely made by the initial mental model. Hence, these process models do reflect assumed cognitive processes and they are similar enough to formalizations as proposed for syllogistic reasoning (cp. Klauer et al., 2000). In the following sections, we will investigate all previously mentioned theories: The mental model theory with and without directionality, the mental logic theory, the probabilistic theory and the suppositional theory – formalized as MPTs and the respective extended models with a reasoning part and a guessing tree (as described above). we systematically replace the reasoning parts by pure guessing trees (we will describe in detail later in the section “MPT analysis for model comparison”). We can assume that some reasoning subtrees may even have a negative impact, so we systematically eliminate for each theory the reasoning subtree in the extended models.

The Experimental Data

Selection of the experimental studies

We searched the literature and the internet for articles on classic conditional reasoning and non-monotonic conditional reasoning and reporting at least the number of participants as well as absolute number of reasoners or percentages for all four inference types (MP, MT, DA, AC). Extensive search of studies in Pubmed, Science Direct, Google and Google Scholar with the keywords “(conditional reasoning) or (conditionals) or (prepositional reasoning) or (counterfactual) or (alternatives) or (enabler) or (disabler)” was performed. Most of the articles are not suitable for this analysis as the endorsement percentages/frequencies of the four inference rules were not provided. We have included all experiments reporting the four inference types as within subject factor and the questions presented to participants was in the form of “what (if anything) follows (necessarily)?” or “think about what conclusion you can draw from the information” or “assess whether these conclusions follow logically from the information” or “Therefore, ”; with two to four answer options provided to participants. For MP and DA, the answer options were “q”, “$\neg q$“ (and “may or may not q” and “not sure” or “nothing can be concluded”); and “p”, “$\neg p$“ (and “may or may not p”, and “not sure” or “not nothing can be concluded”) for MT and AC. We have excluded 3 experiments with special manipulation of the content of the conditional statements. This selection eliminates as possible the factors due to experimental design. Finally, 16 studies of indicative conditionals (first premise being in the form of “If p then q”; 45 experiments in total) and 6 studies of counterfactual conditional reasoning (12 experiments in total) of adult data were included[1]. We need the frequencies of participants endorsing each inference for our later analysis. For studies providing the endorsement percentages, we computed the frequencies by the percentages and the number of participants.

Reliability of data

We assessed the overall homogeneity for each inference by examining the respective rank orders of the endorsed inferences using Kendall’s coefficient of concordance (W), which ranges from 0, no consensus, to 1, perfect consensus. The datasets are rather homogeneous for both indicative conditionals and counterfactuals, $W = .617$, $p < .001$ and $W = .767$, $p < .001$, respectively.

Conditional reasoning with counterfactuals

Most of the studies on counterfactual reasoning were carried out by Byrne and colleagues (Byrne & Tasso, 1999; Thompson & Byrne, 2002). Usually, conditional statement in subjunctive mood (for native alphabetic languages speakers) were presented to participants to indicate the counterfactual (unreal) property of the situation described in the statement. In these studies, the two negative inferences, DA and MT, usually had a much higher endorsement percentages in the counterfactual than factual condition (but the endorsement percentages of the two positive inferences remained statistically the same). The results support the hypothesis of Byrne and colleagues that reasoners consider two alternatives when they encounter such counterfactual arguments, namely the fact and supposed “fact” (also known as the “presupposed factual reality” and “counterfactual conjecture”). Reasoners constructed already the following two models as the initial mental model and thus the two negative inferences are more likely to be drawn:

$p \quad q$	(counterfactual conjecture)
$\neg p \; \neg q$	(presupposed factual reality)

Besides, there are three other proposals applicable to counterfactual reasoning. For example, Lewis and Stalnaker’s possible world semantics of modal logic (Lewis, 2013; Stalnaker, 1968). They proposed that reasoners assume another world which is most similar to the real world. They perform counterfactual reasoning through reasoning about this most-similar world. However, many researchers criticized the assumption that ordinary people do not judge the closeness of the world/possibility. We have only fitted the models of the adapted mental model theory for counterfactuals (both with and without directionality) as this theory explicitly makes assumptions about cognitive processes in human reasoning.

MPT Analysis for Model Comparison

We fitted each model to the aggregated data via the maximum likelihood method using `MPTinR` (Singmann & Kellen, 2013). The package uses four measures, and the smaller their values, the better the fit between a model and the data: First, G^2 measures the goodness of fit using the maximum-likelihood method, which maximizes the likelihood of the frequencies of observations given the parameter values. It underlies the remaining three measures. Second, the *Akaike information criterion* (AIC) indicates how much information is lost when a model represents the process that generates the data, taking into account both its goodness of fit and number of parameters. Third, the *Bayesian information criterion* (BIC) is a Bayesian analog of AIC that selects the best model from a finite set of them, penalizing models according to the number of their parameters. Fourth, the *Fisher information approximation* (FIA) measures the amount of information that an observed frequency carries about a parameter which models the observation. It provides a good measure of the flexibility of a cognitive model. We evaluated the five cognitive theories and some adaptation of the theories systemically. Firstly, we com-

[1] For studies and MPTs, see: www.cc.uni-freiburg.de/data

pared the MPT implementations of different cognitive theories (the original version) with only one guessing trees extension at the root node, see the $(1-r)$-paths in Fig. 1. Secondly, we systematically eliminated the reasoning subtree, the r-path in 1, 2, or 3 of the inference schemas MT, DA, and AC and kept the guessing tree only. If we replaced the reasoning subtree for the MT we denote it as Guess2. If we replace the DA and AC reasoning subtree, we denote it as Guess34. The reason is to investigate the positive or negative impact of the reasoning tree. The MP reasoning tree is never replaced by a guessing tree as most humans do not have difficulty in drawing MP inference and the sole use of guessing would thus be unnecessary and redundant. We use a PureGuess model which exclusively consists of guessing trees (no reasoning part in any of the four inference schemas) as the base line. The impact of the inference part and how it may disguise the processes can then be evaluated – by comparing if the reasoning part of the theories may add something substantially or not in the information criteria. We repeat the two analysis steps with datasets from counterfactual reasoning to check if the best models for indicative conditionals apply to counterfactual reasoning too.

Theory evaluation

Our first analysis deals with testing the predictive power of the aforementioned cognitive theories for human conditional reasoning. Table 2 reports the results of the four theories (excluding the mental logic as it only makes predictions for MP and MT) and additionally the pure guessing model PureGuess. The lower the G^2, AIC, BIC, and FIA the better the models are. Table 2 shows that the theory (extended with a guessing subtree) which fits best the data is the mental model theory with directionality, which differs from the suppositional theory in a better FIA. The PureGuess model performs worst, i.e., this shows that the reasoning parts of the theories contribute in explaining the data considerably.

Table 2: Results of MPT fits to the aggregated data set of classical conditionals, original version

Model	No. of parameters	G^2	FIA	AIC	BIC
MMTd	4	16.6	3	25	53
SUP	4	16.6	21	25	53
MMT	3	139.5	81	146	167
IND	3	492.5	*	499	520
PureGuess	1	653.1	331	655	662

Note. SUP = suppositional theory; MMTd = mental model theory with directionality; MMT = mental model theory; IND = independence model. PureGuess = pure guessing trees for MP, MT, AC, and DA. * The independence model is not a binary MPT so FIA cannot be computed.

Impact of guessing

In the next analysis, we investigated what happens if we systematically eliminate inference parts according to the theories. Table 3 reports the 5 best fitting theories out of 34 theories. Except the mental logic (with only 2 variants: ML-Guess34 and ML-Guess234), all the other 4 theories have 8 variants (total = 4*8 + 2 = 34). The models are ordered regarding the best values for the information criteria BIC, AIC, and FIA, as the G^2 does not take the number of parameters or the model size into account. Table 3 shows that three models have the best performance regarding the information criteria: The mental model theory with directionality and exclusive guessing at MT (MMTd-Guess2), the mental model theory with exclusive guessing at DA and AC (MMTd-Guess34) and the mental logic with exclusive guessing at DA and AC(as in the original theory by Rips, ML-Guess34). The selection values with these pure guessing trees are much better compared to the original versions. This indicates that theoretical accounts on DA and AC may have to be revised.

Table 3: Results of the MPT fits to the aggregated data set of indicative conditionals by replacing reasoning by guessing.

Model	No. of parameters	G^2	FIA	AIC	BIC
MMTd-Guess2	3	1.6	12	8	29
MMT-Guess34	3	1.6	12	8	29
ML-Guess34	3	1.6	12	8	29
IND-Guess2	4	0	*	8	37
SUP-Guess2	4	0	15	8	37

Note. SUP = suppositional theory; MMTd = mental model theory with directionality; MMT = mental model theory; IND = independence model. Guess2 = MT replaced by the guessing tree; Guess34 = AC and DA with guessing tree only. * FIA cannot be computed for non-binary MPTs.

Counterfactual conditional reasoning

For the third analysis, we tested the performance of the MPT trees of the original theories for conditional reasoning on the counterfactual data. We constructed other sets of MPT models for the mental model theory (cMMT: without directionality and cMMTd: with directionality) according to the aforementioned account of Byrne, which assumes that people build two initial mental models for counterfactuals. But both versions of mental model theories show similar performance. For the counterfactual data, however, the best models are now the independence model with exclusively guessing at the modus tollens (cf. Table 4).

General discussion

While almost all cognitive theories of reasoning aim at explaining human reasoning with conditionals, systematic comparisons are rare. We implemented different theories as multinomial process trees and systematically extended each of the theories with guessing trees and evaluated the goodness-of-fit of (i) the original theories, (ii) the extended models by systematically replacing reasoning subtrees by guessing trees for one to three of the MT, DA, AC-patterns, and (iii) models on

Table 4: Results of the multinomial model fit to the aggregated counterfactual data

Model	No. of parameters	G^2	FIA	AIC	BIC
IND-Guess2	4	0	*	8	31
SUP-Guess2	4	0	12	8	31
MMTd-Guess2	3	6.5	12	13	30
cMMTd-Guess2	3	6.5	12	13	30
ML-Guess34	3	6.5	12	13	30

Note. Models are ordered for the information criteria AIC, BIC, and FIA. Guess2 = MT with guessing tree only; Guess34 = AC and DA with guessing tree only.

counterfactual theories. We performed additionally a literature search and found a high homogeneity of the data. Most of the reported studies asked the reasoner to hold the conditional and the categorical fact as true. The best fitting theory regarding the information criteria AIC, BIC, and FIA (that penalize additional parameters) in case (i) and (ii) is the mental model theory with directionality. For counterfactuals, the best model is the independence model with the modus tollens replaced by pure guessing. Such a difference can be expected as models that perform well in one domain do not necessarily perform well in another. Secondly, the strength of the Bayesian accounts is to represent the difference in strength between antecedent and consequent, which is rarely reflected in most experiments. Another interesting finding is that when comparing models with reasoning and guessing versus guessing alone, some theories are in fact better to assume that some patterns are in fact guessed. In line with the finding of Oberauer (2006), guessing is a very important part in conditional reasoning. The goodness of fit (wrt. AIC and BIC) improves by replacing parts of the theories by guessing in one or more of the three inference rules, especially for MMT. One phenomenon is that reasoner either guess for both AC and DA or MT alone. This might suggest that the processing of MT inference might not be the same as that of the two invalid inferences, AC and DA. Current reasoning theories underestimate the influence of guessing on participant's responses – especially in reasoning with conditionals.

Acknowledgement

The paper was partially supported by DFG-project RA 1934-2/1 and Heisenbergproject RA 1934-3/1.

References

Barrouillet, P., Grosset, N., & Lecas, J.-F. (2000). Conditional reasoning by mental models: Chronometric and developmental evidence. *Cognition*, *75*(3), 237–266.

Braine, M. D., & O'Brien, D. P. (1998). *Mental logic*. Psychology Press.

Byrne, R. M., & Johnson-Laird, P. N. (2009). 'if'and the problems of conditional reasoning. *Trends in Cognitive Sciences*, *13*(7), 282–287.

Byrne, R. M., & McEleney, A. (2000). Counterfactual thinking about actions and failures to act. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(5), 1318.

Byrne, R. M., & Tasso, A. (1999). Deductive reasoning with factual, possible, and counterfactual conditionals. *Memory and Cognition*, *27*(4), 726–740.

Evans, J. S. B. T., & Beck, M. (1981). Directionality and temporal factors in conditional reasoning. *Current psychological research*, *1*(2), 111–120.

Evans, J. S. B. T., & Over, D. E. (2004). *If.* Oxford: Oxford University Press.

Johnson-Laird, P. N., & Byrne, R. M. (1991). *Deduction*. Lawrence Erlbaum Associates, Inc.

Klauer, K. C., Musch, J., & Naumer, B. (2000). *On belief bias in syllogistic reasoning.* American Psychological Association.

Lewis, D. K. (2013). *Counterfactuals*. John Wiley & Sons.

Macmillan, N. A., & Creelman, C. D. (2004). *Detection theory: A user's guide*. Psychology press.

Oaksford, M., & Chater, N. (1994). A rational analysis of the selection task as optimal data selection. *Psychological Review*, *101*(4), 608.

Oaksford, M., Chater, N., & Larkin, J. (2000). Probabilities and polarity biases in conditional inference. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(4), 883.

Oberauer, K. (2006). Reasoning with conditionals: A test of formal models of four theories. *Cognitive Psychology*, *53*(3), 238–283.

Riefer, D. M., & Batchelder, W. H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, *95*(3), 318–339.

Rips, L. J. (1994). *The psychology of proof: Deductive reasoning in human thinking*. MIT Press.

Singmann, H., & Kellen, D. (2013). MPTinR: Analysis of multinomial processing tree models in R. *Behavior Research Methods*, *45*(2), 560–575.

Singmann, H., Klauer, K. C., & Beller, S. (2016). Probabilistic conditional reasoning: Disentangling form and content with the dual-source model. *Cognitive Psychology*, *88*, 61–87.

Stalnaker, R. C. (1968). A theory of conditionals. In *Ifs* (pp. 41–55). Springer.

Thompson, V. A., & Byrne, R. M. (2002). Reasoning counterfactually: Making inferences about things that didn't happen. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *28*(6), 1154-1170.

Beyond the Visual Impedance Effect

Alice Ping Ping Tse[1], **Marco Ragni**[2], **Johanna Lösch**[3]
[1]tse@tf.uni-freiburg.de, [2]ragni@tf.uni-freiburg.de, [3]j.losch@student.maastrichtuniversity.nl
Cognitive Computation Lab, Department of Computer Science, Georges-Köhler-Allee 52,
79110 Freiburg, Germany

Abstract

Whether the mental representation of reasoning problems is spatial or visual (or mixed) in nature has been the subject of considerable debate for years. The visual impedance effect found in Knauff & Johnson-Laird (2002) has provided us with new insights into this question. The study found that the forming of excessive visual images induced by the premises can impede relational reasoning. This study aimed at investigating the factor of complexity on the visual impedance effect in two folds, namely number of term series (i.e., total number of premises plus the conclusion) and whether the entities in the premises are presented in a continuous manner (i.e., whether the subject of the argument is the same as the object of the previous argument). In line with previous studies, relational category, number of term series and successiveness were the main factors of the response time. Results of the parameter estimation by generalized estimating equation showed that visual relations, 5-term series and discontinuous problems were the only significant parameters. The results again suggested that irrelevant visual images can hinder reasoning processes, in addition to the complexity of the problem. We proposed a combined cognitive model of ACT-R and PRISM for the findings in this study.

Keywords: Visual impedance effect; generalized estimating equation; PRISM; ACT-R.

Introduction

Reasoning is one of the complex cognitive processes which requires several fundamental processes and the role of working memory is irrefutable (e.g., García-Madruga et al., 2007; Barrouillet & Lecas, 1999). The multicomponent model of working memory proposed by Baddeley and colleagues is the most influential one. They proposed a model with four components, namely the central executive, the phonological loop, the visuospatial sketchpad and the episodic buffer (Baddeley & Hitch, 1974; Baddeley, 2000). The central executive acts as the managing unit which supervises the integration of information and coordinates the two "slave systems" (the phonological loop and visuospatial sketchpad, which act as short-term storage units for information) and other cognitive processes. The phonological loop stores phonological information while the visuospatial sketchpad stores visual and spatial information, which can be further divided into the visual subsystem (for visual information such as color and shape) and the spatial subsystem (for information related to location or space, e.g., Sima et al., 2013). This distinction is supported by the findings in behavioral experiments with a dual-task paradigm (that visual tasks were more hindered by a visual secondary task than a spatial one but spatial tasks by a spatial secondary task than a visual one (e.g., Klauer & Zhao, 2004)) as well as functional brain imaging studies (e.g., Smith et al., 1995).

There are proposals of individual differences in reasoning strategy that some people reason with a verbal and propositional while others reason with visuospatial representation (e.g., Ford, 1995; Bacon et al., 2007). However, the focus of this study will be on the representation of the problems instead of the strategy (see also the two-streams hypothesis – the ventral and dorsal streams for visual and spatial locational processing respectively (Goodale & Milner, 1992; Smith et al., 1995)). On the other hand, the presence of the two separate subsystems of the visuospatial sketchpad may suggest two possible representation formats of the reasoning problems during processing.

Some studies showed that the mental representation of reasoning problems can be either visual or spatial (Landau & Jackendoff, 1993; Huttenlocher, 1968). If the mental representation is visual, visual-rich materials should enhance the reasoning performance as some extra effort is required to represent visually opaque objects. However, many studies did not find this results (e.g., Sternberg, 1980). For example, similar behavioral performance was found for both abstract and concrete problems (Johnson-Laird et al., 1989; Newstead et al., 1986). However, the first study on visual images and relational reasoning by De Soto et al. (1965) found that the ease of visualizing the materials was a significant factor of the performance in their deductive reasoning experiment of "3-term series" problems:

Ann is taller than Beth.
Cath is shorter than Beth.
Who is tallest?

Only the relations between A and B as well as B and C were given. Participants had to infer the relation between A and C from the two given relations. In other studies, participants were asked to evaluate the validity of the conclusion (e.g., Ann is taller than Cath) given the premises. The results were replicated in the studies by Shaver et al. (1975) and Clement & Falmagne (1986).

On the other hand, Knauff & Johnson-Laird (2002) found the exact opposite phenomenon – the visual impedance effect that excessive representations of visual information can slow down the reasoning process. They then concluded that representation of reasoning problems should be spatial in nature. Visual representations are assumed to be visual mental images which resemble the real object (Knauff & May, 2006); while spatial representations relate objects in spatial organization. By default, visual relational problems are represented visually while spatially for spatial problems after the comprehension process of the reasoning problems. Therefore, spatial problems are ready for reasoning processing if reasoning takes entities which are represented spatially. On the other hand, visual problems need an extra step to represent them spatially. Furthermore, the irrelevant visual information (to reasoning processing) can take up cognitive resources for and compete with the reasoning processes and thus causing an overall longer processing time. They argued that the opposite results in some previous studies were due to the neglect of the two possible different representations – visual and spatial: that some "visual" items were also "spatial". Therefore, the opposite results to the visual enhancement proposal were found when the target words of the relational problems were carefully controlled to visual (e.g., cleaner-dirtier: easy to visualize but difficult to form a spatial representation), visuospatial (e.g., above-below: easy to form spatial representation) and control (e.g., better-worse: difficult to visualize and form a spatial representation). They then revised the mental model theory[1] that mental models should be iconic spatial in nature, rather than visual (in addition to the fact that depicting abstract relations in a spatial manner is plausible and possible). The hypothesis was supported by further studies with blind people Knauff & May (2006) and on individual differences ("verbalizers" versus "visualizers" by Castañeda & Knauff, 2013).

Another source of reasoning difficulty depends on the number of mental model operations that are necessary to integrate information into a spatial model. The PRISM-model (Ragni & Knauff, 2013) formalizes these operations to build a preferred and alternative mental models (in cases which the descriptions allow for several models) and it provides a cognitive complexity measure[2] to predict and explain reasoning difficulty for model-based complexity.

This study aimed to investigate the influence of task complexity on the visual impedance effect in relational deductive reasoning tasks. We varied the task complexity in two ways. Firstly, besides the 3-term series problems, we included also the 4-term (3 premises) and 5-term series (4 premises) problems. Difficulty is expected to increase with the number of relations per task. Secondly, difficulty in terms of premises integration was varied. Discontinuous and continuous premise order problems (successiveness) were presented. For continuous problems, the subject of the premise sentence was the same as the object of the previous premise, except for the last premise preceding the conclusion, see Table 2 for examples. For continuous tasks, the information of the first premises can be integrated directly but in discontinuous tasks, the integration process cannot be proceeded until the presentation of the last premise. The information of previous premises has to be stored in the working memory for further processing. Therefore, problems presented in discontinuous order are expected to require more cognitive effort. We analyzed these three factors of difficulty/complexity (relational category, number of terms and successiveness) in terms of both the percentage of correct responses and response time. We hypothesized that only the number of terms and successiveness would affect the percentage of correct responses but not the relational category. However, all the three factors were expected to affect the response time as they can all slow down the reasoning processes. If the two complexity factors can affect the response time of visual relational problems, this might suggest that the underlying processes are the same, or at least involved.

The remainder of the article is structured as follows: In the following section, we present two experiments that investigate the visual impedance effect and model complexity. We then present a combined cognitive model in the next section. A general discussion concludes the article.

Experiments

One criticism of previous studies is that the target words in the experiments were not good examples of the categories, i.e., some "visual" items were also "spatial". In order to eliminate this potential limitation, we performed a preliminary survey to select two pairs of words for each of the four categories. We selected 37 pairs of antonyms (74 words) that may belong to the categories according to our intuition. They are all frequent words in daily usage. Two pairs of words for each category were selected to construct the reasoning problem set according to the results of the word survey.

Word Survey

Purpose of this word survey is to select the best two pairs of antonyms of adjectives describing the 1. visual, 2. spatial, 3. visuospatial, and 4. non-visual and non-spatial features of a noun. Knauff & Johnson-Laird (2002) have shown that the difficulty in forming spatial representation of words can be independent of that in forming visual images. Word pairs were then selected according to the criteria stated in Knauff & Johnson-Laird (2002).

Participants

The survey was performed on the Amazon Mechanical Turk platform. 97 native English speaker participated in the study (mean age = 36.8 years, SD = 13.48; 56 females). They received a nominal fee for their participation.

Materials and Design

We tested 37 pairs of antonyms such as fatter-thinner and below-above. Each trial consisted of one pair. Participants had to rate the ease of forming detailed visual images and spatial representation (position of the object in space) for each of the two words, i.e., giving four ratings in each trial. They were asked to move separate sliders for the two scales to indicate their responses, from 0 (extremely easy) to 100 (extremely difficult).

Results

We ranked the 74 words according to the means and medians of the two ratings (visual and spatial) respectively and

[1] The Mental Model Theory (e.g., Johnson-Laird & Byrne, 1991) proposes that reasoners apply three stages: premise comprehension, a subsequent integration to form an initial model, and conclusion validation by searching for counterexamples.

[2] The model: `http://spatialmentalmodels.appspot.com/`

Table 1: Mean rating for the ease of forming a visual image and a spatial representation for the two pairs of words.

		Visual	Spatial
	Word Pairs	(mean)	(mean)
Visual	prettier–uglier	15.9	72.2
	cleaner–dirtier	15.3	67.7
Visuospatial	above–below	17.8	18.0
	in front of–behind	17.6	15.5
Spatial	right–left	25.4	17.6
	east–west	28.8	19.0
Control	kinder–crueler	42.0	68.8
	braver–more cowardly	43.6	70.7
Overall		29.7	51.7

Note. Range: 0 (extremely easy) to 100 (extremely difficult).

selected two pairs for each of the four categories. The two visual pairs are at the highest rank for the visual rating but are relatively low for the spatial rating. We could not find two pairs which were high for the spatial rating but very low for the visual rating and thus we chose two pairs which are ranked high for the spatial rating but not too high for the visual rating as the spatial pairs. The visuospatial pairs are at a high rank for both the visual and spatial ratings. We then chose two control pairs which were ranked relatively low for both the visual and spatial rating. Please refer to Table 1 for the ratings of the words we selected for each of the categories.

Reasoning Experiment

In this experiment, we investigated if the two other factors (complexity) of reasoning processing can affect reasoning performance and along with the visual impedance effect. The complexity was varied by increasing the number of premises (3-term, 4-term and 5-term series) and the successiveness of the entities in the problems (continuous or discontinuous). Visual relational category problems were expected to have the longest response time, then the control, and finally the visuospatial and spatial problems. 5-term discontinuous visual problems were expected to have the longest response time among the 24 possible condition combinations (4 category x 3 different term series x continuous or discontinuous).

Participants

96 native English speakers, different from the previous word survey, participated in the experiment (mean age = 36.35 years, SD = 11.96; 65 females) via the same platform. They received a nominal fee for their participation.

Materials and Design

We used the 8 pairs of words listed in Table 1 to construct 48 relational reasoning problems. We adopted a 4 (category: visual, spatial, visuospatial, and control) x 3 (term: 3-term, 4-term and 5-term) x 2 (continuous or discontinuous) x 2 (valid or invalid) design. All factors were within-subject as we were not interested in individual differences in this study. Therefore, there were 12 problems for each category and 16 problems for each term condition. We used the same animal-nouns for all the problems: ape, cat and dog for 3-term problems, ape, cat, dog and bird for 4-term problems and ape, cat, dog, bird and fish for 5-term problems. Half of the problems were continuous that the subject (the first noun) of the premise was the same as the object (last noun) of the previous premise, except for the last premise that either the subject or object was the same as the one in the previous premise (depending on the validity of the problem). Please refer to Table 2 for examples of the problems. The relational target word of the last premise was always the antonym of the previous premise(s). Each term in the antonym pair was presented equally often in the premises and conclusions. Each premise and the conclusion were presented on separate screens and the participants had to press the spacebar in order to read the next premise or the conclusion (offline self-paced design). The premises were presented in black letters while the conclusion was in red. Participants were asked to evaluate whether the given conclusion followed necessarily from the premises, by pressing the J (for yes) and D (for no) keys. Two practice trials were presented before the experiment. The relational target words (older-younger and faster-slower) in the practice trials were not repeated in the experiment. Reading time for each premise, response time to the conclusion and the response were recorded.

Results

Regarding the percentage of correct responses, the overall result was 69%. We have performed a 4 (category: visual, spatial, visuospatial and control) x 2 (term: 4-term or 5-term) x 2 (successiveness: continuous or discontinuous) x 2 (validity: valid and invalid) repeated measure ANOVA, excluding the 3-term problem (as there was no continuous and discontinuous distinction for the 3-term problems). The repeated measures ANOVA with a Greenhouse-Geisser correction showed that only successiveness and validity were significant main within-subject effects for percentage of correct responses, successiveness: $F(1,98) = 12.493, p = .001$, $\eta_p^2 = .113$; validity: $F(1,98) = 7.564, p = .007$, $\eta_p^2 = .072$. The post-hoc Wilcoxon Signed Ranks Test showed that valid problems had a significantly higher percentage of correct responses than invalid ones as well as continuous than discontinuous problems, $Z = 2.350, p = .019$ and $Z = 3.546, p < .001$, respectively. In line with previous studies, we did not find any effect due to the relational category of the target words. However, it was unexpected that the number of terms did not affect the percentage of correct responses.

Only response times for correct responses were included in the following analyses. We excluded also the response times which were not within $+/-$ 2 standard deviations of the mean response time of individual participant. Regarding the response time, we firstly tested if the validity was a significant factor of the response time. The Wilcoxon Signed Ranks Test showed that there was no significant difference

Table 2: Examples of 3-term and 5-term series. Conclusions can be valid (= v) or invalid (= i).

	3-term series	5-term series	
		Continuous order	Discontinuous order
Premise 1	The ape is dirtier than the cat	The bird is right of the ape	The ape is braver than the cat
Premise 2	The dog is cleaner than the cat	The ape is right of the fish	The dog is braver than the fish
Premise 3		The fish is right of the cat	The bird is braver than the ape
Premise 4		The dog is left of the cat	The dog is more cowardly than the cat
Conclusion	The ape is dirtier than the dog? (v)	The bird is right of the dog? (v)	The fish is braver than the bird? (i)

between the response time for valid and invalid problems, $Z = .837, p = .402$. Therefore, we aggregated these 2 conditions in order to simplify the following analyses. As there was no successiveness distinction for the 3-term condition, we performed a 4 (category: visual, spatial, visuospatial and control) x2 (term: 4-term or 5-term) x2 (successiveness: continuous or discontinuous) repeated measure ANOVA, excluding the 3-term problems. The repeated measures ANOVA with a Greenhouse-Geisser correction showed that category, term and successiveness were all significant main effects of response time, category: $F(1.899, 34.178) = 13.162, p < .001$, $\eta_p^2 = .422$; term: $F(1,18) = 36.680, p < .001$, $\eta_p^2 = .671$; successiveness: $F(1,18) = 21.032, p < .001$, $\eta_p^2 = .405$. However, none of the interaction effects were significant. We then performed a separate one-way repeated measure ANOVA with a Greenhouse-Geisser correction for the 3-term problems regarding the factor category. The results suggested that response times for different category conditions were significantly different, $F(2.316, 215.426) = 7.504, p < .001$, $\eta_p^2 = .075$. The response times are shown in Table 4.

As our aim was not to investigate individual differences, but rather factors on reasoning processing and related to visual impedance effect, all the aforementioned factors were within-subject factors. Regression analysis is not recommended for repeated measure data. We then used a generalized estimating equation (GEE) to estimate the parameters on response time. GEE was used to estimate the parameters of a generalized linear model with a possible unknown correlation between outcomes (Pickles, 1998) and is suitable for repeated measure data. GEE attempts to treat the within-subject covariance as a nuisance and model only the mean response. In additional, it is not necessary to specify the covariance structure correctly for reasonable estimates of regression coefficients and standard errors as it is not modeled and thus is relatively more flexible. GEE was used to analyze the data with response time as the dependent variable and category, term and successiveness as the factors. In order to simplify the model, only the 3 main effects were included, as the results of the repeated measure ANOVA suggested no significant interaction effects. The model suggested that for the response time $RT = 112118 + 2224 \cdot visual + 2354 \cdot (5 - term) + 1145 \cdot discontinous$. Visual, 5-term and discontinuous problems required longer processing time with 5-term and visual required about the double time than discontinuous problems. We could actually hypothesize the relative response time for the problems of each of the condition combinations according to the equation, but space in this article does not permit an extensive discussion here. Please refer to Table 3 for the results.

A Cognitive Model

We investigated in one experiment two different often replicated effects: The (dis-)continuity effect (successiveness) and the visual impedance effect that both have an implication on the reasoning time. While the former can be explained by the PRISM-model (Ragni & Knauff, 2013), the later has been recently explained by a model of memory effect in ACT-R (Albrecht et al., 2015). PRISM is a computational cognitive model that can be used to simulate and explain how spatial mental models are constructed, inspected, and varied in a spatial array that functions as if it were a spatial working memory. A spatial focus inserts tokens into the array, inspects the array to find new spatial relations, and relocates tokens in the array to generate alternative models of the problem description, if necessary. Each of these focus operations in PRISM imply 1 standard cost (see website in footnote 2), defining a cognitive complexity measure. It does not reflect, however, how visual or spatial the relations are – it calculates for each problem the costs necessary for performing the associated mental model operations to reason about this problem.

While the ACT-R model of memory effect does not necessarily reflect mental model operations, it is compatible with PRISM (Albrecht et al., 2015). The cost function for PRISM, combined with an additional costs for a visual representation (1 standard unit) and a reduction cost of -0.5 standard unit for spatial or visuospatial representation compared to the control problems (as the spatial representation is already built during the comprehension process), can be expressed by the following cost function:

$$\begin{aligned} cost(x) = \quad & 5 + 1 \cdot visual - 0.5 \cdot visuospatial - 0.5 \cdot spatial \\ & + 2 \cdot (\#terms - 3) + 2 \cdot discontinuous \end{aligned}$$

with four boolean variables (i.e., 1 if *true* else 0) *visual*, *visuospatial*, *spatial*, and *discontinuous*. *#terms* represent the number of terms in the problem. While we are working on a computational cognitive modeling incorporating the cognitive complexity measure in the formula, the correlation between the respective behavioral response times and values (costs) calculated by the cost function is very high (Kendall's $\tau = .974$, $p < .001$). This makes the effects on building and

Table 3: Results of the parameter estimation by the Generalized Estimating Equations.

Parameter Estimates							
Parameter	β	Std. Err.	95% Wald CI		Hypothesis Test		
			Lower	Upper	Wald χ^2	df	p-value
(Intercept)	12051	669	10739	13363	324.172	1	.000
Visuospatial	338	631	-898	1574	.287	1	.592
Visual	2633	772	1121	4146	11.647	1	.001
Spatial	-21	355	-718	676	.003	1	.953
Control	Redundant						
5-Term	2482	446	1607	3356	30.948	1	.000
4-Term	Redundant						
discontinuous	1151	338	488	1814	11.593	1	.001
continuous	Redundant						
(Scale)	123440405						

Note. β = beta coefficient; Std. Err. = Standard Error; Wald CI = Wald confidence interval.

Table 4: Mean response times (in ms) for correctly answered problem. Response times not within +/- 2 standard deviations of the mean time of individual participant were excluded. Cont = continuous; disc = discontinuous.

	3-term	4-term		5-term	
Category		cont	disc	cont	disc
Control	9727	13791	15490	16888	18156
Spatial	9056	12254	16140	16406	17723
Visual	10972	15219	17556	19765	21016
Visuospatial	8948	12833	16295	15352	17484

keeping visual models and cognitive operations comparable on the common ground of mental operations.

General Discussion

Knauff & Johnson-Laird (2002) suggested that visually-rich materials can evoke extra cognitive loading due to the processing of unnecessary visual information in reasoning which causes the visual impedance effect; in contrast with the common belief that reasoning problems with materials which are easier to be visualized are easier to solve, having a faster processing speed. Representation of reasoning problems should be spatial rather than visual according to the results. Otherwise, visual-relational problems should be faster to solve if the problems are represented visually. In line with the results of Knauff & Johnson-Laird (2002), we found a slower response time for visual problems even when the complexity of the problems was varied. The same trend was found in 3-term series as well as 4-term and 5-term series for both continuous and discontinuous problems. Moreover, we found that the number of term and successiveness of the entities in the problems were also factors of the response time. Only successiveness and validity were the significant factors for the percentage of correct responses. However, validity does not affect the response time of the problem.

It might be possible that the phenomenon is due to more spreading activation in memory of more (visual) features for visually-rich relational terms Albrecht et al. (2015). As more linked features are activated, the access time is longer. However, we found a stronger visual impedance effect for 5-term discontinuous problem than 5-term continuous problems (with Wilcoxon Signed Ranks Test, difference between response time for visual and control problems were significant, for continuous: $Z = 2.262, p = .024$; for discontinuous: $Z = 2.809, p = .005$). The impedance effect is quite possibly affecting the same underlying processing, i.e., hindrance in reasoning processing. If the more spreading of activation in memory solely causes the visual impedance effect, discontinuous problems should not have an even slower response time as the items in both kinds of problems were the same and they should evoke similar activations in memory.

As suggested by the mental model theory, after representing the premises into a iconic spatial array with the semantics process, where the visual impedance effect is induced, the reasoning processes (i.e., the construction of iconic spatial mental model(s), validity evaluation, and revision of the initial model) are nonlinguistic in nature. This hypothesis was reflected in our results. We did not find any difference between percentage of correct responses for visual relational problems and other relational categories. On the other hand, successiveness and validity were significant factors. Invalid problems had a lower percentage of correct responses as well as discontinuous problems. It was out of our expectation that 5-term series problems were not more difficult. The difference between 4-term and 5-term series problems was very small (4-term: 67%; and 5-term: 65%). The results suggested that the ease in representing or forming the integrated representation is a more important issue in reasoning.

Results of the GEE analysis were in line with those of the repeated measure ANOVA of the response time. Both results suggested that 5-term series discontinuous visual problems had the slowest response time. Furthermore, the regression coefficient of the visual problems was biggest among the three significant parameters which might suggest that the impedance due to visual processing was a more important factor for the impedance effect on reasoning than the number of premises and successiveness of the entities.

In conclusion, our study replicated the visual impedance effect for problems with different complexity. We found that visual relational terms can slow down reasoning processing but spatial relations can enhance reasoning in some of our conditions. This supported the hypothesis of Knauff & Johnson-Laird (2002) that the mental representation of reasoning problems should be spatial in nature. Also, the factors affecting the complexity of reasoning problems are important for the response time. Limitations of our study include that we could not find any spatial relation terms that are easy to represent spatially but difficult to visualize. In our word survey, we found many pairs which are easy to be visualized but only few are rated to be easy to envisage spatially. However, this is one of the confounding factors of many previous studies. Further studies include the correlation between the easiness in visualizing the problem and the strength of the visual impedance effect. The effect can also be tested with other kinds of reasoning problems such as syllogisms.

Acknowledgement

Supported by DFG-projects RA 1934-2/1 and RA 1934-3/1.

References

Albrecht, R., Schultheis, H., & Fu, W.-T. (2015). Visuo-spatial memory processing and the visual impedance effect. In D. C. Noelle et al. (Eds.), *Proceedings of the 37th Annual Meeting of the Cognitive Science Society.* Pasadena, California: Austin, TX: Cognitive Science Society.

Bacon, A. M., Handley, S. J., & McDonald, E. L. (2007). Reasoning and dyslexia: A spatial strategy may impede reasoning with visually rich information. *British Journal of Psychology*, *98*(1), 79–92.

Baddeley, A. D. (2000). The episodic buffer: a new component of working memory? *Trends in Cognitive Sciences*, *4*(11), 417–423.

Baddeley, A. D., & Hitch, G. (1974). Working memory. *Psychology of learning and motivation*, *8*, 47–89.

Barrouillet, P., & Lecas, J.-F. (1999). Mental models in conditional reasoning and working memory. *Thinking & Reasoning*, *5*(4), 289–302.

Castañeda, L. E. G., & Knauff, M. (2013). Individual Differences, Imagery and the Visual Impedance Effect. In M. Knauff, M. Pauen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th Annual Meeting of the Cognitive Science Society.* Berlin, Germany.

Clement, C. A., & Falmagne, R. J. (1986). Logical reasoning, world knowledge, and mental imagery: Interconnections in cognitive processes. *Memory & Cognition*, *14*(4), 299–307.

De Soto, C. B., London, M., & Handel, S. (1965). Social reasoning and spatial paralogic. *Journal of Personality and Social Psychology*, *2*(4), 513.

Ford, M. (1995). Two modes of mental representation and problem solution in syllogistic reasoning. *Cognition*, *54*(1), 1–71.

García-Madruga, J. A., Gutiérrez, F., Carriedo, N., Luzón, J. M., & Vila, J. O. (2007). Mental models in propositional reasoning and working memory's central executive. *Thinking & reasoning*, *13*(4), 370–393.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, *15*(1), 20–25.

Huttenlocher, J. (1968). Constructing spatial images: A strategy in reasoning. *Psychological Review*, *75*(6), 550.

Johnson-Laird, P. N., & Byrne, R. M. (1991). *Deduction*. Lawrence Erlbaum Associates, Inc.

Johnson-Laird, P. N., Byrne, R. M., & Tabossi, P. (1989). Reasoning by model: The case of multiple quantification. *Psychological Review*, *96*(4), 658.

Klauer, K. C., & Zhao, Z. (2004). Double dissociations in visual and spatial short-term memory. *Journal of Experimental Psychology: General*, *133*(3), 355.

Knauff, M., & Johnson-Laird, P. (2002). Visual imagery can impede reasoning. *Memory & Cognition*, *30*(3), 363–371.

Knauff, M., & May, E. (2006). Mental imagery, reasoning, and blindness. *The Quarterly Journal of Experimental Psychology*, *59*(1), 161–177.

Landau, B., & Jackendoff, R. (1993). Whence and whither in spatial language and spatial cognition? *Behavioral and Brain Sciences*, *16*(02), 255–265.

Newstead, S. E., Pollard, P., & Griggs, R. A. (1986). Response bias in relational reasoning. *Bulletin of the Psychonomic Society*, *24*(2), 95–98.

Pickles, A. (1998). Generalized estimating equations. *Encyclopedia of biostatistics*.

Ragni, M., & Knauff, M. (2013). A theory and a computational model of spatial reasoning with preferred mental models. *Psychological Review*, *120*(3), 561.

Shaver, P., Pierson, L., & Lang, S. (1975). Converging evidence for the functional significance of imagery in problem solving. *Cognition*, *3*(4), 359–375.

Sima, J. F., Schultheis, H., & Barkowsky, T. (2013). Differences between spatial and visual mental representations. *Frontiers in psychology*, *4*.

Smith, E. E., Jonides, J., Koeppe, R. A., Awh, E., Schumacher, E. H., & Minoshima, S. (1995). Spatial versus object working memory: PET investigations. *Journal of Cognitive Neuroscience*, *7*(3), 337–356.

Sternberg, R. J. (1980). Representation and process in linear syllogistic reasoning. *Journal of Experimental Psychology: General*, *109*(2), 119.

Implementing Mental Model Updating in ACT-R

Sabine Prezenski (sabine.prezenski@tu-berlin.de)
Cognitive Modeling in Dynamic Human-Machine Systems, Department of Psychology and Ergonomics, Technical University Berlin, Berlin, Germany

Abstract

This paper demonstrates how mental models and updates of mental models due to system changes can be modeled with the cognitive architecture ACT-R using explicit mechanisms. The mental model building and updating is modeled with a representation chunk and a control chunk. The representation chunk holds the strategy, the expected outcome and an evaluation mechanism of the strategy. The control chunk holds information over environmental conditions and the learning history. This modeling approach was developed and tested for smartphone application tasks and then implemented in a dynamic decision-making task investigating strategy development with complex stimuli. The later task used different multi-feature auditory stimuli material. The modeling approach explained data of participants in the smartphone studies very well and met the trends found in the dynamic decision-making task.

Keywords: *ACT-R; mental model updating; general model; learning; dynamic decision-making, applied*

Introduction and Theory

Our behavior is guided by our internal representation of tasks and situations (Norman, 1983). However, such representations or mental models are not static but they change and are adjusted, due to experience gain, environmental changes etc. Understanding how people update or adapt their mental model is relevant in many fields, from updates in technical systems to real-life tasks that require strategy learning and dynamic decision-making. The later investigates serial decisions. Such decisions are dependent on previous decisions and are made under time constraints in a changing environment (Edwards,1962; Gonzalez, 2014). Dynamic decision-making can be seen as a continuous cycle of mental model updating, made up of conceptualization – experimentation – reflection (Li and Maani, 2011). In the conceptualization phase a general concept of the situation is obtained. Hereby, the outcome of potential decisions is mentally simulated. The current situation is compared to information in the decision maker's mental model.

New information obtained from the environment is integrated to develop a set of decisions. In the experimentation phase, these decisions are tested. The outcome (e.g. feedback) of the experimentation phase is evaluated on in the reflection phase. If the expected outcome is achieved (e.g. positive feedback), initial decisions are kept. If, the outcome is unexpected (e.g. negative feedback) the mental model of the decision maker is updated. Thus, alternatives are sought for, such as new sources of information.

In real-life settings adaptations of mental representations of users are required in many different circumstances. Typical situations which require a user to update his or her mental model are situations leading to errors, due to incomplete or wrong representations. For example, if a user repeatedly fails to install the connection settings for the universities Wi-Fi, he or she needs to adjust his or her mental model, about how to install Wi-Fi on phones. Situations in which changes to the system (due to aspects outside of the person) make the current (in the past correct mental model) inadequate also require adjustment to the user´s mental representation. Examples for the later are a) that due to system-upgrades a new version of an application is launched or b) that past-successful strategies used in decision-making tasks are misleading due to environmental changes.

Nevertheless, the core mechanisms of mental model adaptation should be the same for both situations. This paper demonstrates how mental model build-up and adjustment due to environmental can be addressed using the cognitive architecture ACT-R.

Cognitive architectures allow computationally implementing theories about human cognition in a broad spectrum. The cognitive architecture ACT-R has been applied in many applied domains such as smartphone usage (Prezenski, Bruechner and Russwinkel, 2017) or air-traffic control (Raufaste, 2006) but also in more ground-based research (Halbrügge and Russwinkel, 2016).

ACT-R has symbolic and subsymbolic parts which together produce the modeled behavior. The symbolic parts are chunks, production rules, buffers and modules. The modules resemble the architecture of the human brain. are specified, each of them handles different types of information (chunks). The chunks have slots, they store the smallest pieces of information. The different modules interact through their corresponding buffers. For example, visual information is processed by the visual module and its two buffers. Motor movement is controlled by the manual module and buffer. The declarative module is the long-term memory of ACT-R. Information for this module is retrieved via the retrieval buffer. The imaginal module and buffer are important for learning new information and can be seen as ACT-Rs working memory. Model steering is controlled by the goal module and buffer. The procedural module connects the modules and selects (production-) rules. These production rules are the core part of an ACT-R model- they govern the model behavior. Production-rules can be selected and executed, if buffer states are met. The selected production-rule can then change the states of the modules. An example of a subsymbolic process in ACT-R is the activation level. Thus, if a production requests a chunk and more than one chunk matches this request, this results in the selection of the chunk with the highest activation level. The activation level of a chunk is composed of how often it was used when it was last accessed and how long ago the chunk was created. There are many more subsymbolic processes built into the architecture of ACT-R (e.g. blending, partial matching). Subsymbolic processes are used for modeling implicit learning, e.g. usage of activation mechanism to model information that is well- known can be better retrieved than information that is less well-known.

However, learning (especially in early phases) is also an explicit process (Tenison, Fincham & Anderson, 2016). Thus, the learner is deliberately processing information and deciding what to do next stepwise. This can be modeled by building of new chunks via specific production rules. They can represent the strategies given to a model by the modeler. For an overview and a discussion of implicit and explicit mechanisms in ACT-R in context of intuitive decision-making see Thomson et al (2016).

Explicit mechanisms seem especially important in mental model updating. According to Li and Maani (2011) mental model updating occurs in the reflection phase when negative feedback (unexpected outcome) is observed. Then new sources of information need to be sought for. Such processes require the modeler to use explicit mechanism.

Cognitive models are useful to make precise predictions about theories on human cognition. Models build with cognitive architectures moreover allow precise prediction about behavior influenced by different cognitive processes. They try to capture cognition as a whole. Enough effort, modeling skills and free parameters make it possible to precisely match behavior of participants with models. But for models to be useful. they should be able to predict data in other situations as well. Therefore, modelers should avoid using many specifications to match the data, but attempt to use broader concepts. A successful example for this are models using instance based learning (Gonzalez, 2005). Instance based learning is used to model intuitive decision-making (Thomson et al, 2016). Hereby problem-solving instances are stored in declarative memory and decisions are made by retrieving these instances. The activation mechanism of ACT-R is used to determine which instances are retrieved. However, in early phases of learning and when previously-learned instances become invalid (due to changes in the environment) explicit mechanisms are needed. Such explicit mechanism should be constructed in a general manner and thus be applicable in a variety of tasks.

Aim and Previous Work

The aim of this paper is to show how the same modeling approaches and mechanisms relevant for mental model building and updating can be used in very different applied tasks. Both tasks have in common that they require the participants to explicitly a) learn and b) notice changes and thus to readjust their mental model. Otherwise the tasks are different, thus two ACT-R models are used. Nevertheless, this paper resembles a general modeling approach, since it demonstrates how the core model mechanisms developed in one study (Prezenski and Russwinkel, 2016) are applied to a different study (Prezenski et al. submitted).

The first study investigated a search-and select task with two different smartphone applications. One application allows users to select items to assemble a shopping list and the other to select search-criteria for real-estates. Initial and repeated usage of these applications was investigated. Furthermore, users' adaptation to changes in the applications due to updates influencing the menu-structure (shopping application) and adaptations (real-estate application) was studied.

The second study examined strategy learning in an auditory dynamic decision-making task. In this task, multi-feature sounds were repeatedly presented to the participants. The task was to decide if the presented sound was a target or a non-target. To solve this task a combination of features had to be chosen as targets. The relevance of feature combinations had to be learned from the feedback given in the experiment. In the middle of the task a uniformed switch of targets and non-targets occurred. The task can be seen as an example for dynamic decision-making, because it requires participants to repeatedly make decisions on whether or not a stimulus is a target or a non-target and learn (e.g. improve their decisions) from feedback of the previous decisions. The decisions have to be made under time-constraints. Other feature-combinations become targets at a given point in the experiment due to changes in the environment.

Methods

The methods section of this paper is structured in the following way: First, the core mechanisms for mental model building and mental model updating are described. Second, the results of the first study on smartphone interaction and the implementation of the mechanisms in the first study is summarized. Third, the second study on dynamic decision-making and the transfer and implementation of the mechanisms is explained.

Mechanisms

Mental model building The core part of the mental model (or abstract representation) of a situation, strategy or solution is stored in the *representation chunk* (see figure 1). The slots of this *representation chunk* hold information on the strategy and the expected outcome of applying this strategy. The information on the strategy consists of a representation of the situation and the action.

During mental model building (conceptualisation phase) the *representation chunk* needs to be placed in the imaginal buffer. Only here ACT-R allows chunks to be altered. In the experimentation phase, the expected (or predicted) outcome of this *representation chunk* is tested and then reflected on (reflection phase).

In the reflection phase, mental models can either be revised or strengthened. On the one hand, revision is required, if the outcome is different from what is expected. On the other hand, if the outcome is as expected mechanisms for strengthening the mental model are needed. Here fore, explicit mechanisms are used; namely a slot that notes if a strategy is correct and other slots that keep track (until a threshold) how often a strategy was correct. Other implicit ACT-R strengthening mechanisms are also used, such as that frequently used chunks, are retrieved more often and have a higher activation and this again makes them more likely to be retrieved.

Furthermore, as learning evolves, mental models often become more specific (Gonzalez and Lebiere, 2005). For example, a user experienced in installing Wi-Fi on phones for university networks might have two or more mental models depending on the different types of phones the user installed Wi-Fi for university network in the past. Thus, learners may know that a solution is only applicable for a specific situation (e.g. for one version of an application) such knowledge should also be stored in the *representation chunk*.

Besides a representation of the situation, expected outcome and observed success (core part of the mental model), the building of such a model also requires some form of control over the environmental conditions and the learning history. Such information is stored in the *control chunk* (see Figure 1). This chunk is kept in the goal buffer.

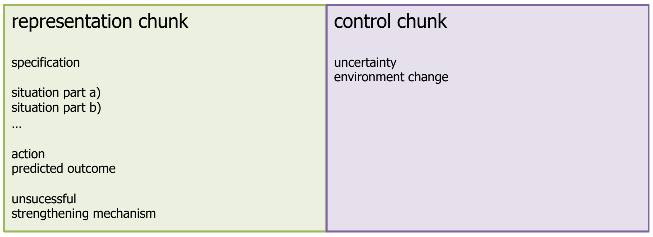


Figure 1: Main chunks and slots required for mental model building and updating

Mental Model Updating In this paper *mental model updating* refers to the modification of an established *representation chunk,* e.g. a strategy that has been successful in the past.

The mechanism, illustrated in Figure 2 works the following way: First, the strategy of the suggested action of the *representation chunk* leads to unexpected outcome. This unexpected outcome is then encoded in a slot of the *control chunk.* This slot represents the uncertainty of the current strategy that something may have changed. The *representation chunk* is nevertheless kept as mental model and tested again. If following the strategy proposed by the *representation chunk* produces unexpected outcome again, this is noted in a slot of the control chunk. This represents that a change has occurred and that a different strategy needs to be built up from now on.

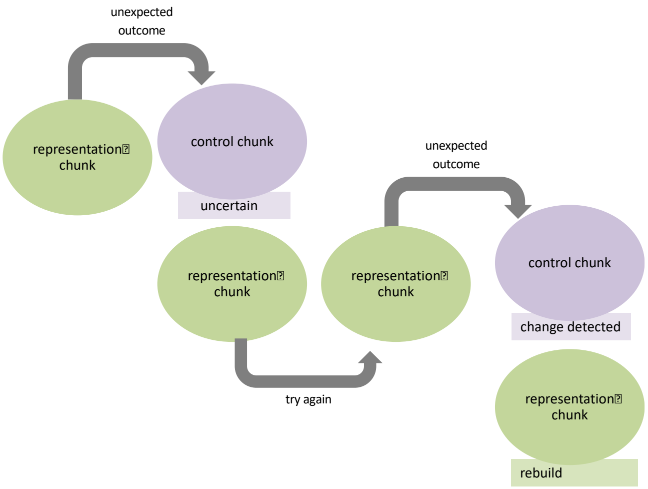


Figure 2: Mental model updating process, governed by specified production rules.

Studies

In the following section the two studies, first the smartphone study and then the decision-making study are presented. Both sections first provide an overview of the tasks and material and then focus on how the core model mechanism from above are implemented respectively.

1) Smartphone Application Study These mechanisms were implemented in a model of users search and select behavior via navigating two smartphone applications. This study has been presented in greater detail elsewhere (Prezenski and Russwinkel, 2016). Thus, only a brief short summary of the applications (material), task, participants, study-design and the implementation of the mental model building and updating mechanism is given.

Material/Applications Two self-designed Android applications (a shopping list application and a real-estate application) each with two different versions were used. The shopping list applications differed in overall menu-depth (three layers vs. four layers). The real-estate application adapted to prior selection, this affected the menu-depth and the positions of some items. These applications were installed on Google Nexus 4.

They are hierarchical-list style applications that support search and select task. Targets and subtargets are spread out over different pages of the applications. See Figure 3 for an impression of the applications.

Task In the shopping-list application participants had to search and select shopping items via navigating through different pages of the application. The participants had to search and select targets (shopping items) via selecting subtargets (e.g. categories, shops) placed on different layers of the application.

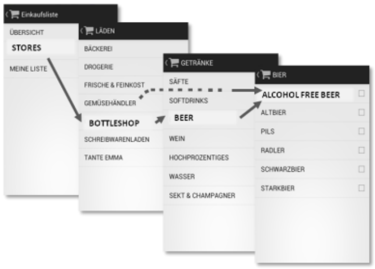


Figure 3: Application layout, reprinted from (Prezenski et al, 2017, p. 170)

In the real-estate application participants had to search and select search criteria for real-estates via selecting different subcategories which were again placed on different layers of the application.

Study-Design The design in the four substudies was similar. In the shopping-list study the participants were required to search for the same nine items for four times. In the first two blocks, they used one version of the application (either three or four layers) in the last two blocks the version "updated" and they had to use the other version. They were not informed about the occurrence of a version switch. In the real-estate study the participants were required to search for either a house or an apartment with six or seven other criteria (e.g. specific size, rent) and after two blocks they had to search for the other one twice (e.g. those who searched for a house twice had to search for an apartment and vice versa). Depending on the pre-selection of house or apartment the position and the menu-depth of other search-criteria could differ (e.g. if house was pre-selected the search-criteria 60qm was positioned higher in the list then if apartment was pre-selected).

The dependent variable is the average target selection time per block. Each block consists of the selection of all items (eight items per block for the shopping list studies and six or seven items for the real-estate studies). Thus, four blocks per study existed.

The four sub studies were conducted with student participants**.** 10 participants took part in the real-estate study where apartment was selected first, and 12 in the one where house was selected first. 17 took part in the shopping list study that used the three-layer version first and 12 in the one that used the four layer version first.

Model implementation The apps were implemented in Lisp and the model was run with ACT-R 7.1. 10 model runs per study were implemented[1]. In the following the modeling principles are summarized. This section focuses on how the mechanism for mental model building and updating are implemented. Other supplemental mechanisms will be briefly introduced in the following section, as well.

Mental model building in smartphone studies The task is to find a target via navigating through different layers of the application. In the beginning of the task, a mental representation of the application is not inherent to the model. Thus, navigation of the application if achieved using *knowledge of the world chunks*. These are made up of associations between different words (e.g. the target-word *alcohol-free beer* is related to the word *bottle shop).* Thus, each item of the application is read and a request for a *knowledge of the world chunk* linking the current processed item and the target, is made. If such a *knowledge of the world chunk* can be found the item is selected, otherwise the next item is processed. The *knowledge of the world chunk* is used to build up a r*epresentation chunk* in the imaginal buffer. If *a representation chunk* is available, it will be used to navigate to the target. This chunk contains the path leading to the target, e.g. which item needs to be selected in order to reach the target. Thus, the items are the *situation* and the target is the *expected outcome*. There is no strengthening mechanism used in this model. But a *specification* mechanism that clarifies when a *representation chunk* is adequate to be used, e.g. use *representation chunk* for a menu-depth of three. However, this is part of the *control chunk* held in the goal buffer. The control chunk of this model also holds information about *uncertainty* of the current strategy (or path chunk) and on *detected changes* (e.g. updates).

Mental model updates in smartphone studies After the second block a change (either a version update or an adaptation due to prior selection) is made to the application. Thus, the established *representation chunk*s will not lead to the expected outcome anymore. So, targets, or subtargets cannot be found with these *representation chunks*. This *uncertainty* is noted in the *control chunk.* Another attempt to find the target using this *representation chunk* is made. If it again does not lead to the target, then a change in the environment is noted. Thus, a strategy change is initiated and the *knowledge-of the world* chunks are used to build a different *representation chunk*. For the next target, a new *representation chunk* is built directly. If the model is required to search for a target with a new version a second time it can retrieve the correct *representation chunk* using the specification (see Figure 4).

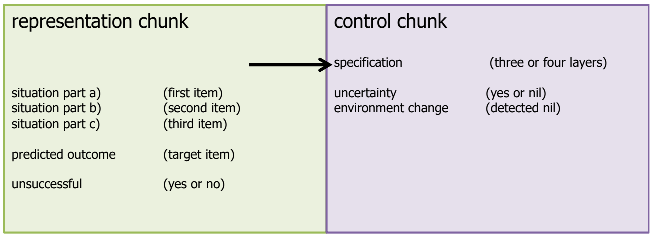

representation chunk
situation part a) (first item)
situation part b) (second item)
situation part c) (third item)
predicted outcome (target item)
unsuccessful (yes or no)
control chunk
specification (three or four layers)
uncertainty (yes or nil)
environment change (detected nil)

Figure 4: Two chunks which are implemented in the Smartphone application study

2) Dynamic Decision-Making Study Mental model building and updating should be the same process even in very different tasks. Thus, it should be modeled in the same way

[1] The data of the model did not show much variance. Thus, additional model runs were not necessary.

as other tasks that require mental model building and updating. Such another task was investigated in the second study. It required the participants to make sense of multi-feature auditory stimuli. The experiment and the model are presented in more detail in Prezenski et al (submitted). In the following section, a short overview of material, task, participants will be given. Followed by a more description of how the mental model building and updating mechanisms were implemented.

The stimuli were 160 different tones. These were made up of a combination of different category features, namely duration (short vs. long), direction of frequency modulation (rising vs. falling) and intensity (quiet vs. loud) and frequency (high vs. low). Tones which included a combination of specific category feature (e.g. loud and falling) were the target stimuli (25%), while the other where the non-targets (75%). Different category-feature combinations were the target for different participants.

In each trial (there were 240 altogether), a tone was presented to the participant and he or she was required to press one of two buttons to classify if the tone was a target or a non-target. After the button-press auditory feedback was presented (“wrong” or “correct”) and then after a pseudo-random time of six, eight or ten seconds the next trial began. After 120 trials, there was a switch of the button allocations, the participants were not informed about this. There were four different randomizations of the experiment; each had different category features as targets.

The dependent variable was the average percentage of correct responses per block. 20 trials were always grouped together as a block. Thus, the experiment consisted of 12 blocks.

55 student participants took part in the experiment.

Model implementation The experiment for the model was implemented in Lisp using the new-other-sound command for the tones and using 16 tones (all possible combinations of the category-feature) pairs as auditory stimuli. The model was written with ACT-R 7.1.

Mental model building The task is to find the correct strategy to classify tones into targets and non-targets. The fact, that a combination of feature-value pairs is the correct solution is unknown to model. Thus, first a single feature-value-pair strategy is used and this is changed to a two feature-value-pair strategy in the course of the experiment.

Two main chunks are part of the model (see Figure 5). The first is a *representation chunk* which holds the current strategy in the imaginal buffer. The second is a *control chunk* in the goal buffer. In the beginning of a trial a tone is heard and a decision has to be made if the tone is a target or not.

The *representation chunk* holds the current strategy the in the imaginal buffer. It contains information about the relevant feature(s) and value(s) (e.g., *the sound is quiet* or the *sound is quiet and its frequency range is high*) and the proposed response (0 or 1). This can be seen as the *situation* and the *predicted action*. Furthermore, the *specification slot* of the *representation chunk* holds information on the degree of complexity of the strategy (e.g. one or two-feature strategy). An evaluation mechanism is part of the representation chunk as well. The evaluation’s result determines if a strategy was unsuccessful and keeps record of how many times a strategy was successful. It marks if the first attempt to use this strategy is successful. Furthermore, the number of successful strategy uses are counted until a certain value is reached. This is meant to reflect the subjective feeling that a strategy was useful often. If a strategy was useful often, then is well-established. The same representation chunk is held in the imaginal buffer as long as feedback is positive. If feedback is negative a different *representation chunk* will be retrieved from memory.

The control chunk holds information on the *uncertainty* about a current strategy and on detected *environmental changes*.

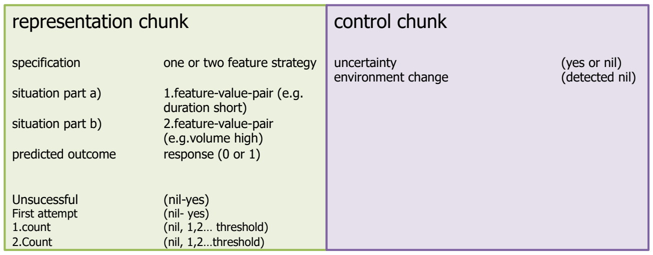


Figure 5: Two chunks which are implemented in the dynamic decision-making study

Mental model updating If an established strategy (in other words representation chunk) causes unexpected negative feedback uncertainty about this current strategy is noted in the control chunk. Nevertheless, this strategy is used a second time. If again unexpected outcome occurs, the strategy will be changed using the mechanism seen in figure 2. In the course of the experiment, this can occur in two different situations. The first situation is, when a one-feature strategy (e.g. volume loudness is 1) is successful often but after repeated unexpected outcome (negative feedback) it is changed into a two-feature strategy. Thereby, the first feature-value pair (volume loudness is 1) is kept as part of the strategy and complemented by another feature-value pair.

The second situation is, after the environment changed when a past establish two-feature strategy repeatedly leads to unexpected outcome. Then different two-feature strategies are sought for.

To sum up, both the smartphone and the decision task implemented mental model building and updating in the same way. Mental model building and updating is modeled using a representation and a control chunk. The followed strategy is held in the representation chunk. This chunk is retrieved from declarative memory and altered using working memory. Information over environmental conditions and the learning history is encoded in the control chunk which is held in the goal buffer. In both models, well-established strategies are not discarded directly in case of unexpected outcome, but tested once more. If they lead to failure again, they are partly revised and rebuilt.

However, the type of behavioral data that the models´ performance was compared to, differed: average item selection time was used for the smartphone studies and percentage of correct responses for the dynamic decision-making experiment.

Results

The results section briefly presents the results of the empirical data together with the modeled data. The results of the smartphone studies are presented in greater detail in Prezenski & Russwinkel, 2016. The results of the decision-making experiment in Prezenski et al (submitted).

Study 1: Smartphone Interaction

In all smartphone sub studies, the model captured the trends found in the empirical data. The trends show a decrease in item selection time from the first to the second block in all four studies. An increase from the second to the third block found in three studies (both real-estate app studies and the shopping-list app, that added an additional layer (shopping 3-4), see Figure 6). In the other shopping-list app study the model also captured the decrease found between the second and the third block. Finally, in all four studies there is a decrease in the mean item-selection time this was again captured by the model.

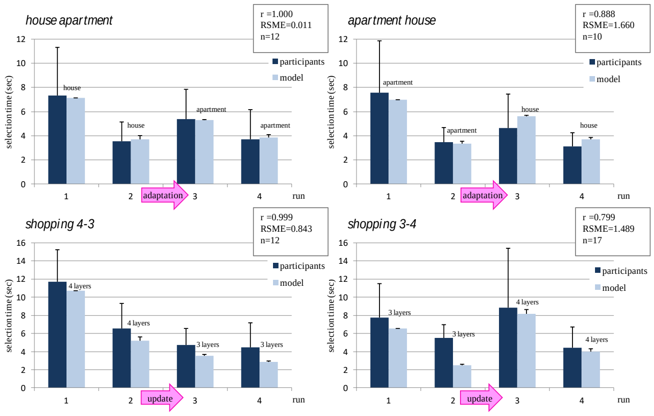


Figure 6: Mean target selection time, reprinted from (Prezenski and Russwinkel, 2016, p. 205)

In the other shopping-list app study the model also captured the decrease found between the second and the third block. Finally, in all four studies there is a decrease in the mean item-selection time this was again captured by the model.

To sum up, the model captured learning and relearning (update detection and new learning). It matched the participant's behavior in mean item selection time very well for all four studies (r² > 0.799).

Study 2: Dynamic Decision-Making

In this study, the empirical data show an increase in the proportion of the correct response from the first to the sixth block (see Figure 7). This is followed by a drop in correct responses in the seventh block, which is pursued by a performance increase until the twelfth block. The model resembles these trends. The overall r² is at 0.672. Nevertheless, the descriptive data indicates that the participants have almost "recovered" from the change in the eighth block, while the model takes longer.

In summary, the model captured the empirical data well; an improvement in performance in the first half of the experiment, the performance drop after the strategy changed and the recovering in performance in the second half of the experiment.

The overall fit of the dynamic decision-making task is not as precise as the fit of the model in the smartphone studies. One explanation hereof is that more measurement points in the decision-making study (12) then in the smartphone study (4) make it less likely to achieve a good fit.

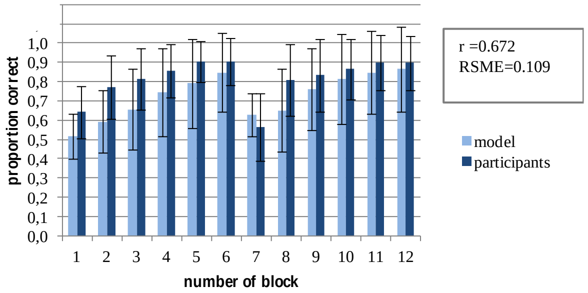


Figure 7: Proportions of correct responses of the model and participants

Another explanation could be that the participants need less long to find an adequate strategy (adequate update of their mental model) after the switch, because they tried the strategy of pressing the other button for the same strategy. Such an explicit strategy was not modeled to keep the model simple and more general.

Discussion

Two very different real-life tasks were modeled with ACT-R using the same explicit mechanisms for mental model building and updating.

The building process of a mental model involves implementing a preliminary version of a mental model and a subsequent testing of this model or strategy. If the strategy performs as expected, it is strengthened, if not it will be updated with a different strategy. Established strategies are not changed immediately in case of unexpected outcome but tested another time before they are changed. Changes to the strategy are gradual; a strategy is not completely discarded; some aspects are kept.

Explicit mechanisms were used, because the changes investigated are registered by the humans. Such distinct noticed changes lead to changes in behavior. Examples for these kinds of changes in real-life settings are software-updates or changes in environmental conditions during outdoor activity (e.g. sudden rain while climbing).

The scope of the presented model mechanisms is not mental model updating during highly automated processes for very skilled users. However, the presented mechanisms can reproduce initial learning, usage and relearning of strategies. Implicit mechanisms are nevertheless part of the models. For example, the previous activation of chunks, as well as if a chunk has been retrieved recently, influence the course of the model.

Modeling the change in strategy and the relatively fast occurring relearning of the participants using solely implicit mechanism with ACT-R is a challenge.

From a cognitive psychological point of view, explicit mechanisms are superior to data driven machine learning approaches, such as deep neural networks because they provide explanations of the underlying mechanisms of participants. Knowledge about explicit strategies of participants is valuable for the design and testing of interactive systems because such knowledge does not only provide summative performance metrics of an interface but also gives hints towards the causes of usability shortcomings and possible solutions.

The examples that have been demonstrated assume specific mental models and provide mechanisms on how such mental models might be updated in human cognition. There is of course no guarantee that such strategies and mental models closely resemble the real strategies, this is not at last grounded on the fact that the human brain does not employ explicit symbol manipulation mechanisms, such as the explained process of building and updating of mental models does. However, the studies that were presented here show that such models provide a reasonable approximation of participant performance.

Potential next steps are investigating the proposed mental model building and switching strategies empirical, with studies targeting these mechanisms.

Summary

This paper demonstrates how mental models updating due to system changes can be modeled using explicit mechanism. This explicit mental model updating mechanism was first implemented in a model of smartphone application usage (Prezenski & Russwinkel, 2016). The mechanism was then applied to a dynamic decision-making task, where participants were presented with different multi-feature auditory stimuli material (Prezenski, Brechmann, Wolff & Russwinkel, submitted). While the model explained data of participants in the smartphone studies very well, the data in the dynamic decision-making task was not explained to such extend.

References

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

Edwards, W. (1962). Dynamic Decision Theory and Probabilistic Information Processing. *Human Factors, 4,* 59–73.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science, 27*(4), 591–635.

Gonzalez, C., & Lebiere, C. (2005). Instance-based cognitive models of decision-making.

Gonzalez, C. (2014). Decision Making: A Cognitive Science Perspective. In S. E. F. Chipman (Ed.), *The Oxford Handbook of Cognitive Science* (Vol. 1). Oxford University Press.

Halbrügge, M., & Russwinkel, N. (2016). The sum of two models: How a composite model explains unexpected user behavior in a dual-task scenario. *D. Reitter & F. E. Ritter (eds.), Proceedings of the 14th international conference on cognitive modeling* (pp. 137–143). University Park, PA: Penn State.

Norman, D. A. (1983). Some observations on mental models. *Mental models, 7*(112), 7-14.

Prezenski, S., Bruechner, D., Russwinkel, N. (2017). Predictive Cognitive Modelling of Applications. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: HUCAPP, (VISIGRAPP 2017)* (pp. 165-171).

Prezenski, S., Brechmann, A., Wolff, S. and Russwinkel, N. (submitted). A Cognitive Modeling Approach to Strategy Formation in Dynamic Decision Making. *Frontiers in Psychology*.

Prezenski, S. and Russwinkel, N. (2016). Towards a general model of repeated app usage. *D. Reitter & F. E. Ritter (Eds.), Proceedings of the 14th International Conference on Cognitive Modeling* (pp. 201–207). University Park, PA: Penn State.

Raufaste, E. (2006). Air traffic control in ACT-R: A computational model of conflict detection between planes. In *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics (HCI - Aero'06),* (pp. 258–259).

Thomson, R., Lebiere, C., Anderson, J. R., & Staszewski, J. (2015). A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture. *Journal of Applied Research in Memory and Cognition, 4*(3), 180-190.

Tenison, C., Fincham, J. M., & Anderson, J. R. (2016). Phases of learning: How skill acquisition impacts cognitive processing. *Cognitive psychology, 87,* 1-28.

Sequential search behavior changes according to distribution shape despite having a rank-based goal

Tsunhin John Wong (jwong@mpib-berlin.mpg.de) [1]

Jonathan D. Nelson (jonathan.d.nelson@gmail.com) [1,2] **Lael J. Schooler (lschoole@syr.edu)** [1,3]

[1] Max Planck Institute for Human Development, Lentzeallee 94, Berlin 12167, Germany

[2] School of Psychology, University of Surrey, Guildford, GU2 &XH, The United Kingdom

[3] Department of Psychology, 509 Huntington Hall, Syracuse University, NY 13244, USA

Abstract

In the area of sequential choice, the 'Secretary Problem' has been a prominent paradigm within the study of optimal stopping for sequential search tasks. Most recent studies of the Secretary Problem present decision makers with the relative ranks of options. A recurring finding is that decision makers tend to end their search earlier than optimal decision strategies (e.g. Helversen, Wilke, Johnson, & Schmid, 2011; Seale & Rapoport, 1997, 2000). By revealing only relative ranks of options or items, issues of learning and incomplete knowledge are avoided; however, this leaves open the question of how sensible human decision makers are when they know more about the distribution of items. Rather than presenting merely ranks to decision makers, we presented numerical values drawn from three distinct distributions in which relatively high value items were scarce, evenly distributed, or abundant. We found that they selected their items earlier than they would if they utilized the optimal selection rule. More importantly, in contrast to the conclusion of Kahan, et al. (1967), we found the selection points of decision makers were sensitive to the underlying distribution. In contrast, the optimal strategy is totally based on quantile ranks regardless of the type of distributions.

Keywords: Sequential choice, Secretary Problem, Heuristics

Introduction

In everyday life, there are many situations in which we need to choose from options presented sequentially. The decision makers may need to choose the best option out of a randomized sequence and may not have the chance to choose an option they have previously rejected. One version of the problem with the goal to find the largest option in the sequence appeared in the mathematical games column by Gardner (1960a, 1960b) in Scientific American. This problem is also known as the *Secretary Problem*.

In the Secretary Problem, there is a reward only if the best item (an interchangeable term for 'option' in our paper) in the sample is chosen. This scenario does not occur too often in daily life, as every option usually has its own value. However, these scenarios do exist, for example, if you are going to strategically sponsor a presidential candidate during their elections for the future benefit of your company, and you probably have to choose only one out of many. At the end, there will only be a single president. In another example, when companies compete to become the contractor of projects, at the end, in most cases, there is only one contractor per project; as an investor or collaborator, you want to choose the one and only winner. Basically, this scenario holds true for winner-takes-all games.

Previous studies with no-information problems

Since the 1960s, many mathematical and behavioral studies have investigated various aspects of the Secretary Problem and its variants that similarly share the goal of choosing a desirable option based on a single attribute of quality. The mathematical studies usually aimed at finding out the optimal choice strategies in the targeted sequential choice problems. Many mathematical analyses assume that the options are drawn from a distribution fully known to the decision maker, also known as *full-information* problems. The behavioral studies typically compare human behavior to an optimal strategy and attempt to explain whether and why human decision makers are optimal or not. Some behavioral studies are based on *partial-information* problems, in which the decision maker knows some (perhaps distribution family and some parameters), but not all, aspects of the distribution from which the options are drawn. The relative rank-based problems, also called *no-information* problems, are where only the relative ranks of options are presented; the ranks of previous options are updated as new options appear.

In recent years, studies of the Secretary Problem have mainly considered no-information problems (e.g. Helversen et al., 2011; Lee, 2006; Lee, O'Connor, & Welsh, 2004; Rapoport & Tversky,1970; Seale & Rapoport, 1997, 2000). No-information problems present only relative ranks of items and make the Secretary Problem more tractable because complexities such as how decision makers learn the underlying distribution and their individual differences in learning and understanding are altogether avoided. However, in daily life, much of the time decision makers judge an option with some degree of knowledge or prior belief about the distribution it is coming from. Therefore, partial-information problems are closer to most of the sequential choice scenarios in daily life. In this study, we tried to approach this classic sequential problem by presenting values to the decision makers instead of relative ranks. By using real values, it becomes possible to manipulate the distribution environment and investigate the choice behavior of decision makers in different underlying distribution shapes.

How distribution shapes affect strategies on the full-information Secretary Problem

Some studies have investigated the effect of distribution shape in sequential problems. Rapoport and Tversky (1966, 1970) trained their subjects for a few weeks on distributions of item quality and concluded that their subjects performed optimally on two-thirds of the tasks. However, Rapoport and Tversky only used a uniform distribution for item quality. Distribution shapes, for instance positive skew, negative skew, or uniform, have been manipulated during experimental investigations of the Secretary Problem by only a few studies (e.g. Guan & Lee, 2014; Kahan, Rapoport, & Jones, 1967). One early study was conducted by Kahan et al. (1967); they trained their participants for over 3 weeks in the experiments. They concluded that there was not sufficient evidence to say their participants used different stopping points in environments with the different underlying distribution shapes. Guan & Lee (2014) tested their participants in a slightly different setting: without the benefit of the extensive training that Kahan et al's participants underwent, Guan & Lee's participants worked on randomized sequences of five items drawn from one of two distributions derived from the Beta distribution. They concluded that their participants could have used multiple-thresholds with decreasing values towards the end of the sequence, and that these thresholds are not affected by the value of preceding items.

Gilbert and Mosteller (1966, Section 3) explain how to derive the optimal strategy for the Secretary Problem under different distributions, when the goal is to find the highest item in the sequence and the distribution shape is fully known. The optimal strategy is in the form of a multi-threshold rule, a sequence of nonincreasing thresholds, usually monotonically decreasing. The largest and first threshold is for deciding whether to accept the first item; provided that the first item was not accepted, the second threshold is used to decide whether to accept the second item; and so on. The rules of the game require that the last item must be accepted if no previous item has been accepted. In the case of full knowledge problem of the Secretary Problem, the optimal strategy is based on a distribution-related percentile-based multithreshold rule (See Gilbert & Mosteller, Section 3). The percentile-based thresholds vary with the number of items in the sequence, however, the strategy is basically the same across distributions. What will be different is the exact numerical threshold values for different distributions. For a number of continuous and discrete distribution families, we derived the corresponding multi-threshold optimal decision strategies, which numerically vary according to the underlying distribution shapes.

Consider the behavior of the optimal strategy given three Beta distributions: a positively-skewed distribution (β(1,3.7); skew = 1.00), a negatively-skewed distribution (β(3.7,1); skew = -1.00), and the uniform distribution (β(1,1); skew = 0). If we consider the underlying distribution about the quality of items to be an 'environment', we can characterize a positively-skewed distribution as a scarce environment with a lot of low quality items and only a few high quality items; and correspondingly a negatively-skewed distribution can be characterized as an abundant environment with mostly high quality items. These distributions are normalized (Z-scored) so that irrespective of the underlying distribution, a random response strategy has expected payoff of zero, and standard deviation of 1. The optimal thresholds are highest for the positively-skewed distribution, and then for the uniform distribution, while the negatively-skewed distribution has the lowest thresholds.. We wanted to see (1) how well the optimal models account for the decision makers' performance and (2) to what extent the decision makers are sensitive to the different underlying distributions.

When the optimal strategy for the full-information Secretary Problem is used (Gilbert & Mosteller, 1966), although the numeric values of the multithreshold rule vary according to the underlying distribution, the probability of stopping a search follows a fixed set of probabilities, and does not vary with the underlying distribution. However, it is unknown whether search behavior will be affected by the underlying distributions in the Secretary Problem, even when decision makers are familiar with the distributions. If people are insensitive to the underlying distribution shapes, their search lengths (how long they reach the randomized sequence before selecting an item) and success rates will remain the same in different distributions, as predicted by the optimal models.

We used a between-subjects manipulation of distribution shape in an experiment with real-money payoffs, and a scarce distribution, a neutral distribution and an abundant distribution, to test these predictions with human subjects.

Method

We used an Internet card game with real money payoffs to implement the Secretary Problem. In a between subjects design, each subject was randomly assigned to the positively-skewed, negatively-skewed, or uniform distribution. There was a training phase of at least 10 rounds of the card game, and a test phase with exactly 10 rounds of the same card game. The value of each card that was shown to the subjects was formed by taking the underlying Z-scored distribution value, adding 4, and then multiplying it by 1200, such that all cards would have positive point values, between approximately 2500 and 9500.

Stimulus

In each round, subjects were asked to obtain the card with the largest number out of a 25-card sequence. Among all 10 test-phase rounds of the experiment, a successful round required selection of the largest card, and this added a fixed bonus to the participation payment. The conversion to real money was 2000 points to $1 US; this was disclosed at the outset. Subjects were encouraged to achieve as high of a score as they could.

Subjects

We recruited subjects through the Amazon M-Turk platform, and allowed only US subjects with consistently good reviews under the Amazon M-Turk monitoring system ('M-Turk Master workers') to participate. Bonus payments proportional to subjects' performance were rewarded through the M-Turk system, on top of a one US dollar participation fee. Informed consent was obtained from all subjects.

Training After the instructions and a practice example, there was a training phase. A subject had to correctly select the largest card for at least 4 rounds in their 10 rounds of training; otherwise, the phase would begin all over again with the count of their successful rounds reset. No bonus payment was offered for the training phase. If the training phase was successfully completed, then the subject had to complete a test phase consisting of 10 rounds with a real-money (U.S. dollars) bonus payment of 5 cents for each time successfully selecting the largest card in the sequence of the 10 test rounds.

Result

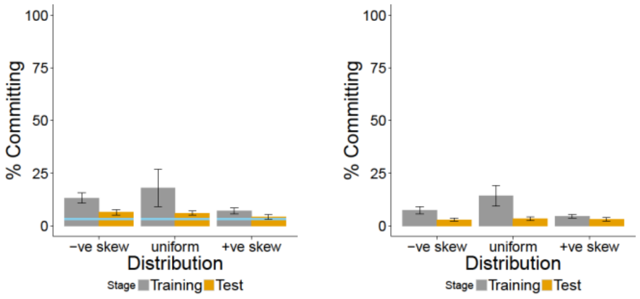
208 recruited subjects were randomly assigned among the conditions: 68 subjects to the positive skew distribution condition, 70 to the uniform condition, and 70 to the negative skew condition. 155 subjects completed the experiment with 54 in the positive skew condition, 52 in the uniform condition, and 49 in the negative skew condition.

Conscientiousness of subjects

In the full-information Secretary Problem, the optimal strategy sometimes (about 3.4% of the time) selects the very first card. However, selecting a card that is smaller than any previously viewed card guarantees failure. For each distribution and phase (Figure 1), we checked the rate of these behaviors in our subjects. These behaviors were rare, consistent with conscientious behavior of the subjects.

Average success rate

Among the number of rounds of Secretary Problem attempted, the percentage of rounds in which the best item in the sequence was selected – success rates (Figure 2, thick horizontal lines) of these three distributions are similar. The success rate of random responding (4% in all conditions, denoted by the dashed grey lines) lies well beyond the interquartile range (IQR) for all conditions and beyond the whiskers (first quantile – 1.5 x IQR, and third quantile + 1.5 x IQR) for most of the rounds. When using the Wilcoxon one-sample tests to the success rates of training rounds and test rounds to the 4% random success rate, p-values are all < 0.001 for the 3 distributions in the training rounds, and for the test rounds; therefore the possibility that subjects were responding randomly can be ruled out. Subjects from the positively-skewed distribution condition had the highest success rates, closely followed by the subjects from the uniform distribution, and then from the negatively skewed distribution (Figure 2).



1a) % choosing 1^{st} card 1b) % choosing non-max

Figure 1a & b: Left – Percentage of rounds choosing the first card. Error bars cover the range one standard error above and below the mean. The full-knowledge optimal strategies also select the first item out of 25 with a 3.24% chance for all the distributions (see the blue line); Right – Percentage of rounds not choosing the largest card. Error bars cover the range one standard error above and below the mean. The graph indicates how often subjects selected cards that were not the largest card seen in the sequence so far.

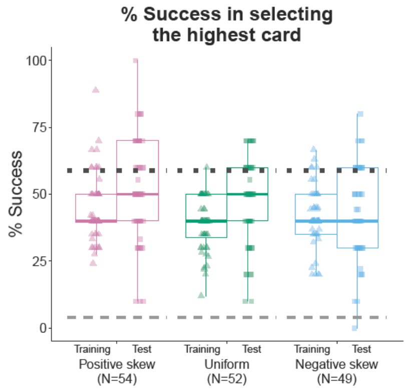


Figure 2: Percentage of success in selecting the highest card. The lower and upper hinges of the boxplots correspond to the 25th and 75th percentile of the data. The whiskers extend to 1.5 times beyond the bounds of the interquartile range. Dotted horizontal dotted lines denote the average score of the optimal strategy for each distribution. Dashed grey lines denote the success rate of random responding (4%, in all conditions). Scores have a slight ordering from positive-skew to uniform and then to negative-skew distribution.

Values of chosen cards

Values of cards our subjects selected at every position under each distribution were plotted, and compared to the theoretical threshold values that the optimal strategies predict: conditional that the card values are higher than the threshold for a given position and that no card previously shown is higher than the selected card (Figure 3). The ordering of card values obtained was highest for the positively-skewed distribution, second highest for the uniform distribution, and lowest for the negatively-skewed distribution, resembling the ordering in the optimal strategies. Subjects initially would select cards with lower values that showed up in the first few positions during their training (Figure 3, top); however, they might have already adjusted and increased their thresholds when they reached the test rounds. In the test rounds, the accepted values are closer to the optimal strategies in the earlier positions, and the accepted values, unlike the pattern exhibited by the optimal strategies, have a stretched plateau shape, generally extending from 1st position until to almost the 23rd or 24th position out of 25. Data seem to get closer to the patterns of chosen card values of the optimal multithreshold rules from training rounds to test rounds, nevertheless, their behavior is not optimal. Given the similarity between the patterns of the values of chosen cards and the conditional expected values of chosen cards using optimal strategies with non-increasing multithreshold rules, the ordering of underlying thresholds utilized by our subjects among the conditions is probably consistent with the ordering of the non-increasing multithreshold rules.

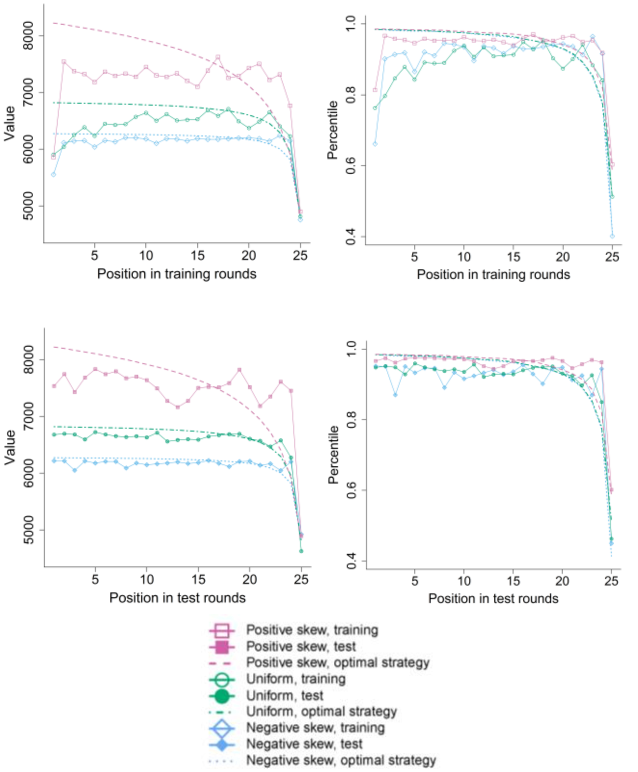


Figure 3: Values and percentiles of chosen cards at positions where they are chosen. We grouped the results by their decision points in the sequence and found out the mean values for each position on condition that the subjects stopped there. The dashed lines are the values of card at a certain position when the decision maker makes use of the optimal (full-knowledge) strategy from the very beginning of the sequential choice problem, on condition that the stop happens to be at that position

Efficiency

The optimal strategies suggest that in the positively-skewed distribution, the thresholds used should be the highest, followed by the uniform distribution, and then by the negatively-skewed distribution. This trend was demonstrated in the percentile ranks of cards selected by our participants. The optimal strategies suggested that in order to select the best item with the optimal thresholds, search length would be the same for all three distributions regardless of their characteristics and skewness; this trend in search length was not exhibited by our subjects (Figure 4). One-sample Wilcoxon tests showed that the search lengths of all distributions from both training rounds and test rounds are shorter than that of the optimal strategies (all p-values < 0.001). There were also differences in search length as a function of distribution shape (Kruskal-Wallis rank sum test, p = 0.004); it appears that they searched longer in the positively-skewed distribution than in the other two distributions. In addition to the above observations, the subjects also selected cards with higher values in the test rounds than in the training rounds (Wilcoxon Signed-Rank Sum Tests, W=268, p <0.001, d= - 0.58 for the negatively skewed distribution; W=418, p = 0.017, d= - 0.25 for the uniform distribution; and W=373, p <0.001, d= - 0.44 for positively skewed distribution), despite having similar search length in training and test rounds. It seems that our subjects have learnt about the distributions and had their performance improved towards the performance of the optimal strategies.

Search length

Besides looking at the means of search length, we also looked at the cumulative distribution of how frequently a selection had already been made by a particular position in the sequence (Figure 5). Optimal strategies for the full-information Secretary Problem lead to the same pattern of search length (Figure 5, solid black line without marker) irrespective of the underlying distribution (Gilbert & Mosteller, 1966). For our subjects, when combined as a group, search length was closer to the optimal strategies in the positive skew condition, and not as close in the negatively skewed distribution or in the uniform distribution. The search lengths of the subjects follow a pattern of concave downward until the 24th item in the sequence; shapes of the cumulative distribution curves suggest a tendency for the subjects to select an item sooner than the optimal strategies: the curves of all conditions are concave downward whereas the curve of the optimal strategy is slightly concave upward – in other words, they all have the tendency to stop too soon in both training rounds and test rounds. Although the outcome of optimal strategies does not predict that, there are some obvious differences between the curves of the positively skewed distribution and that from the other two distributions in both training rounds and test rounds. For the comparison of these curves, we conducted the Kolmogorov-Smirnov tests to examine their differences: in particular, the difference of search lengths among the three underlying distributions; however, none of the Kolmogorov-Smirnov tests was significant, including the comparison between the curves from subjects' data to the curves from the optimal strategies. We suspect the Kolmogorov-Smirnov test for distributions may not be for right test for our purpose. The sudden increases from item 24 to item 25 reflect the rule of the task that participants have to accept the last item as long as it is reached. For the cumulative selection probability, learning was reflected only a bit more obviously in the positive skew distribution as a shift closer to the optimal strategies from training to test (Figure 5, top to bottom).

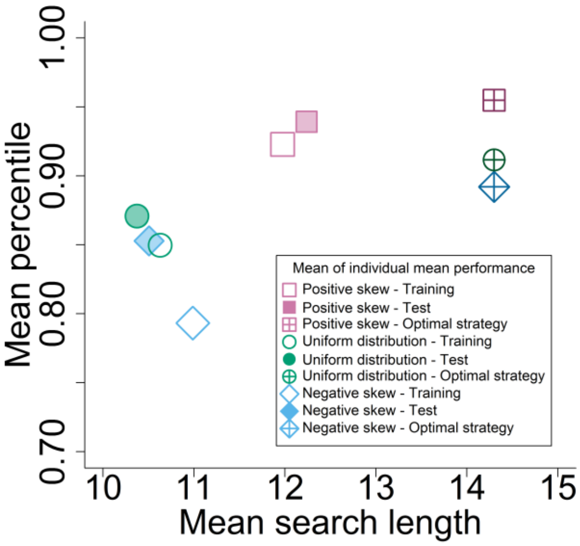


Figure 4: Values of chosen cards vs search length. Means of individual scores are plotted as a function of means of individual search length (number of items searched up until an item is accepted). The aggregate means for all subjects in each distribution are plotted with different markers (unfilled for training, filled for test). The optimal strategy is plotted with markers with internal plus signs. The upward switch of locations of these markers from when subjects were during training rounds to when they were in the test rounds shows how performance improved from training to test for each distribution, despite similar search length.

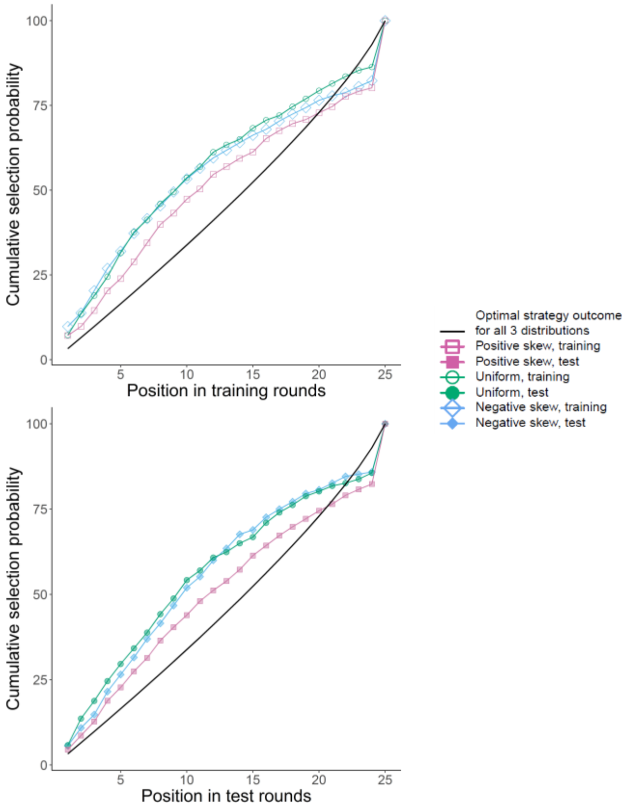


Figure 5: The patterns of search length are similar between training rounds and test rounds, having similar concave downward patterns. All patterns showed they stopped earlier than the optimal strategy does. Subjects in the positively skewed distribution seemed to stop early with a lesser tendency. The jumps towards the last position were due to the rule of the Secretary Problem that the last item when reached must be selected.

Discussion

There have been experiments on the Secretary Problem (Gardner, 1960a), in which a payoff is obtained only if the largest item is selected (e.g. Bearden, Rapoport, & Murphy, 2006; Lee et al., 2004; Seale & Rapoport, 1997, 2000), as well as related tasks in which the goals are to obtain an item in the top 10% or 25% (Todd & Miller, 1999). Although many real-life situations involve real value options, Secretary Problems with real values have received little attention in psychological experiments. We conducted a study of the Secretary Problem with information, to look at how human sequential choice behavior may vary as a function of distribution shape. Our subjects were sensitive to the underlying distribution shape (as shown in Figure 4 and Figure 5).

The age-old claim continues: our subjects stopped their searches early, i.e. earlier than the optimal strategies do (Figure 4 and Figure 5), and it was also not likely for them to keep searching until the last few items of the sequence. There are a few plausible explanations. The tendency to stop earlier than predicted could be obtained by an enhanced model that incorporates a small intrinsic sampling cost (Seale & Rapoport,1997). Another possibility is that in addition to search cost, subjects were affected by a goal that is more natural for them, for example a goal that involves satisficing (Todd & Miller, 1999), that is, an implicit goal to be happy about obtaining a high value item, instead of waiting out for the highest item to show up.

After training, the performance of our subjects got closer to the optimal strategy, for all of the distributions tested (Figures 3 and 4). Moreover, our subjects searched for cards slightly longer in positively-skewed environment than in the other two environments. This behavior contrasts with the optimal strategies, which have exactly the same search length pattern, irrespective of the underlying distribution of item quality. It is not totally clear why decision makers exhibit these behavior patterns. It is possible that our subjects exhibited these behaviors because of imperfect knowledge of the underlying distributions, and there was noise in sampling when they sampled to set and adjust their thresholds. A further direction to explore is plausible simple strategies that decision makers may employ (Gigerenzer, Hertwig, & Pachur, 2011) to select cards in a manner with monotonic decreasing thresholds. Such a strategy in theory could potentially come quite close to the optimal strategies in terms of success rate and efficiency.

Acknowledgments

Part of this work was presented at the 2013 Mathematical Psychology Conference in Potsdam, Germany. We thank Gerd Gigerenzer, the ABC Research Group, and Peter Todd for very helpful feedback and ideas. This research was supported by German Academic Exchange Service (DAAD) grant 91540389 to TJW, and by Deutsche Forschungsgemeinschaft (DFG) grant 1713/2 to JDN as part of the priority program “New Frameworks of Rationality” (SPP 1516).

References

Bearden, J. N., Rapoport, A., & Murphy, R. O. (2006). Sequential Observation and Selection with Rank-Dependent Payoffs: *An Experimental Study. Management Science, 52*(9), 1437–1449.

Gardner, M. (1960a). Mathematical Games: A fifth collection of brain teasers. *Scientific American*, 150–155.

Gardner, M. (1960b). Mathematical Games: The games and puzzles of Lewis Carroll, and the answers to February’s problems. *Scientific American*, 172–182.

Gigerenzer, G., Hertwig, R., & Pachur, T. (2011). *Heuristics: The Foundations of Adaptive Behavior*.

Gilbert, J. P., & Mosteller, F. (1966). Recognizing the Maximum of a Sequence. *Journal of the American Statistical Association*, *61*(313), 35–73.

Guan, M., & Lee, M. D. (2014). Threshold Models of Human Decision Making on Optimal Stopping Problems in Different Environments. In P. Bello, M. Guarini, M. McShane, & B. Scassellati (Eds.), *Proceedings of the 36th annual conference of the cognitive science society* (pp. 553–558). Austin, TX: Cognitive Science Society.

Helversen, B. V., Wilke, A., Johnson, T., & Schmid, G. (2011). Performance Benefits of Depression: Sequential Decision Making in a Healthy Sample and a Clinically Depressed Sample. *Journal of Abnormal Psychology, 120*(4), 962–968.

Kahan, J. P., Rapoport, A., & Jones, L. V. (1967). Decision making in a sequential search task. *Perception & Psychophysics, 2*(8), 374–376.

Lee, M. D. (2006). A hierarchical bayesian model of human decision-making on an optimal stopping problem. *Cognitive science*, *30*(3), 1–26.

Lee, M. D., O’Connor, T. A., & Welsh, M. B. (2004). Decision-Making on the Full Information Secretary Problem. In K. Forbus, D. Gentner, & T. Regier (Eds.), *Proceedings of the 26th annual conference of the cognitive science society* (pp. 819–824). Mahwah, NJ: Erlbaum.

Rapoport, A., & Tversky, A. (1966). Cost and accessibility of offers as determinants of optional stopping. *Psychonomic Science, 4*(1), 145–146.

Rapoport, A., & Tversky, A. (1970). Choice behavior in an optional stopping task. *Organizational Behavior and Human Performance, 5*(2), 105–120.

Seale, D. A., & Rapoport, A. (1997). Sequential Decision Making with Relative Ranks: An Experimental Investigation of the "Secretary Problem". *Organizational Behavior and Human Decision Processes, 69*(3), 221–236.

Seale, D. A., & Rapoport, A. (2000). Optimal Stopping Behavior with Relative Ranks: The Secretary Problem with Unknown Population Size. *Journal of Behavioral Decision Making*, *13*, 391–411.

Todd, P. M., & Miller, G. F. (1999). From Pride and Prejudice to Persuasion: Satisficing in Mate Search. In G. Gigerenzer & P. M. Todd (Eds.),*Simple heuristics that make us smart* (pp. 287–308). Oxford University Press.

Wong, T. J., Nelson, J. D., & Schooler, L. J. (2016). Wait for the best, or settle for less? Search adapts to distribution shape in a value-based Secretary Problem. Manuscript submitted for publication.

Decisions from Experience: Modeling Choices due to Variation in Search Strategies

Neha Sharma (neha724@gmail.com) & Varun Dutt (varun@iitmandi.ac.in)
Applied Cognitive Sciences Laboratory, Indian Institute of Technology Mandi, Kamand, Himachal Pradesh, India – 175005

Abstract

Decisions from Experience (DFE) research involves a paradigm (called, sampling paradigm), where decision-makers search for information before making a final consequential choice. Although DFE research involving the sampling paradigm has focused on accounting for information search and final choices using computational cognitive models. However, little attention has been paid to how computational models could account for final choices of participants with different information-search strategies. In this paper, we perform an individual-differences analysis and test the ability of computational models to explain final choices of participants with different search strategies. More specifically, we take an Instance-Based Learning (IBL) model, which relies on recency processes, and we calibrate this model to final choices of participants exhibiting more-switching (piecewise strategy) or less-switching (comprehensive strategy) between options in different problems. Our results indicate more reliance on recency of information among participants exhibiting piecewise strategy compared to comprehensive strategy. Overall, the IBL model calibrated to individual participants using a single set of parameters could account for both piecewise and comprehensive strategies. We highlight the implications of our results for DFE research involving information search before consequential decisions.

Keywords: information search; experience; search strategy; computational cognitive models; Instance-Based Learning Theory; multi-arm bandit problems.

Introduction

In words of famous philosopher Plato, a good decision is based on knowledge and not numbers (Stutman & Kevin, 2015). Knowledge can be obtained by searching the environment for information before making consequential decisions. For example, investment decisions are likely to be affected by an investor’s previous knowledge of a company’s stocks (Subramanyam, 2016). An investor could invest in a wide range of companies in the stock market. To ensure a good decision, one must gather information about various returns offered by different stocks before making a consequential choice for a company’s stocks. While gathering information, some people may explore the prices of a company’s stock repeatedly before switching to a different company’s stock (comprehensive strategy). However, some people may explore prices of a company’s stock once and then switch to exploring the stock prices of a different company (piecewise strategy). In both cases, it is important to investigate how influential computational cognitive models account for consequential choices among both kinds of search strategies. This investigation is the main goal of this paper.

The act of making choices based on information search is a common exercise involving people in different facets of their daily life (choosing smartphones, choosing TV channels etc.). In fact, information search before a choice constitutes an integral part of Decisions from Experience (DFE) research, where the focus is on explaining human maximizing decisions based upon one’s experience with sampled information (Hertwig & Erev, 2009). To study people’s search and choice behaviors in the laboratory, DFE research has proposed a “sampling paradigm” (Hertwig & Erev, 2009).

In the sampling paradigm, people are presented with two or more options to choose between. These options are represented as blank buttons on a computer screen. People are first asked to sample as many outcomes as they wish and in any order they desire from different button options (information search). This sampling of information among different options is costless. Once people are satisfied with their sampling of options, they decide from which option to make a single final (consequential) choice for real.

Hills and Hertwig (2010) have analyzed the search strategies of people asked to make choices in the sampling paradigm. Hills and Hertwig (2010) report two search strategies prevalent among participants: comprehensive and piecewise. In the comprehensive strategy, people search one option repeatedly before switching to the other option. In contrast, in the piecewise strategy, people search for one option once and then switch to the other option. They sample the other option once and again switch back to the first option, searching for information in a zigzag manner.

Computational cognitive models of human choice behavior have thus far predicted choices at an aggregate level in the sampling paradigm, i.e., when people’s final choices are averaged over several participants (Busemeyer & Wang, 2000; Gonzalez & Dutt, 2012; Lejarraga, Dutt, & Gonzalez, 2012). For example, the Instance-Based Learning (IBL) model is a popular DFE algorithm for explaining aggregate choices (Erev et al., 2010; Gonzalez & Dutt, 2011; Lejarraga, Dutt, & Gonzalez, 2012; Hertwig, 2012). The IBL model (Gonzalez & Dutt, 2011) consists of experiences (called instances) stored in memory. Each instance’s activation is used to calculate the blended values for each option, thereby helping the model make a final choice. The IBL model relies on ACT-R framework for its functioning (Anderson & Lebiere, 1998).

Prior DFE research has shown that, at the aggregate level, the IBL model exhibits superior performance compared to other computational models in the sampling paradigm (Erev et al., 2010; Gonzalez & Dutt, 2011).

Although computational cognitive models have been evaluated at the aggregate level; yet, less attention has been paid to the evaluation of models in their ability to account for individual differences, especially in terms of search strategies. Given that people exhibit two specific search strategies (Hills and Hertwig, 2010), comprehensive and piecewise, it would be interesting to see how computational cognitive models with a set of parameters account for consequential choices for participants exhibiting these strategies.

In this paper, our main goal is to evaluate how computational cognitive models, which explain choice behavior at the aggregate level (e.g., IBL model), perform in capturing consequential decisions of participants exhibiting different search strategies with single set of parameters. For this purpose, we use risky problems involving two options and outcomes with different probabilities (rare events and common events). We calibrate an IBL model, which was evaluated in prior research at the aggregate level, to preferences of participants showing different search strategies. In what follows, we detail the problems used and the working of the IBL model. Then, we discuss the methodology of calibrating the IBL model to consequential decisions in different problems. Next, we present the results of model evaluation and the role of recency and frequency mechanisms in accounting for consequential decisions involving different search strategies. We close the paper by discussing the implications of our results for DFE research in the sampling paradigm.

Problem Dataset

Eighty students at Indian Institute of Technology Mandi, India, participated in a study where the objective was to evaluate participant preferences for options after information search. The study involved the sampling paradigm, where participants searched for information and then decided an option they preferred across two between-subjects problem conditions: Rare-Event (RE; N = 40) and Common-Event (CE; N = 40). In the CE problem, a variable option had a high probability (0.8) value associated with a high (H) outcome (1.18 return on the allocated amount); whereas, in the RE problem, the variable option had a low probability (0.1) associated with the H outcome (3.28 return on the allocated amount). Across both problems (CE and RE), the low (L) outcome (0.88) in the variable option always occurred with a complementary chance. An alternative with a fixed return on investment (1.1 return on the invested amount with certainty) was present in both RE and CE conditions as second option. Thus, in each problem, participants were presented with two options: an option with a fixed return on allocation (non-maximizing option); and, an option with a variable return on allocation (maximizing option). The maximization was defined based upon the expected value of options in problems. The nature of outcomes and probabilities in different CE and RE problems were like those described in Hertwig et al. (2004).

In each problem, participants were first asked to sample options (presented as blank buttons; sampling phase). During the sampling phase, every time an option was chosen in a problem, participants could see an outcome based upon the associated probability in the option. Sampling of options was costless in the sampling phase and participants were free to sample options in any order and as many times as they desired. At any time during the sampling phase, participants could click the “Make a Final Decision” button. Clicking this button terminated the sampling phase and moved participants to the final-decision phase. In the final-decision phase, participants were asked to make a final choice for one of the options for real.

To understand the effect of different sampling strategies, we calculated the switch ratio, which was defined as the total number of switches made by a participant between options divided by the total number of switches possible (= number of samples – 1). Like done by Hills and Hertwig (2010), we calculated the median value of switch ratio by pooling participants across both CE and RE problems. Participants possessing switch-ratios less than median were classified as following comprehensive search strategy (called LM) and participants possessing switch-ratios greater than or equal to median were classified as following piecewise strategy (called GM). By pooling the CE and RE problems, there were N = 40 participants in the LM group and N = 40 participants in the GM group.

Human Results

Figure 1 shows proportion of final choices by human participants in the GM and LM condition. As seen in the Figure, the pattern of preferences across problems in the LM and GM conditions was similar in human data: higher allocation to the fixed option compared to the variable option. Next, we consider whether an IBL model can account for these effects via its cognitive mechanisms (model results will be described in a future section).

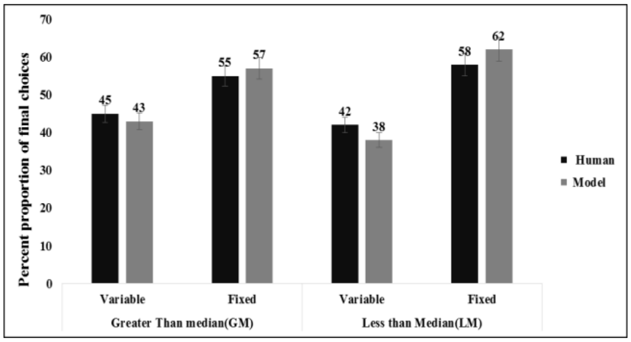


Figure 1: Percentage Proportion of final choices in each option by human and model for both GM and LM condition.

The Model

In this section, we detail the working of a model based upon Instance-Based Learning Theory (IBL model; Gonzalez & Dutt, 2011; 2012), which was calibrated to LM and GM search strategy groups separately.

Instance-Based Learning (IBL) Model

The IBL model (Dutt & Gonzalez, 2012; Gonzalez & Dutt, 2011; 2012; Lejarraga, Dutt, & Gonzalez, 2012) is based upon the ACT-R cognitive framework (Anderson & Lebiere, 1998). In this model, every occurrence of an outcome of an option is stored in the form of an instance in memory. An instance is made up of the following structure: SDU, here S is the current situation (two blank option buttons on a computer screen), D is the decision made in the current situation (choice for one of the option buttons), and U is the goodness (utility) of the decision made (the outcome obtained upon making a choice for an option). When a decision choice needs to be made, instances belonging to each option are retrieved from memory and blended together. Blended value of an option is a function of activation of instances corresponding to outcomes observed on the option. Activation of an instance is a function of the frequency and recency of observed outcomes that occur on choosing options during sampling. The blended value of option *j* at any trial *t* is defined as (Lebiere, 1999):

$$V_{j,t} = \sum_{i=1}^{n} p_{i,j,t} x_{i,j,t} \qquad (1)$$

where $x_{i,\ j,\ t}$ is the value of the U (outcome) part of an instance *i* on option *j* at trial *t*. The $p_{i,j,t}$ is the probability of retrieval of instance *i* on option *j* from memory at trial *t*. Because $x_{i,j,t}$ is value of the U part of an instance *i* on option *j* at trial *t*, the number of terms in the summation changes when new outcomes are observed within an option *j* (and new instances corresponding to observed outcomes are created in memory). Thus, $n = 1$ if *j* is an option with one possible outcome. If *j* is an option with two possible outcomes, then $n = 1$ when one of the outcomes has been observed on an option (i.e., one instance is created in memory) and $n = 2$ when both outcomes have been observed (i.e., two instances are created in memory).

At any trial *t*, the probability of retrieval of an instance *i* on option *j* at trial *t* is a function of the activation of that instance relative to the activation of all instances (1, 2, … *n*) created within the option *j*, given by

$$p_{i,j,t} = \frac{e^{(A_{i,j,t})/\tau}}{\sum_{i=1}^{n} e^{(A_{i,j,t})/\tau}} \qquad (2)$$

Where τ, is random noise defined as $\sigma . \sqrt{2}$ and σ is a free noise parameter. Noise captures the imprecision of recalling past experiences from memory. The activation of an instance *i* corresponding to an observed outcome on an option *j* in each trial *t* is a function of the frequency of the outcome's past occurrences and the recency of the outcome's past occurrences (as done in ACT-R). At each trial *t*, activation $A_{i,j,t}$ of an instance *i* on option *j* is

$$A_{i,j,t} = \sigma * \ln\left(\frac{1-\gamma_{i,j,t}}{\gamma_{i,j,t}}\right) + \ln \sum_{t_p \varepsilon \{1..,t-1\}} \left(t - t_p\right)^{-d} \qquad (3)$$

where *d* is a free decay parameter; $\gamma_{i,j,t}$ is a random draw from a uniform distribution bounded between 0 and 1 for instance *i* on option *j* in trial *t*; and t_p is each of the previous trials in which the outcome corresponding to instance *i* was observed in the binary-choice task. The IBL model has two free parameters that need to be calibrated: *d* and σ. The *d* parameter controls the reliance on recent or distant sampled information. Thus, when *d* is large (> 1.0), then the model gives more weight to recently observed outcomes in computing instance activations compared to when *d* is small (< 1.0). The σ parameter helps to account for the sample-to-sample variability in an instance's activation. In the IBL model, we feed the sampling of individual human participants to generate instance activations and blended values. Every time a choice is made and outcome is observed, the instance associated with it is activated and thereafter blended values are computed for options faced by an individual participant.

In one version of the IBL model, we use parameters suggested by Lejarraga, Dutt, and Gonzalez (2012) to test the model's ability in capturing final choices for different search strategy groups, LM and GM. In a second version of the model, we found values for the *d* and σ parameters by calibrating these parameters to final choices from human participants separately in the two strategy groups. For this calibration, we determine the model's likelihood for making the same choice as made by each human participant given a set of model parameters.

For each model participant, the model applied the following softmax function across both options in a problem (Bishop, 2006; Sutton & Barto, 1998):

$$Prob(Option\ X) = \frac{e^{S_{Mean}X}}{e^{S_{Mean}X} + e^{S_{Mean}Y}} \qquad (4)$$

Where, $S_{Mean}X$ and $S_{Mean}Y$ are the blended values calculated for the two options and *Prob (Option X)* is the probability of choosing Option X given a set of model parameters (also, called the "likelihood"). If *Option X* was chosen by a human participant in a problem, then the *Prob (Option X)* is used to calculate the likelihood value of making the same choice from the IBL model given its set of parameters. The log-likelihood *L* is defined as:

$$L = \sum_{i=1}^{i=N} \log(Prob(Option\,X_i)) \qquad (5)$$

Where, *i* refers to the *i*th model participant playing a problem and *N* is the total number of human participants in the LM and GM groups (the model was calibrated separately to each of the two switching groups). The *log* in equation 5 is the natural logarithm and we calibrated the IBL model by minimizing the negative of the log-likelihood value (- *L*).

Furthermore, to derive a choice from the IBL model, we use the following rule: If the human chose Option X and the value of *Prob (Option X)* is greater than or equal to 0.5, then the model makes a choice like the human choice; else, the model chooses the option that is opposite of what human participant chose. We calculated the error proportion by comparing the model participant's choice to the human participant's choice.

Method

Dependent Variables

In this paper, we account for the final choices of participants with different search strategies. For this purpose, a choice made by a model participant is evaluated against a choice made by a corresponding human participant in either of the LM and GM groups, separately.

A choice in a problem is classified as maximizing if the chosen option's expected value is greater than the expected value of the non-chosen option. Those cases for which this criterion failed were termed as having non-maximizing choice. The expected value of an option was calculated by multiplying the probability of occurrence of outcomes with the outcomes and summing the multiplications together. For a model, the error proportion was calculated in a problem as:

$$ErrorProportion = (M_H N_M + N_H M_M)/(M_H N_M + N_H M_M + N_H N_M + M_H M_M) \quad \ldots (6)$$

Where, $M_H N_M$ was the number of cases where the human participant made a maximizing choice but the model predicted a non-maximizing choice. $N_H M_M$ was the number of cases where the human participant made a non-maximizing choice but the model predicted a maximizing choice. Similarly, the $M_H M_M$ and $N_H N_M$ were the number of cases, where the human participant made the same choice (maximizing or non-maximizing) as predicted by the model. Smaller the value of the error proportion, the more accurate the model is in accounting for maximizing individual choices of human participants.

Model Calibration

The IBL model described here had two free parameters *d* and *σ*. The model was calibrated on final choices for both groups, GM and LM, using a genetic algorithm program. A single set of parameters were used to calibrate the model by minimizing the negative of the Log-Likelihood value. The genetic algorithm has features that help prevent the algorithm getting trapped in local minima. The genetic algorithm repeatedly modifies a population of individual parameter tuples to find the tuple that minimizes *-L*. In each generation, the genetic algorithm selects individual parameter tuples randomly from a population to become parents and uses these parents to select children for the next generation. Over successive generations, the population evolves toward an optimal solution. The population size used here was a set of 20 randomly-selected parameter tuples in a generation (each parameter tuple was a value of *d* and σ parameters). The mutation and crossover fractions were set at 0.1 and 0.8, respectively, for an optimization over 150 generations. The model was calibrated separately in the LM and GM groups. Within each group, for each parameter tuple, the model was run 10-times across participants in a problem and the average *–L* value across 10-runs was minimized. The 10-runs ensured that the run-to-run variability in the *–L* value was small and the 10 value was derived after trying different integer values between 1 and 20 runs.

Model Results

First, we evaluated the IBL model's ability to account for final choices in the GM group. The best calibrated values of *d* and σ parameters in the IBL model were found to be 15.05 and 0.29, respectively (see Table 1). The large *d* value exhibited extreme reliance on recency during sampling. Also, the smaller σ value exhibited lower sample-to-sample variability in instance activations. The lowest value of log-likelihood obtained during calibration was -25.19.

Table 1: Parameters and Likelihood Values

Condition	Parameters	Log-Likelihood
GM	d=15.05	-25.19
	σ=0.29	
LM	d=8.82	-29.03
	σ=0.73	
GM-LDG	d=5.0	-127.07
	σ=1.5	
LM-LDG	d=5.0	-106.33
	σ=1.5	

The parameters obtained from the IBL model for the LM group were *d* = 8.82 and σ = 0.73. The value of *d* in the LM group again made participants rely on recency of information during sampling; however, this reliance on recency processes was less than that for the GM group. Furthermore, the noise parameters value represented lesser variability in activations across samples. Overall, the calibrated likelihood value was -29.03, which was slightly lesser than that in the GM group. Furthermore, the calibration of IBL model to both LM and GM groups resulted in improved likelihoods compared to the parameters suggested by Lejarraga, Dutt, and Gonzalez (2012) (*d* = 5; σ =1.5). The model parameters fitted using log-likelihoods by us in this paper are for individual participant choices in the two groups, LM and GM. However, the model parameters fitted by Lejarraga, Dutt, and Gonzalez (2012) were for choices aggregated across several participants. Given the high values of *d* parameter in our results, it seems that the recency and frequency processes are stronger among individual participants compared to the average across several participants.

Figure 1 shows proportion of final choices by model participants compared to human data in the GM and LM conditions. In both conditions, the IBL model performed like human participants: The model showed greater preferences for the fixed option compared to the variable option in both GM and LM conditions. The model's preference for fixed (variable) option was slightly higher (lower) compared to those for human participants. Due to recency effect, the model's account for human preferences was better for those who switched more and followed the piecewise search strategy compared to participants who switched less and followed the comprehensive strategy. Thus, perhaps, recency processes were more prevalent among the piecewise strategy group compared to the comprehensive strategy group.

Lastly, we analyzed the IBL model's performance in accounting for individual decisions. According to error proportion criterion, more number of N_HN_M and M_HM_M combinations help minimize the error proportion (which is desirable), while higher number of M_HN_M and N_HM_M combinations increase the error proportion. Table 2 shows the individual-level results from different LM and GM groups. As seen in Table 1, the calibrated IBL model for GM group produced 55% of N_HN_M combinations and 30% of M_HM_M combinations, respectively. In contrast, the erroneous N_HM_M and M_HN_M combinations were 12 % and 3%, respectively, from the model. Due to comparatively higher values for the N_HN_M and M_HM_M combinations in the GM group compared to the LM group, the IBL model possessed smaller error proportion in the GM group compared to the LM group. Overall, the IBL model showed superior performance for GM group compared to the LM group (15% error proportion < 32% error proportion). Finally, the error proportions from models fitted in this paper at the individual participant level were comparatively less compared to the error proportions from models fitted to the aggregate data by Lejarraga, Dutt, and Gonzalez (2012) (LDG model in GM and LM groups). Thus, it seems that fitting models using individual choices makes such models perform better compared to when the same models are fitted using aggregate choices.

Table 2: The error proportions from IBL model in the LM and GM groups

Human and Model data combination H/M	**GM**	**LM**	**GM (LDG)**	**LM (LDG)**
No. of Observations	40	40	40	40
N_HN_M	55	45	22	33
M_HM_M	30	23	13	20
N_HM_M	12	15	45	27
M_HN_M	03	18	20	20
Error Proportion	0.15	0.32	0.65	0.47

Discussion & Conclusions

So far, models in decisions from experience (DFE) paradigms had been evaluated to aggregate human choices (Gonzalez & Dutt, 2011; 2012). In such comparisons, the average risk-taking from the model was compared to the average risk-taking from human data. However, in this paper, we compared a model's performance by calibrating the model to individual human choices. More specifically, we calibrated an Instance-Based Learning (IBL) model to individual preferences with different information-search strategies. Overall, the IBL model showed superior performance when calibrated to both search-strategy groups, piecewise and comprehensive. The high value of decay parameter showed stronger reliance on recency processes among individual participants. In fact, the recency effect was stronger among participants who switched more and followed the piecewise search strategy compared to participants who switched less and followed the comprehensive strategy.

One likely reason for differing recency effect among different search strategy is that when participants use the piecewise strategy, they tend to compare the most recent outcome on one option with the most recent outcome on the other option. For this comparison to work, participant needs to rely on recent information. Furthermore, this comparison is less prevalent in the comprehensive strategy, where participants tend to search one option repeatedly before moving to investigate the other option.

In fact, the observation about high *d* parameter value for the piecewise strategy also helps us explain why the error proportion from the model was much less for the piecewise strategy compared to the comprehensive strategy. That is because recency is more suited to piecewise strategy compared to comprehensive strategy.

In this paper, we took one model of experiential choice; however, as part of future research, we plan to extend this investigation to a larger set of models and application areas. Also, it would be interesting to investigate how recency effects explain choices among different search strategies in environments where the outcomes and probabilities are non-stationary and change overtime. Some of these ideas and others form the immediate next steps for us to pursue in the near future.

Acknowledgments

The authors are grateful to Indian Institute of Technology, Mandi for providing computational resources for this paper. Also, the authors would like to thank Tata Consultancy Services' research fellowship to Neha Sharma for supporting this project.

References

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

Associates. ISBN 0-8058-2817-6

Busemeyer, J. R., & Wang, Y. (2000).Model comparisons and model selections based on the generalization criterion methodology. *Journal of Mathematical Psychology,* 44, 171–189.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Erev, I., Ert, E., Roth, A. E., Haruvy, E., Herzog, S. M., & Hau, R.(2010). A choice prediction competition: Choices from experience and from description. *Journal of Behavioral Decision Mak*ing, 23, 15–47.

Gonzalez, C., & Dutt, V. (2011). Instance-Based Learning: Integrating Sampling and Repeated Decisions From Experience. *Psychological Review*,118(4), 523-551.

Gonzalez, C., & Dutt, V. (2012). Refuting data aggregation arguments and how the instance-based learning model stands criticism: A reply to Hills and Hertwig *Psychological Review*, *Vol* 119(4),893-898.

Hertwig, R., & Erev, I. (2009). The description-experience gap in risky choice. *Trends in Cognitive Sciences*, 13, 517-523.

Hills, T. T., & Hertwig, R. (2010). Information Search in Decisions From Experience Do Our Patterns of Sampling Foreshadow Our Decisions?. *Psychological Science*.

Hertwig, R. (2012). The psychology and rationality of decisions from experience. *Synthese*, 187, 269-292.

Lebiere, C. (1999). Blending: An ACT–R mechanism for Aggregate retrievals. *Paper presented at the 6th Annual ACT–R Workshop at George Mason University*. Fairfax County, VA.

Lejarraga, T. & Dutt, V. & Gonzalez, C. (2012). Instance-Based Learning: A general model of repeated binary choice. *Journal of Behavioral Decision Making*,25: 143-153.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1, No. 1). Cambridge: MIT press.

Stutman, M., & Conklin, K. (2015). *The Ultimate Book of Inspiring Quotes For Kids.* CreateSpace Independent Publishing Platform.

Subramanyam, P.V. (2016, Nov 21). 8 biases that may lead to bad investment decisions. *Economic Times.* Retrieved from http://economictimes.indiatimes.com/wealth/invest/8-biases-that-may-lead-to-bad-investment-decisions/articleshow/55508378.cms

Quantum Entanglement, Weak Measurements and the Conjunction and Disjunction Fallacies

Torr Polakow (torrpolakow@gmail.com)

Goren Gordon (goren@gorengordon.com)
Curiosity Lab, Department of Industrial Engineering
Sagol School Of Neuroscience, Tel-Aviv University, Israel, 6997801

Abstract

The conjunction and disjunction fallacies are expressions of irrational judgments. We propose a quantum cognition model that represents each concept as a separate qubit and the measurement process as a weak measurement. Using an on-line questionnaire, we analyzed the relation between irrational judgment and the corresponding quantum state entanglement. The model enables us to follow an individuals' quantum cognitive representation throughout the questionnaire and shows that, on average, participants get more entangled as they progress in the questionnaire. Our model accounts for multiple concepts simultaneously for rational and irrational decisions and suggests that quantum entanglement of mental concepts is correlated with irrational judgments.

Keywords: Quantum cognition; Irrationality; Quantum entanglement; Probability judgment fallacies.

Introduction

People tend to make irrational decisions (Tversky and Kahneman, 1983). Irrational behavior is any behavior that reflects a violation of basic laws that stem from classical probability theory (Kolmogorov, 2013). In this paper, we focus on the conjunction and disjunction fallacies, which violate the law of total probability: The conjunction fallacy occurs when a person judges the probability of the conjunction of two events to be more likely than either of the constituent events. The disjunction fallacy occurs when a person judges the probability of the disjunction of two events to be less likely than either of the constituent events. Quantum cognition is a developing field (Busemeyer and Bruza, 2012) that takes methods and concepts from quantum probability theory and uses them to explain and model decision-making findings. The hypothesis behind quantum cognition models is that irrational behavior obeys the laws of quantum theory rather than classical probability theory.

Studies of irrational behavior using classical methods have shown that people violate the unicity principle. This assumption is broken as soon as we allow incompatible questions into the theory, which causes measurements to be non-commutative. Incompatible questions cannot be evaluated on the same basis, so they require setting up separate sample spaces. This leads to conjunction and disjunction fallacies (Tversky and Kahneman, 1983; Tversky and Shafir, 1992). Quantum probability does not assume the principle of unicity, thus allowing one to use a partial Boolean algebra; each set of questions can be answered using one sample space in a Boolean fashion. All Boolean sub-algebras are pasted together in a coherent but non-Boolean fashion.

Previous work has shown that quantum probability (QP) can be used to model cognitive fallacies, specifically, conjunction and disjunction (Trueblood and Busemeyer, 2011; Pothos and Busemeyer, 2009; Franco, 2016). Two concepts analyzed in the fallacies lie in the same Hilbert space and represent two different reference frames. This is a framework that can account for the irrationalities but not for the rational behavior.

Quantum entanglement is a unique quantum phenomenon wherein two systems cannot be described as two *separable* systems. The only way to describe their joint quantum state is by describing it as a whole (Stolze and Suter, 2004; Salimi et al., 2012). Quantum entanglement has been used in the quantum interaction community to describe joint concepts (Grdenfors, 2004; Nelson and McEvoy, 2007; Bruza et al., 2009), albeit only when considering the population and not individual participants.

The theoretical framework of quantum weak measurements describes a quantum system by a generalized quantum state that propagates from the future as well as from the past (Aharonov and Vaidman, 1991). By ”pre-selecting” an initial quantum state and “post-selecting” a final quantum state, one can describe the full dynamics of a quantum system, sometimes enabling a description of peculiar phenomena, such as the Aharonov–Bohm effect (Aharonov and Bohm, 1959). This type of measurement is called “weak measurement” and will be exploited in our proposed quantum cognitive model.

We propose a quantum model based on entanglement and weak measurements that can account for rational and irrational behaviors as well as the dynamics of the mental state of the participants. Previous studies have not addressed the dynamics of irrationality, nor have they examined entanglement–irrationality relations. In this study, an on-line questionnaire (see Methods) containing several instantiations of conjunction and disjunction fallacy scenarios was administered. Using our model, we first show that it describes *all* participant results, i.e., both rational and irrational, for all the questions. We then show that our model enables an analysis of the multi-qubit quantum mental operations of each participant regarding each question, namely, a quantitative measure of bipartite quantum entanglement.

Our analysis of the survey data reveals that irrational judgment is represented by an entangled quantum state, whereas a separable quantum state represents a rational judgment in both the conjunction and disjunction fallacies. Finally, our model enables the analysis of *dynamics* throughout the questionnaire for each participant. We show that as more information is revealed about the concepts, the more entangled these concepts become. This formulation enables a more generic, scalable and intuitive representation of cognitive concepts.

Methods

Conjunction and disjunction fallacies. In our formalism, for two concepts A and B: (i) The conjunction fallacy occurs when $p(A \cap B) > \min(p(A), p(B))$, i.e., when the probability of the conjunction is greater than either of the constituent probabilities. (ii) The disjunction fallacy occurs when $p(A \cup B) < \max(p(A), p(B))$, i.e., when the probability of the disjunction is smaller than either of the constituent probabilities. We defined the irrationality measure as follows:

$$irr = p(A \cap B) - \min(p(A), p(B)) \qquad \text{conj. (1)}$$

$$irr = \max(p(A), p(B)) - p(A \cup B) \qquad \text{disj. (2)}$$

In the analysis, we defined an answer as irrational only if $irr > 0.1$, i.e., this is a stricter condition for irrationality.

Questionnaire. To study the conjunction and disjunction fallacies, we used the following personality sketches of two fictitious individuals, Emma and Liz, followed by a set of occupations and avocations associated with each or both of them. In each question, the participants were asked to give a probability for each option by using a horizontal slider/bar. All the options were initialized to the neutral probability value of 0.5.

We briefly outline the questionnaire as follows:

Q1: *Emma is outgoing and lives in an apartment within the center of the city with her two cats. She takes yoga classes at the gym three times a week, enjoys reading science-fiction books and volunteers in an animal shelter at least once a week. For each statement, please move the horizontal slider to represent how much do you think the statement represents Emma.*

Emma is a manger. $(= p(A))$
Emma is a pianist and a runner.
Emma is a writer.
Emma likes to paint.
Emma is vegan.
Emma likes to exercise. $(= p(B))$
Emma is a manger and likes to exercise. $(= p(A \cap B))$
Emma is a blogger.

Q2: *Liz lives in Oakland in a Victorian house. She is an analytical thinker and works in a start-up. In addition, she tries to go to a few classes at the gym every week. She is very ambitious in her job. She enjoys cooking very much and she is very good at it. She also likes camping. For each statement, please move the horizontal slider to represent your opinion about Liz.*

Is Liz a programmer? $(= p(C))$
Does Liz like to paint? $(= p(D))$
Is Liz a programmer and likes to paint? $(= p(C \cap D))$

Q3: *Emma and Liz want to do an extracurricular activity together.For each statement, please move the horizontal slider to represent how likely it is they will choose this activity and why.*

Take spinning class, because Emma likes to exercise. $(= p(B))$
Try out gourmet restaurants in the city.
Take realistic painting classes, because Liz like to paint. $(= p(D))$
Take singing classes near Emma's apartment.
Take photography class near Liz house.
Take spinning class or realistic painting classes. $(= p(B \cup D))$

Q4: *Recently Emma got to the conclusion that she doesn't have enough time in a day. For each statement, please move the horizontal slider to represent how likely it is that this is the reason Emma needs more time.*

For her work, as manager. $(= p(A))$
Volunteer.
Exercise. $(= p(B))$
Join Book club, because she likes to read.
To work as manager or to exercise. $(= p(A \cup B))$
Meet with friends.

Participants were 100 Amazon Mechanical Turk users. We asked for Amazon Mechanical Turk Masters that were native English speakers from North America that had completed at least 100 tasks with an approval rate $> 95\%$. Each participant was presented with the questionnaire on Qualtrics. After completing the questionnaire, the participants had to pass three screening checks:

a) "Trap" question - we inserted a question that contained internal text telling the participant how to answer. Participants that answered incorrectly were excluded (15 participants). **b)** Response time - participants who answered too quickly/slowly, i.e., more than a 3σ deviation from the mean response time in either direction were excluded (8 participants). **c)** "Focus" - Participants that answered too many questions (3σ deviation from the mean "focus") with the probabilities $0, 0.5, 1$, i.e., they did not pay attention to the answers, were excluded (2 participants).

After screening, 78 participants were left (there were participants who failed more than one screening test).

Results

Participant Irrationality

We first present the data from the on-line survey we performed, Fig. 1. The survey included four questions; the first and second measured the conjunction fallacy, while the third and fourth measured the disjunction fallacy. As can be seen, the percentage of irrational judgments replicates previous reporting (Charness, 2009). Furthermore, only 8 out of the 78 participants were rational in all questions.

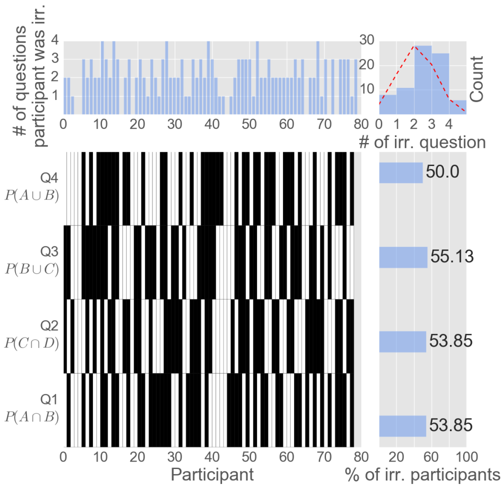


Figure 1: Top-left: Number of irrational answers per participant; Bottom-left: (ir)rational answers (white=rational, black=irrational); Top-right: histogram of the number of irrational answers per participant; Bottom-right: distribution of the irrationality value per question.

These data suggest that a questionnaire involving multiple questions and different types of fallacies can reveal the ubiquity of irrational judgments.

Moreover, as detailed in the next sections, analyzing each participant individually throughout the questionnaire enables a glimpse into the *dynamics* of irrational decision making. For example, comparing the first and last questions, which asked about the same concepts, reveals that only 45 participants (58%) maintained their "rationality", i.e., answered both questions (ir)rationally. While this can be interpreted as inconsistency, the participants were given more information about the concepts between the two questions and thus may represent a dynamic mental process (see below).

Weak Measurements of Concept-Qubits

In our model, we propose that each concept is represented by a single qubit: Concept A is represented by $|\psi\rangle_A$, while a different concept, B is represented by another qubit, $|\psi\rangle_B$. Thus, the complete two-concept quantum state is represented by

$$\begin{aligned}
|\psi\rangle_{AB} &= a_{00}|0\rangle_A|0\rangle_B + a_{10}|1\rangle_A|0\rangle_B + \qquad (3)\\
&\quad a_{01}|0\rangle_A|1\rangle_B + a_{11}|1\rangle_A|1\rangle_B\\
|\sum_{ij} a_{ij}|^2 &= 1 \qquad (4)\\
P(A) &= Tr_B({}_A\langle 1|\psi\rangle_{AB}\langle\psi|\psi_0\rangle_A) \qquad (5)\\
&= (1/\sqrt{2})(a_{10}(a_{00}+a_{10})+a_{11}(a_{01}+a_{11}))\\
P(B) &= Tr_A({}_B\langle 1|\psi\rangle_{AB}\langle\psi|\psi_0\rangle_B) \qquad (6)\\
&= (1/\sqrt{2})(a_{01}(a_{00}+a_{01})+a_{11}(a_{10}+a_{11}))\\
P(A\cap B) &= {}_A\langle 1|_B\langle 1|\psi\rangle_{AB}\langle\psi|\psi_0\rangle_B|\psi_0\rangle_A \qquad (7)\\
&= (1/2)a_{11}(a_{00}+a_{10}+a_{01}+a_{11})\\
P(A\cup B) &= ({}_A\langle 1|_B\langle 0| +_A \langle 0|_B\langle 1| +_A \langle 1|_B\langle 1|)\times\\
&\quad \psi\rangle_{AB}\langle\psi|\psi_0\rangle_B|\psi_0\rangle_A \qquad (8)\\
&= (1/(3\sqrt{2}))(a_{10}+a_{01}+a_{11})\times\\
&\quad (a_{00}+a_{10}+a_{01}+a_{11})
\end{aligned}$$

This representation has three free parameters due to normalization (eq. (4)) as compared to the two parameters of previous models (Trueblood and Busemeyer, 2011; Pothos and Busemeyer, 2009; Franco, 2016). For each participant and each conjunction (disjunction) question, we obtain three reported probabilities, namely, $p(A), p(B)$ and $p(A\cap B)$ (or $(p(A\cup B))$.

We introduce *weak measurements* as the measurement process in our model (Aharonov and Vaidman, 1991). For new questions regarding concepts about which there is no information, the pre-selected state is given by the fully superposed state $|\psi_0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$. The post-selected state is the answer in the questionnaire, in our case, always $|1\rangle$ of the relevant concept-qubit. The mental quantum operation each participant performs in each question transforms the initial state to the final one. This is represented by $|\psi\rangle_{AB}\langle\psi|$. In other words, the participants' mental process of how they incorporate new information is represented by a projection operator.

This model enables the calculation of the full quantum mental state representation given the reported question probabilities. Under the formalism from eq. (3) we denote the constraints eqs. (4) and (8).

For the conjunction questions, we used eqs. (4)–(7)), and for the disjunction questions, we used eqs. (4)–(6) and (8)). We numerically solved this set of four non-linear equations with four variables, which resulted in a full quantum state for each participant and each question.

Entanglement and Irrationality

While the calculation of the full quantum state from the probabilities does not generate any prediction, it does enable us to calculate entanglement. We calculated the two-qubit pure-state entanglement using the concurrence measurement (Stolze and Suter, 2004):

$$C(|\psi\rangle) = 2\cdot|a_{00}\cdot a_{11} - a_{01}\cdot a_{10}| \qquad (9)$$

where $C\in[0,1]$, so that if $C=0$, the state is factorized, whereas if $C>0$, the state is entangled. In the data analysis, we defined a stricter threshold for entanglement, namely, a state representing a participant's answer is considered entangled only if $C>0.2$.

This quantum entanglement calculation enables us to analyze its relation to the amount of irrationality of the reported probabilities, eq. (1). Hence, we can compute both entanglement and irrationality for each participant and each question , as shown in fig. 1.

As can be seen in Fig. 2, with our strict definitions of irrationality and the entangled state, only two out of the 78 participants were both irrational and non-entangled in this question. Fig. 3 shows that this "quadrant" was sparse in all questions, i.e., out of all the participants/questions (312 in total), only 5.8% (18 answers) were irrational and separable. More quantitatively, we can compute the following conditional probabilities:

$$\begin{aligned}
P(\text{rational}|\text{low entanglement}) &= 75\% \qquad (10)\\
P(\text{high entanglement}|\text{irrational}) &= 81\% \qquad (11)
\end{aligned}$$

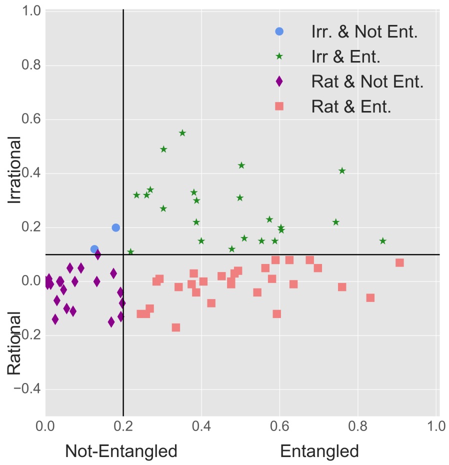


Figure 2: Irrationality as a function of Entanglement (concurrence calculation).

Finally, the questionnaire enables us to follow the dynamics of irrational judgments and the ensuing entanglement. As can be seen in Fig. 3, entanglement is monotonically non-decreasing as the questionnaire progresses. This is expected as more information regarding the concepts is revealed, i.e., as the story of Emma and Liz unfolds. The intricate connections between the storylines generate a quantum entanglement of the representative quantum states.

Discussion and Future Work

We have presented a quantum model with respect to the conjunction and disjunction fallacies that represents each concept as a separate qubit and treats the questions as quantum weak measurements.

While previous quantum cognitive models have treated concepts as qubits, (Busemeyer and Bruza, 2012), they have done so on overall data, i.e., by aggregating answers from many participants, thus representing the "concepts" as a whole. In contrast, our model attempts to represent *individual* mental states of participants by fitting their answers to a specific quantum projection operator within the weak-measurement framework. This framework assumes that participants start with an ignorant representation of the concept, represented as a full superposition of all possible representations as the "preselected" quantum state. The framing of the question "post-selects" the end quantum state, which enables us to fit the quantum operator, represented as a projection of a full quantum state of both concepts, from the data.

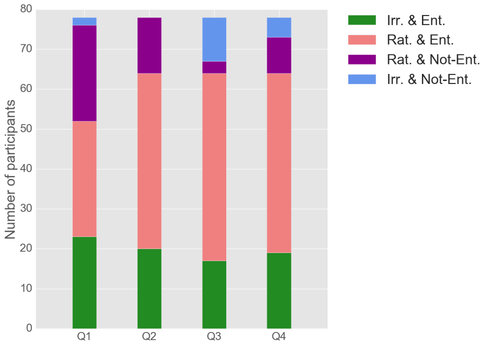


Figure 3: Distribution of rational/irrational and entangled/non-entangled participants for all the questions.

This representation gives new insights into the connection between the quantum mental representation of concepts and irrational judgments. More specifically, the data suggest that irrational judgments mostly occur for entangled quantum states, whereas separable states occur mostly when rational behavior is observed.

Cognitively, one can speculate that rational judgment regarding two concepts implies that they are separable and do not relate to or influence each other. This separability thus conveys no "cognitive interference" that can cause irrational judgments. On the other hand, highly entangled concepts, i.e., concepts that relate to and influence each other in a tight manner, will result in more irrational judgments.

Our participant–question-specific model enables us to analyze the *dynamics* of entanglement throughout the questionnaire. As hypothesized by the connection between concepts and entanglement, the more information is revealed throughout the questionnaire regarding the concepts, the more entangled they become. We have shown that entanglement indeed rises on average, and more specifically, that for the same question, for answers at the beginning and the end of the questionnaire, entanglement increased.

The proposed model holds promise in the form of scalability. The previous concept-as-basis models did not scale well when introducing more than two concepts, since introducing even a single new concept immediately imposes *two* relations between the previous two concepts. This occurs since all concepts lie in the same Hilbert space. Our concept-as-qubit model enables the introduction of more concepts, as they expand the Hilbert space and enable arbitrary relations between the concepts. While the increase in free parameters is exponential in the number of concepts in our model, measures can be computed from the inferred quantum state; the most promising are multipartite entanglement measures. Future work will explore this direction with a more detailed questionnaire that involves more than two qubits and their interaction.

References

Aharonov, Y. and Bohm, D. (1959). Significance of Electromagnetic Potentials in the Quantum Theory. *Physical Review*, 115(3):485–491.

Aharonov, Y. and Vaidman, L. (1991). Complete description of a quantum system at a given time. *Journal of Physics A: Mathematical and General*, 24(10):2315.

Bruza, P., Kitto, K., Nelson, D., and McEvoy, C. (2009). Extracting Spooky-Activation-at-a-Distance from Considerations of Entanglement. In *Quantum Interaction*, pages 71–83. Springer, Berlin, Heidelberg.

Busemeyer, J. R. and Bruza, P. D. (2012). *Quantum Models of Cognition and Decision*. Cambridge University Press.

Charness, G. (2009). On the Conjunction Fallacy in Probability Judgment: New Experimental Evidence Regarding Linda. Economics Working Paper Archive 552, The Johns Hopkins University,Department of Economics.

Franco, R. (2016). Towards a new quantum cognition model. *arXiv:1611.09212 [quant-ph, q-bio]*. arXiv: 1611.09212.

Grdenfors, P. (2004). *Conceptual Spaces: The Geometry of Thought*. A Bradford Book.

Kolmogorov, A. N. (2013). *Foundations of the Theory of Probability*. Martino Fine Books. Google-Books-ID: a5ubnQEACAAJ.

Nelson, D. and McEvoy, C. (2007). Entangled Associative Structures and Context. *Journal of Physics A: Mathematical and General*, AAAI Spring Symposium: Quantum Interaction:98–105.

Pothos, E. M. and Busemeyer, J. R. (2009). A quantum probability explanation for violations of rational decision theory. *Proceedings of the Royal Society of London B: Biological Sciences*, page rspb.2009.0121.

Salimi, S., Mohammadzade, A., and Berrada, K. (2012). Concurrence for a two-qubits mixed state consisting of three pure states in the framework of SU(2) coherent states. *Quantum Information Processing*, 11(2):501–518.

Stolze, J. and Suter, D. (2004). *Quantum Computing: A Short Course from Theory to Experiment*. John Wiley & Sons.

Trueblood, J. S. and Busemeyer, J. R. (2011). A quantum probability account of order effects in inference. *Cognitive Science*, 35(8):1518–1552.

Tversky, A. and Kahneman, D. (1983). Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review*, pages 293–315.

Tversky, A. and Shafir, E. (1992). The Disjunction Effect in Choice under Uncertainty. *Psychological Science*, 3(5):305–309.

Data informed cognitive modelling of offshore emergency egress behaviour

Jennifer Smith (jennifersmith@mun.ca)
Faculty of Engineering & Applied Science, Memorial University of Newfoundland
St. John's, Newfoundland and Labrador, Canada A1B 3X5

Mashrura Musharraf (mm6414@mun.ca)
Faculty of Engineering & Applied Science, Memorial University of Newfoundland
St. John's, Newfoundland and Labrador, Canada A1B 3X5

Brian Veitch
Faculty of Engineering & Applied Science, Memorial University of Newfoundland
St. John's, Newfoundland and Labrador, Canada A1B 3X5

Abstract

This paper applies a cognitive modelling approach to model decision making of naïve subjects in virtual emergency situations. Virtual environments (VE) can be used as a virtual laboratory to investigate human behaviour in simulated emergency conditions. Cognitive modelling methodology and human performance data from VEs can be used to identify the problem solving strategies and decision making processes of general personnel in offshore emergency egress situations. This paper demonstrates the utility of decision trees as a cognitive tool for two main purposes: 1) assessing VE training curriculum and 2) predicting human behaviour. To show these capabilities, the results of two empirical studies are compared using a decision tree induction approach. The first experiment investigated the learning and inference process of participants trained using a lecture based teaching (LBT) approach. The second experiment used another pedagogical approach – simulation-based mastery learning (SBML). Overall, decision trees were found to be a useful method for evaluating the efficacy of VE training, and as a basis for predicting individuals' decision-making performance.

Keywords: decision trees; decision making in emergencies; virtual environments; offshore emergency egress; training efficacy

Introduction

Offshore oil and gas platforms operate in remote and harsh maritime environments. As a result, offshore emergencies are complex, dynamic, and high-risk situations. Personnel responding to these emergencies are faced with uncertainty in managing the situation, and major time pressure in safely evacuating the platform. Decision making in high-stress emergency situations can vary from person to person. This variability could be a result, in part, of conventional training in which people tend to employ different learning strategies and develop their understanding of emergency protocols differently (Musharraf et al., 2016). However, individual differences and unpredictable responses to emergency situations can undermine the emergency response operations. Therefore, effective training in emergency response and preparedness is critical for ensuring offshore safety.

Virtual environments (VE) can address existing training gaps and augment conventional offshore safety training by providing artificial experience that would otherwise be too dangerous to practice (Smith et al., 2017). VE technology can allow offshore operators to familiarize personnel with the worksite and to practice emergency exercises before going offshore. However, verification of the VE training curriculum is required to confirm it meets the intended training purposes.

Cognitive modelling methodology can be used to inform the quality of VE training. Developing a cognitive model of human behaviour in these virtual emergency situations can provide valuable insight with regards to improving offshore safety systems and training programs. The VE allows researchers to observe how humans use information to accomplish specific tasks (Musharraf et al., 2016; Roth et al., 1992). Cognitive modelling methodology and human performance data from virtual environments can be used to identify the problem solving strategies and decision making process (e.g. model the knowledge base and inference process) of personnel in offshore emergency situations.

This paper demonstrates the use of a cognitive modelling methodology – decision trees – to evaluate the efficacy of VE training. This approach was introduced by Musharraf et al. (2016) and is based on two experimental studies that investigated the effectiveness of VE training curriculum on competence. The model focuses on the decision making process of naïve subjects in virtual emergency situations, particularly the participants' route selection strategies. The first experiment involved lecture-based teaching (LBT). Participants in the experiment showed variability in responding to emergency situations. The variability manifested itself in many different decision strategies. To address this variability and to improve learning outcomes, a second experiment was designed, which employed a different pedagogical approach called simulation-based mastery learning (SBML) (McGaghie et al., 2014). Subsequently, the in-simulation performance of participants from both studies was compared using decision trees.

The paper describes the theoretical framework, data collection process, and how the knowledge bases were created. Further, it explains the algorithm that runs the inference engine to produce the decision trees. A process for testing the prediction accuracy of the decision trees is also described.

Theoretical Background

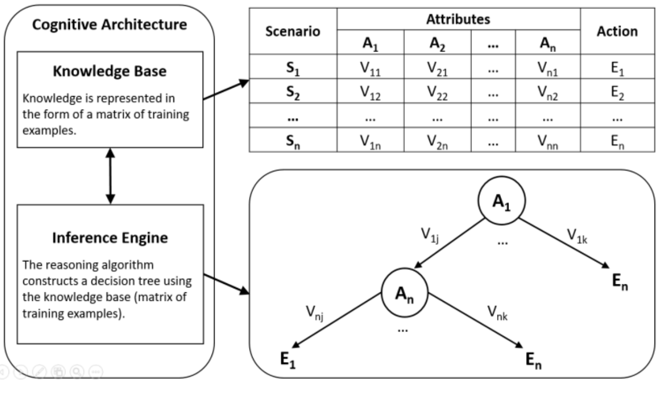
Cognitive Functions

Four major cognitive functions are performed by personnel in emergency egress situations: perception, interpretation, decision making, and execution. For example, in an emergency, personnel hear an alarm and are required to muster at their designated muster or lifeboat stations by following a safe egress route. These cognitive functions are repeated based on the personnel's situational awareness and whether they encounter hazards or obstructed routes.

- Perception – perceive audio-visual cues from the environment.
- Interpretation – analyze the perceived cues and infer what the alarm and public address (PA) mean (i.e. which route is obstructed, where to muster).
- Decision Making – assess different potential egress routes and choose the safest path.
- Execution – follow egress route until designated muster or lifeboat station is reached.

Learning and Inference

This paper investigates how people develop and use different problem solving strategies, specifically route choices, given their VE training. This is modeled by a knowledge base and an inference engine. In the model, all the knowledge gained from training and experience in the VE is stored in a knowledge base. The content of the knowledge base is then used by the inference engine to develop a human reasoning structure. Figure 1 shows the inference process.



Knowledge is represented in the form of a matrix of training examples.

The reasoning algorithm constructs a decision tree using the knowledge base (matrix of training examples).

Scenario	Attributes				Action
	A_1	A_2	...	A_n	
S_1	V_{11}	V_{21}	...	V_{n1}	E_1
S_2	V_{12}	V_{22}	...	V_{n2}	E_2
...	...	...	...	...	...
S_n	V_{1n}	V_{2n}	...	V_{nn}	E_n

Figure 1: Knowledge Base and Inference Engine

Knowledge in the knowledge base is represented using a matrix of training examples. Scenarios are represented by S_1, S_2,…, S_n. Attributes of the scenario are represented by A_1, A_2,…, A_n. Actions taken are represented by E_1, E_2,…, $E_{n.}$ The values of the attributes are represented by V_{ij}, where the value represents the j^{th} value of the i^{th} attribute. The knowledge matrix is used by the inference engine to construct a reasoning algorithm. An inductive reasoning approach – decision tree – was used in this paper. Decision trees were selected based on their visual simplicity and diagnostic capabilities. Decision trees can be constructed relatively quickly compared to other methods, such as artificial neural networks, or support vector machines (Duffy, 2009). Another benefit is that they do not require any prior assumptions about the data.

Decision Tree Induction

The goal of the decision tree induction is to classify the content in the knowledge base into groups such that the data set in each group belongs to the same class (Badino, 2004). The classification is performed based on the value of selected attributes. Several attribute selection measures are available, including *information gain, gain ratio, and Gini index*. This paper uses the ID3 decision tree algorithm, which uses information gain as an attribute selection measure (Han et al., 2011). Information gain is calculated using the idea of entropy. Given the entropy of a data set *S*, information gain of an attribute *A* can be calculated using equation 1.

$$Gain\ (A) = Entropy\ (S) - Entropy\ (A) \qquad (1)$$

Here, *Entropy(A)* presents the weighted average uncertainty of the groups created by classifying the data set using attributes (A_i). Details of entropy calculation can be found in Han et al. (2011). The decision tree algorithm takes two basic inputs: the data set in the knowledge base and the list of scenario attributes. During the decision tree induction, data are iteratively classified using the attribute that has the highest information gain, as highest gain refers to lowest uncertainty. The following steps are repeated until no attributes are left for classification, or the data set is empty, or data in each group belong to the same class and no further classification is needed.

Step 1: For each attribute A_i, compute the value of information gain $Gain(A_i)$.

Step 2: Choose the attribute with the highest gain $Gain(A_i)$ and classify remaining data set based on A_i.

More details on the decision tree algorithm can be found in Musharraf et al. (2016).

Experimental Methodology

Two experiments were conducted: the first focused on conventional LBT methodology and the second focused on SBML. Both studies used a VE called the All-hands Virtual Emergency Response Trainer (AVERT). This section will describe the training, data collection in AVERT, formation of the knowledge base, and resulting decision trees.

AVERT Simulator

AVERT is a first person perspective VE that was developed to train basic offshore safety practices to general personnel –

individuals whose responsibility in an emergency is to muster at their designated muster stations (House et al., 2014). AVERT scenarios involve basic wayfinding, alarm drills, and emergency response exercises. AVERT delivers training scenarios, tracks in-simulation performance metrics, and provides corrective feedback.

LBT was used to train 36 participants in how to successfully muster during offshore emergency situations in the VE. Participants attended three separate sessions. Each session involved a computer based training tutorial, followed by four training scenarios, and four testing scenarios in AVERT. The content of the tutorials included basic offshore emergency preparedness, alarm recognition and assessing the emergency situation, and hazard avoidance. Participants only received one exposure to each scenario and were provided minimal feedback on their performance. Details of the study can be found in (Smith et al., 2015). Data from 17 of the participants (13 male and 4 female, with a mean age of 26.8 years, standard deviation of 5.0 years) were used in this paper for comparison to the SBML approach.

SBML was used to train 55 participants in offshore emergency egress using the VE. This pedagogical approach was used to ensure that participants acquired and demonstrated the knowledge and skills necessary before advancing to more complex emergency situations. SBML involved a series of four training modules. Each module was designed to train specific learning objectives and gradually taught participants the platform layout, how to recognize alarms, what to do in the event of blocked routes, as well as how to assess the situation and avoid hazards while evacuating the platform. As part of the SBML training, participants were required to achieve demonstrated competence in all training and testing scenarios. The participants were tested repeatedly on their competence over the course of the training modules. They received detailed feedback on their performance after each attempt of a scenario. To achieve demonstrated competence, some participants required multiple attempts at the scenarios. Details of the study can be found in (Smith et al., 2017). Data from 15 randomly selected participants (12 male and 3 female, with a mean age of 25.6 years, standard deviation of 8.0 years) were used for comparison.

Data Collection and Modelling

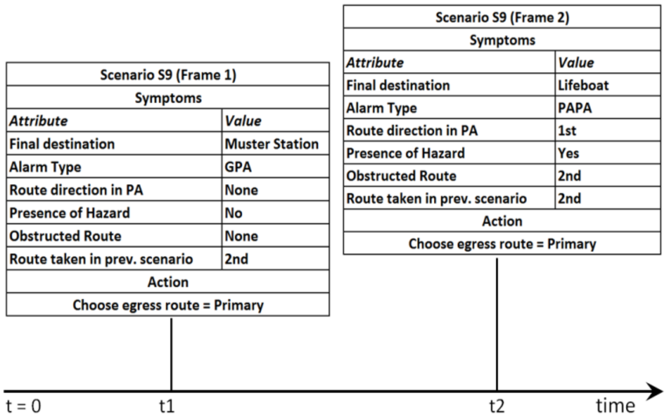
Two training modules were the focus of the decision tree analysis: the ‘Alarm Recognition’ and ‘Assessing Situation’ modules. In both the LBT and SBML approaches, participants had to perform in twelve scenarios. The training scenarios differed between the training approaches as the SBML training provided more in-simulation instruction and feedback. However, the testing scenarios were the same for both training approaches. A subset of scenarios was used to populate the knowledge base (8 and 9 scenarios for the LBT and SBML training, respectively). Half of the scenarios were used to generate the knowledge base for the ‘Alarm Recognition’ module (denoted KB1) and the remaining scenarios were added to the knowledge base for the ‘Assess Situation’ module (denoted KB2). Two test scenarios were used to test the prediction capabilities of the decision trees (scenarios T1 and T2).

Knowledge Matrix Following rule based methodology, a knowledge matrix was created using the data from the participant’s performance in the training scenarios. Data to populate the knowledge matrix was collected from the AVERT report files generated for each scenario and from observations logged in-situ. Table 1 lists the attributes varied for each scenario and their possible values.

Table 1: Possible values for each attribute.

Attribute	Possible Values
Final destination	Muster, Lifeboat
Alarm type	None, GPA, PAPA
Hazard presence	No, Yes
Route directed by PA	None, 1st, 2nd
Obstructed route	None, 1st, 2nd
Previous route selected	1st, 2nd

Scenario Frames Participants were required to complete a series of scenarios of varying complexity. Basic scenarios involved participants practicing their egress routes and muster procedures. More complex emergency scenarios were dynamic in the sense that the value of some attributes changed during the scenarios. To capture the dynamic aspect, these scenarios were split into two or more frames. Figure 2 shows an example of two frames for a training scenario (S9) and how the knowledge matrix is updated based on the change in attributes of the scenario.



Scenario S9 (Frame 1)	
Symptoms	
Attribute	*Value*
Final destination	Muster Station
Alarm Type	GPA
Route direction in PA	None
Presence of Hazard	No
Obstructed Route	None
Route taken in prev. scenario	2nd
Action	
Choose egress route = Primary	

Scenario S9 (Frame 2)	
Symptoms	
Attribute	*Value*
Final destination	Lifeboat
Alarm Type	PAPA
Route direction in PA	1st
Presence of Hazard	Yes
Obstructed Route	2nd
Route taken in prev. scenario	2nd
Action	
Choose egress route = Primary	

Figure 2: Example of scenario frames F1 and F2 for S9.

Table 2 shows the state of the knowledge base for a sample participant in the SBML program after finishing all training modules. Each row in the knowledge base contains the values of different attributes for the scenario and the corresponding action. For both studies, the participants’ perceived scenario attributes and corresponding actions for

Table 2: Knowledge matrix for alarm recognition (KB1) and assessment emergency (KB1 and KB2) training modules.

Category	Scenario	Attributes						Actions
		End Point	Alarm	Route by PA	Hazard	Blocked Route	Previous Route	
KB1	S1	Muster	None	1st	No	None	N/A	Primary
	S2 (F1)	Lifeboat	None	1st	No	None	1st	Primary
	S2 (F2)	Muster	None	2nd	No	None	1st	Secondary
	S3	Lifeboat	None	None	No	None	2nd	Primary
	S4	Muster	GPA	None	No	None	1st	Primary
	S5	Muster	GPA	None	No	None	1st	Primary
Test 1	T1	Muster	GPA	1st	No	None	1st	Primary
KB2	S6	Lifeboat	PAPA	1st	No	2nd	1st	Primary
	S7	Lifeboat	PAPA	2nd	Yes	1st	1st	Secondary
	S8	Lifeboat	GPA	2nd	Yes	1st	2nd	Secondary
	S9 (F1)	Muster	GPA	1st	Yes	2nd	2nd	Primary
	S9 (F2)	Lifeboat	PAPA	1st	Yes	2nd	2nd	Primary

each scenario were included as entries in the knowledge base. Because the SBML training required participants to reattempt scenarios until they correctly completed the task, only successful route strategies were stored as entries in the knowledge matrix.

Decision Trees Decision trees visualize how participants formed decisions based on the knowledge matrix. Decision trees also provide insights on what attributes had the biggest impact on participants' decision making. Figure 3 shows a decision tree based on the knowledge matrix in Table 2.

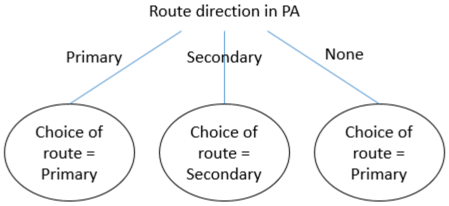

Route direction in PA
Primary
Secondary
None
Choice of route = Primary
Choice of route = Secondary
Choice of route = Primary

Figure 3: Decision tree developed after KB1.

In this case, the participant's route selection was decided based on their understanding of the information from the PA announcement. If the PA directed them to a safe route, then the participant took that route. If the PA did not provide any information regarding the safety of the route options, then the participant's choice defaulted to their primary egress route. The tree can subsequently be used to predict participants' choice of route (i.e. primary or secondary) for a given future scenario.

Results

For presentation purposes, the participants' decision trees after module 2 and 4 ('Alarm Recognition' and 'Assess the Situation') of the SBML experiment were developed to see how the trees evolved as more training content was added to the knowledge base. The different decision trees are summarized in Table 3. The detailed decision trees for the LBT experiment can be found in Musharraf et al. (2016).

Comparing the Alarm Recognition Decision Tree

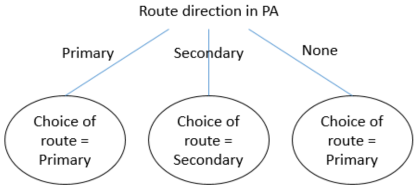
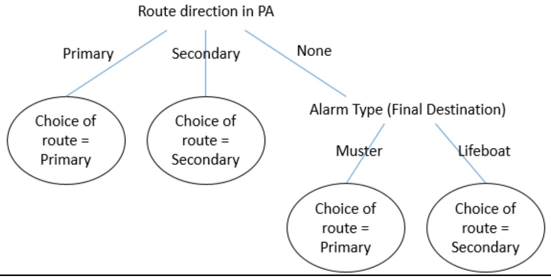
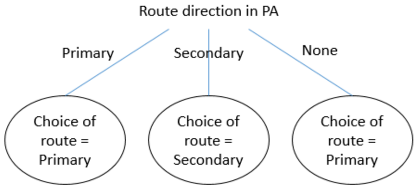
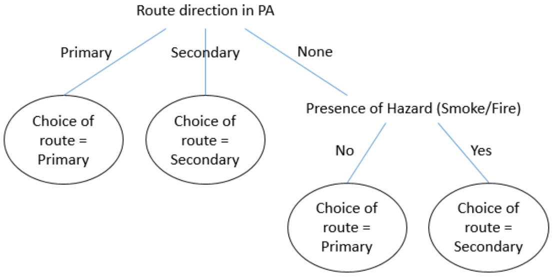
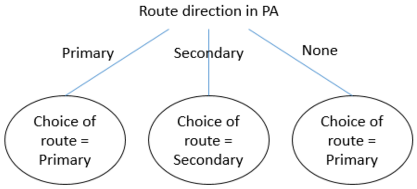
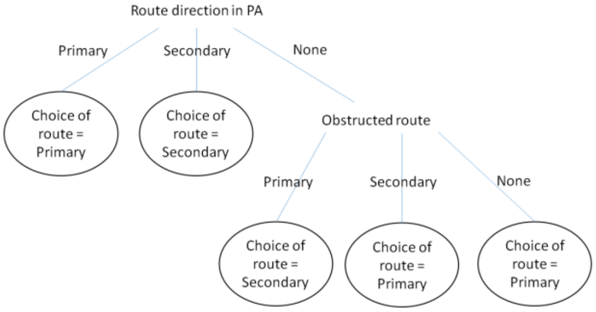
In an emergency situation, the alarm type dictates the final muster location. The main learning objective for this module was for participants to listen to the alarm and the PA announcement and take the safest route available in response to the situation. A decision tree for this situation is depicted in Figure 3. Eighty percent of participants in the SBML study developed the decision tree depicted in Figure 3 before the test scenario (T1). Forty one percent of participants in the LBT study had the same decision tree. Twenty percent of SBML and 24% of LBT participants based their route decision on alarm type or end point when the PA did not provide any route information. In this case, the participants interpreted that the general platform alarm (GPA) meant taking the primary route, and that the prepare to abandon platform alarm (PAPA) meant taking the secondary route.

Comparing the Assess Emergency Situation Tree

In an emergency situation, it is critical that personnel listen to the PA announcement, continually assess their surroundings, and follow the safest egress route available. If personnel encounter an obstructed route, they must re-route in response to the hazardous situation. Building on earlier learning objectives, the 'Assess the Situation' module trained participants how to assess the emergency situation, avoid hazards, and follow the safest egress path to the designated muster or lifeboat station.

It was expected that most participants would select the safest route based on the information in the PA announcement. In the SBML study, 67% of participants continued to use the same decision tree, selecting their egress route based on PA information (as shown in Table 3). In the LBT study, only 24% of participants had the same

Table 3 – Resulting decision trees for 15 SBML participants after finishing training module 2 (S5) and module 4 (S9).

Subject	Decision rules until test scenario T1	Decision rules until test scenario T2(F1-F3)
A02, A06 A10, A19 A22, A38 A44, A45 A53, A60	 Route direction in PA Primary Secondary None Choice of route = Primary Choice of route = Secondary Choice of route = Primary	Remains the same.
A27 A33 A62	 Route direction in PA Primary Secondary None Choice of route = Primary Choice of route = Secondary Alarm Type (Final Destination) Muster Lifeboat Choice of route = Primary Choice of route = Secondary	Remains the same.
A29	 Route direction in PA Primary Secondary None Choice of route = Primary Choice of route = Secondary Choice of route = Primary	 Route direction in PA Primary Secondary None Choice of route = Primary Choice of route = Secondary Presence of Hazard (Smoke/Fire) No Yes Choice of route = Primary Choice of route = Secondary
A42	 Route direction in PA Primary Secondary None Choice of route = Primary Choice of route = Secondary Choice of route = Primary	 Route direction in PA Primary Secondary None Choice of route = Primary Choice of route = Secondary Obstructed route Primary Secondary None Choice of route = Secondary Choice of route = Primary Choice of route = Primary

decision tree. Similarly, 20% of the SBML participants continued to use the strategy in which the alarm type or end point indicated the route choice in the absence of a PA.

When participants failed to perceive the PA instructions, some individuals put emphasis on different attributes to make their decision. Some participants followed the alarm type and PA, whereas others considered the presence of hazards, or route obstructions. Thirteen percent of participants in the SBML study demonstrated more complex decision trees to manage the emergency conditions. Conversely, the remaining participants in the LBT study (76%) had more varied behaviours. The following summarizes the strategies observed for LBT participants:

- 41% developed complex decision trees that incorporated special conditions for the PA announcements, alarm type or end points, obstructed routes, and hazards.
- 12% selected the same route regardless of the emergency conditions.
- 23% appeared lost. Decision trees were not developed for these participants as they were unable to form a generalization from the knowledge base.

Prediction Accuracy of the Decision Trees

To determine the accuracy of the decision trees, they were used to predict decision making in subsequent scenarios. Specially, they were used to predict the participants' route selection in test scenarios (T1 and T2). The predictions were compared to the actual routes the participants took in those scenarios. The prediction accuracy was calculated based on the average number of successful matches between the decision tree predicted outcomes and the observed outcomes of the participant. Table 4 shows the results for the SBML study. The decision trees were able to predict the route selection of participants with 94% accuracy.

Table 4: Percentage Prediction Accuracy.

Participant No.	% Prediction Accuracy
A02	100
A06	80
A10	100
A19	100
A22	80
A27	100
A29	100
A33	75
A38	100
A42	100
A44	100
A45	100
A53	100
A60	75
A62	100
Average	94

Efficacy of LBT and SBML Training

Overall, the SBML participants' behaviours in responding to emergencies over the course of their exposure to several scenarios gradually converged to a few expected decision trees. Conversely, the LBT participants' behaviours in responding to emergencies diverged. At the alarms recognition phase, the participants in the SBML study had 2 different strategies and the participants in the LBT study had 6 different strategies. In the advanced emergency phase, the SBML participants had 4 different strategies and LBT participants had 10 different strategies for assessing the emergency conditions and safely evacuating the platform. All of the observed route strategies for the SBML participants led to the successful completion of the test emergency scenario.

Many of the LBT participants had a poor understanding of the egress procedures and were overall less compliant with rules. In general, participants in the LBT study put more weight on attributes that were not necessarily useful in making egress decisions. The variability and incorrect behaviours modeled in the decision trees by the LBT training show that this form of training was inadequate for preparing participants for emergency conditions. The SBML training resulted in higher safety compliance and more concise decision trees. This suggests that participants from SBML training were better equipped for managing the emergency scenarios. It is likely that these positive results are because the SBML study placed more emphasis on training participants to pay attention to the PA and act according to the directions of the PA. It may also be due to the fact that the SBML participants were required to practice the task until competence was demonstrated. The results of this study show that SBML training resulted in decision trees that better reflect competence and reduced variance in safety compliance in comparison to the LBT training.

Conclusion

Modelling human behaviour in emergency conditions can be difficult. The paper outlined a cognitive modeling approach that is suitable for modeling decision making and predicting human response in virtual emergency scenarios. The decision tree modeling approach was shown to be appropriate for assessing the training efficacy of two different training programs: lecture based training (LBT) and simulation based mastery learning training (SBML). The visual representation of the participants' strategies in emergency situations was useful in identifying the strengths and weaknesses of the training methods. Decision tree modelling could help inform the design and assessment of future VE training curriculum and predict the performance of general personnel in emergency situations.

Acknowledgments

The authors acknowledge with gratitude the support of the NSERC-Husky Energy Industrial Research Chair in Safety at Sea.

References

Badino, M. (2004). An application of information theory to the problem of the scientific experiment. *Synthese*, 140(3), 355-389.

Cacciabue, P. C., Decortis, F., Drozdowicz, B., Masson, M., & Nordvik, J. P. (1992). COSIMO: a cognitive simulation model of human decision making and behavior in accident management of complex plants. *Systems, Man and Cybernetics, IEEE Transactions* 22(5), 1058-1074.

Duffy, V. (2009). *Handbook of digital human modeling: research for applied ergonomics and human factors engineering* (pp. 37-4). CRC Press Taylor & Francis Group.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques*. Elsevier.

House, A., Smith, J., MacKinnon, S., and Veitch, B. (2014). Interactive simulation for training offshore workers. *Oceans'14 MTS/IEEE Conference*, St. John's, NL.

McGaghie, W., Issenberg, S., Barsuk, J., and Wayne, D. (2014). A critical review of simulation-based mastery learning with translational outcomes. *Medical Education*, John Wiley & Sons Ltd, 48(4), 375-385.

Musharraf, M., Smith, J., Khan, F., Veitch, B., & MacKinnon, S. (2016). *A cognitive model of decision making and response during offshore emergency situations*. Manuscript submitted for publication.

Roth, E. M., Woods, D. D., & Pople Jr, H. E. (1992). Cognitive simulation as a tool for cognitive task analysis. *Ergonomics*, 35(10), 1163-1198.

Smith, J., Veitch, B., & MacKinnon, S. (2015). Achieving competence in offshore emergency egress using virtual environment training. *Proceedings of ASME 34th International Conference on Ocean, Offshore and Arctic Engineering OMAE 2015*. St. John's, NL: ASME.

Smith, J. & Veitch, B. (2017). Exploring Virtual Reality and Virtual Environment Technology for Offshore Emergency Egress Training. *SNAME/RINA ONAE Annual Student Technical Conference.* St. John's, NL.

Modelling Workload of a Virtual Driver

Jan-Patrick Osterloh (osterloh@offis.de)
OFFIS Institute for Information Technology, Escherweg 2
26121 Oldenburg, GERMANY

Jochem W. Rieger (jochem.rieger@uol.de)
Carl von Ossietzky University Oldenburg
Applied Neurocognitive Psychology Lab
26111 Oldenburg, GERMANY

Andreas Lüdtke (luedtke@offis.de)
OFFIS Institute for Information Technology, Escherweg 2
26121 Oldenburg, GERMANY

Abstract

In many transportation modes, automation is added to increase comfort, efficiency, or to reduce human errors. Automation has a direct impact on the drivers workload, which can even be higher then without automation. In this paper we propose the development of a virtual driver that can predict human workload in early design phases of automation and assistant systems. We describe the auditory workload model in a closed-loop simulation and an early validation.

Keywords: workload; cognitive modelling; driver model; n-back task

Introduction

In many transportation modes, like in cars, aeroplanes, or on ships, more and more automation is added, with the objective to increase comfort of the passengers, to make transportation more efficient and cheaper, or to reduce human errors. Introducing automation should reduce human workload and consequently human errors, but Metzger and Parasuraman (2005) and others have shown that additional automation can even increase mental workload. They conclude that operators "*should be given an active role in the system to ensure that they can detect and respond to malfunctions in a timely manner*" (Metzger & Parasuraman, 2005, p. 13). This paradigm becomes especially interesting, with the current trend in automotive industry on autonomous driving, where drivers are more and more forced into a monitoring role. In order to allow an evaluation of the workload induced by automation systems on drivers in early design phases, we propose the use of virtual drivers, which predict human behaviour in traffic simulations. Using a virtual driver has many advantages for the automotive industry. First, one can not only use them for evaluations in early design phases, where studies with users are expensive or not even possible, but it also allows to evaluate a lot of different driving scenarios that cannot be covered with driver studies, because it is either to expensive, too time consuming, or to risky.

Our virtual driver[1] is implemented in the cognitive architecture CASCaS (*C*ognitive *A*rchitecture for *S*afety *C*ritical T*a*sk *S*imulation). In the following, we will refer to our virtual driver as "CASCaS driver". In order to allow also prediction of workload, we will extend our cognitive architecture CASCaS with different workload measures. Development of the workload model in CASCaS will be done in iterations, in order to handle the complexity of the workload topic. In a first step, Wortelen, Unni, Rieger, and Lüdtke (2016), described different measures that could be implemented for prediction of workload of different modules in CASCaS, and implemented and validated a first version of a measurement in an open-loop simulation. In this paper, we will describe the second step, the implementation of a closed-loop simulation.

State of the Art

Cognitive Architectures are tools, which provide executable models of human behaviour based on psychological and physiological models of human behaviour. In this paper we will describe the cognitive architecture CASCaS, which has been developed since 2004 (Lüdtke, 2004) in our institute. Main driver for the development of CASCaS was a more application-oriented approach. In contrast to that, many cognitive architectures like ACT-R (Adaptive Control of Thought Rational, (Anderson et al., 2004; Anderson, 2000)), or SOAR (Lehman, Laird, Rosenbloom, et al., 1996) were developed for creation and evaluation of theories and models of human cognition. Beside that, more and more cognitive architectures are now used to predict also pilot or driver behaviour, for example Salvucci (2006) describes a driver model in ACT-R, and Fuller (2010) describes a driver model in QN-MPH. Beside driver modelling, cognitive architectures are also used in aviation, as described in the Human Performance modelling (HPM) element within the System-Wide Accident Prevention Project of the NASA Aviation Safety Program, where they performed a comparison of error prediction capabilities of five cognitive architectures (Foyle & Hooey, 2007), including ACT-R and (Air-)MIDAS (Corker & Smith, 1993; Gore, 2011). CASCaS has been applied in several projects, in order to model perception (Lüdtke & Osterloh, 2009), attention allocation (Wortelen, Lüdtke, & Baumann, 2013), decision making of

[1] Or virtual tester in general, as CASCaS is domain independent

drivers (Weber, Steenken, & Lüdtke, 2013) and human errors of aircraft pilots (Lüdtke, Osterloh, Mioch, Rister, & Looije, 2009) and car drivers (Lüdtke, Weber, Osterloh, & Wortelen, 2009).

There are several model-based approaches to assess the level of cognitive workload in a specific situation. The workload model of McCracken and Aldrich (1984) offers a scale, which assigns workload levels to specific kinds of human actions like "recall, memorize", or "visually inspect". It distinguishes four types of workload: visual, auditory, cognitive and psycho-motor. For example, this model was used to annotate behaviour primitives in cognitive models created with the cognitive architectures MIDAS (Gore, 2011) with associated workload levels. However, the model of McCracken and Aldrich is used in an analytical way and does not assess workload of a human operator online.

In the current work, we outline a model-based approach for the online assessment of workload. For real human operators, online assessment of workload is ongoing research, and typically performed based on *physiological* measures. Physiological measures are quite popular as they can continuously record the operators response without actually intruding into the operators task. The most commonly used physiological measures for workload assessment are electrocardiogram (ECG) and electro-dermal activity (EDA). Previous researches have consistently demonstrated that increased workload levels lead to increased heart rate (HR) and decreased heart rate variability (HRV) (Kramer, 1991). Solovey, Zec, Perez, Reimer, and Mehler (2014) recorded ECG and EDA while driving and were able to discriminate three driving situations with increasing control demand. Brain activation measurements may provide the necessary specificity and state quantification required for online prediction of workload.

Modelling

In the following sections, we will describe our modelling approach, starting with a short introduction to CASCaS and the driver modelling, followed by the workload model implemented in CASCaS.

CASCaS

The cognitive architecture CASCaS (*C*ognitive *A*rchitecture for *S*afety *C*ritical T*a*sk *S*imulation) has been developed since 2004 (Lüdtke, 2004), and has since then been continuously improved and used in several research projects (Lüdtke, Osterloh, et al., 2009; Lüdtke, Weber, et al., 2009; Lüdtke & Osterloh, 2009; Weber et al., 2013; Wortelen et al., 2013). Main focus during the development of CASCaS has been the usage in real-time simulators, mainly car and aircraft simulators, to cover complex scenarios as needed for the industrial application as virtual tester. As many cognitive architectures, CASCaS has several components as depicted in Figure 1, which cover different aspects of human behaviour. Main component of CASCaS is the "Knowledge Processing" component, which is based on Anderson's theory of behaviour levels (Anderson, 2000):

- cognitive layer[2]: decision making in unfamiliar situations
- associative layer: rule-based behaviour and decision making
- autonomous layer: processing without thinking in daily operations, i.e. sensory-motor programs like steering, braking

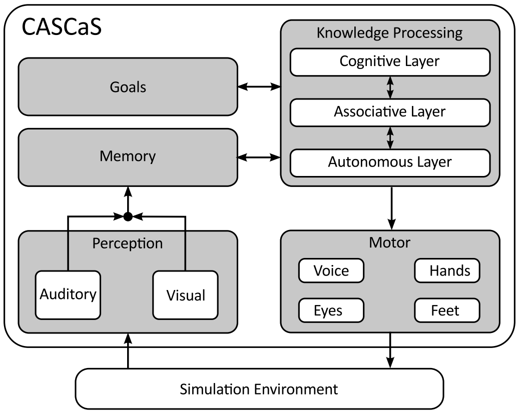


Figure 1: Layered Architecture of CASCaS; (Weber et al., 2013)

For the driver model, only the associative layer and the autonomous layer are used. CASCaS main input is the formal procedure for the associative layer, which describes the interaction with the environment in form of IF-THEN rules. CASCaS procedures are specified in a simple, human-readable, CASCaS-specific text format, allowing also non-computer experts to read, understand, and use the language for modelling without the need to have deep understanding of a programming language. The procedures that are executed by CASCaS are stored in the Memory component, which also contains the declarative memory.

In addition to the Knowledge Processing, additional components for perceptual and motoric processes are part of the architecture, as an interface to the Simulated Environment. The visual component for example, models perception in the focus and in the visual field (Lüdtke & Osterloh, 2009). At each moment, system state and processing of the procedure create the mental model and are expressed as an ordered set of goals and sub-goals that have to be accomplished – the so called goal agenda. Processing of the goal agenda follows these steps:

1. A goal is selected from the goal agenda
2. All rules containing the goal in their Goal-Part are collected, their conditions evaluated by retrieving the needed information from the memory, and organized into a conflict set.

[2] except programming interfaces no model of the cognitive layer is implemented in CASCaS

3. One rule is randomly selected from the conflict set and fired, which means that the motor, percept, and/or memory actions are sent to the motor, percept and memory component respectively, and the sub-goals are added to the goal agenda.

This process is iterated until no more rules are applicable, and all goals are achieved.

Driver Model

The driver model is a combination of a procedure for the associative layer, and some sensory-motor programs on the autonomous layer. The procedure handles decisions that have to be made by the driver, e.g. application of traffic rules, overtaking of other cars, and general interaction with the car and environment. General interaction with the car means operation of possible assistant systems and car interfaces like radio, and GPS by the associative layer. A more detailed description on the driver model that has been used is described in Weber et al. (2013). In our scenario (driving on a German highway), these rules take care of speed limits, and decide if other cars have to be overtaken or to follow them. For this, the traffic is classified onto lanes and positions relative to the ego car, i.e. ahead, behind, left ahead, etc. In addition to that, the distance and speed of the other cars is estimated by the model based on perceived angular sizes.

The motor programs on the autonomous layer cover the actual lateral and longitudinal control, i.e. control of the steering wheel for turns and lane keeping, or control of the pedals for braking and acceleration. For the lateral control, we implemented a simple one point steering control (PD-controller). The longitudinal control has been implemented on the basis of probabilistic models, as described by (Eilers & Möbus, 2014). In general, the probabilistic models are a set of Bayesian Networks, which at each point in time give the probability for a certain output, in our case the braking pedal value and the acceleration pedal value. The probabilities used in the Bayesian Networks are learned from human driver behaviour that has been previously recorded in a highway scenario. Note that the decision to brake, overtake or which speed to drive is made on the associative layer, but the autonomous layer performs the actual motor actions.

Workload in Closed-Loop Simulation

As a first step of the development towards a workload model, Wortelen et al. (2016) implemented a *Working Memory Load* as a mean for workload, which is defined as rate of information elements written to memory, in an open-loop simulation in CASCaS. They tested the working memory load, by using a n-back speed regulation task. N-back tasks have been widely used as a benchmark in the field of neuroscience to influence memory load and task difficulty (Miller, Price, Okun, Montijo, & Bowers, 2009). The n-back speed regulation task requires the driver to follow the speed of the *n*-th speed sign prior to the actual speed sign, as depicted in Figure 2.

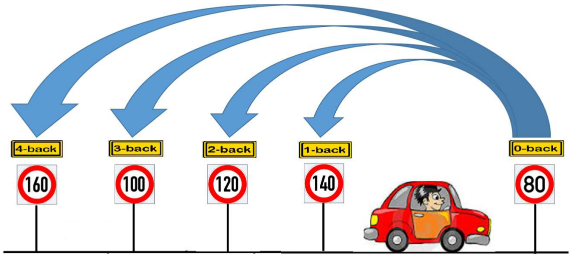


Figure 2: N-back Speed Regulation Task; from (Wortelen et al., 2016)

This approach had two main drawbacks. First, an open-loop simulation has been used, and second the driver model used by Wortelen et al. (2016) was not so sophisticated as the driver model from (Weber et al., 2013). Open-loop simulation in this case meant, that CASCaS was feed with the data from the human drivers, and the steering and acceleration actions of CASCaS are not feed back in a driving simulation. The driver model of Wortelen had therefore only placeholders for the lateral and longitudinal control to mimic multi-tasking. As the objective of CASCaS is to be used as a virtual driver for testing automation and user interfaces in the car, a closed-loop simulation is necessary, i.e. the feedback loop between driver model and driving simulator is closed in a way that the driver model has full control of the simulated car. The closed-loop allows then predictions of the behaviour, without the need of data from real human drivers (with the exception of the data needed for the training of the probabilistic models used for longitudinal control).

The objective of this paper is to describe the integration of Wortelen's workload model in a closed-loop simulation. To achieve a closed-loop simulation, an extended n-back task model from Wortelen et al. (2016) has been integrated with the driver model of Weber et al. (2013).

In a first step, the n-back task procedure for CASCaS has been revised. According to Juvina and Taatgen (2007) humans use two different cognitive control strategies for the n-back task:

1. Phonological rehearsal, i.e. internally rehearsing the list of speeds
2. Time tagging the event

For our model, we have decided to use the phonological rehearsal as strategy for the n-back speed task, as this strategy was, compared to the time tagging strategy, the easiest to implement, due to the lack of a temporal component in CASCaS.

Each time a new speed sign is perceived, the procedure alters a mental list of the speeds. The mental model maintains dedicated associations to the memory chunks at the beginning of the list and it's end, see "list_begin" and "list_end" in Figure 3. When the number of elements is smaller than the current n-back task, the new element is stored into the memory, the "next" association is added from the current "list_begin" to the new element, and then the "list_begin" and "current_rehearsal” associations are moved to the new element. When the number of elements has reached n, the “list_end” association is moved to the “next” chunk to mark the new list end. During the rehearsal, the “current_rehearsal” association is moved over each “next” association from “list_begin” to “list_end”. For each element in the list, an internal speech-action is executed to trigger the phonological loop. Each of this internal speech actions trigger a workload event, which are then accumulated over time as the auditory workload. In our case, we have chosen a seven second interval for the accumulation, because this reflects the response time that is measured with the fNIRS. This is a small difference to Wortelen et al. (2016), as their workload measure captured more than the memory writes from the auditory component, but rather all memory writes from the associative layer.

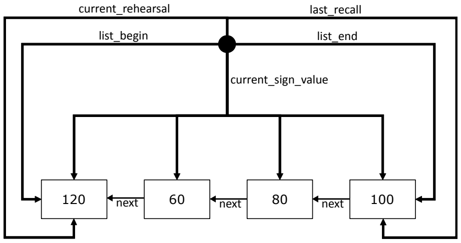


Figure 3: Memory Structure for Rehearsal

Then, as a second step, this procedure has been integrated with the driver model. First, the rehearsal has been added at the appropriate places, i.e. the rehearsal is restricted to phases where the driver is driving ahead, and not overtaking. Second, the sign recognition has been replaced with the one described above, such that the list is maintained, and the “list_end” value is set as the current target speed in the longitudinal control at the autonomous layer. Figure 4 shows a screenshot of CASCaS during the simulation. On the left the visualisation of SILAB, the driving simulator, is depicted, and on the right the workload visualisation of CASCaS. The auditory workload is shown in the diagram in the middle.

Validation

For the validation of the workload measure, we have chosen a two step approach. First, we had an internal model validation as a kind of pre-test. With this step, we make sure, that a) the simulator setup and the scenario is working, b) the data recording is suitable for the planned analysis, and c) the CASCaS driver produces behaviour that is plausible (see below for hypotheses). Second, we will perform another experiment in our driving simulator with human drivers. In this experiment, we will record the driver behaviour as well as physiological data (fNIRS) to measure the workload of the humans.

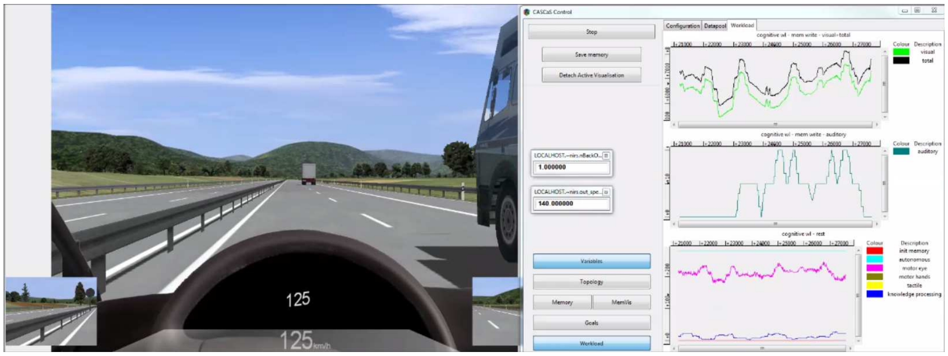
Internal Model Validation

Objective of the model validation is twofold. First this can be seen as a pre-test for the experiment with the humans, where the scenario, the data recording setup, and the data analysis can be tested before the expensive experiment. Second it can be used for model exploration, i.e. checking the plausibility of the CASCaS driver behaviour and improvement of the model subsequently. The plausibility of the model is expressed in multiple hypotheses to be tested before starting the human experiment:

1. The auditory workload can predict the n-back task level.
2. The CASCaS driver will adhere to the correct speed limit according to the n-back level.

As a scenario for the internal model validation, we used the same scenario as described below for the human experiment, but without the traffic.

For hypothesis 1, we calculated Pearson’s correlation *r* between the n-back level and the predicted auditory workload for each of the simulation runs. The mean *r* is calculated with 0.9773, which supports hypothesis 1. In order to analyse hypothesis 2, we analysed how well CASCaS followed the target speed according to the n-back level. For that, in a first step we had to remove the phases where the speed was undefined, because the number of speed signs was lower then the current n-back level. Pearson’s mean correlation *r* between the current speed and the nth target speed of the 15 runs is 0.8445. A more detailed analysis of the speed driven by CASCaS revealed, that in average over all runs, CASCaS



LOCALHOST.~nirs.nBackO...
1.000000

LOCALHOST.~nirs.out_spe...
140.000000

Figure 4: CASCaS workload visualisation while driving; for complete video visit https://hcd.offis.de/wordpress/wp-content/uploads/Workload-02.mp4

made 19 errors of speed out of 169 different speed signs in one run. Error of speed here means, that between two speed limit signs, the actual speed was not in a range of ±10 km/h of the target speed.

Our analysis showed, that when CASCaS had to reduce the speed from a high speed limit (e.g. 140 or 160) to a lower speed limit, the applied braking was not sufficient, such that the new speed limit is not reached before a new sign arrives. An example is shown in Figure 5.

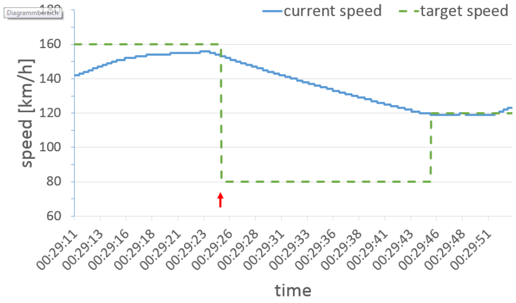


Figure 5: Speed of CASCaS vs. Target Speed

After the target speed of 160 km/h, it goes down to 80 km/h at time 00:29:25. As marked by the red arrow, CASCaS does not decelerate below 90 km/h. The explanation for this is the probabilistic model that is used for the longitudinal control. The probabilistic model has been learned from humans driving in a driving simulator on a normal German highway. In this scenario, subjects never exceeded 130 km/h on the one hand, thus the model has never learned to handle fast driving. In addition to that it can be observed, that the subjects preferred to use the motor break to decelerate, as many people do during normal driving. While this gives a very human-like speed control during normal cruising, the learned probabilistic model does not sufficiently represent driver behaviour for the n-back speed experiment, where active braking and more accurate speed control is required. In future versions of the model, the probabilistic model could be replaced with a mathematical PD controller, to overcome this problem, or re-trained with new experimental data. Beside that, the general driving behaviour, including the overtaking and the rehearsal seems natural, i.e. it shows actual human behaviour.

Comparison with Human Data

For further evaluation, we conducted an experiment with 10 subjects (7 male, 3 female). All subjects where students in an age range between 22 to 42 (mean 27.3 years) with a valid German driving license. Most drivers had more then 10.000 km of total driving experience, 4 had between 5.000 km and 10.000 km, and only one had below 5000 km. During the experiment, subjects where given five different n-back levels, from 0-back to 4-back, and thus five different levels of workload, with 0-back inducing the lowest workload level, and 4-back the highest. Each n-back task lasted around three minutes and consisted of ten different speed changes randomly distributed from 70-140 km/h in steps of 10 km/h. Speed changes where randomly assigned, but we made sure, that speed changes where not larger then 20 km/h at once. We had four repetitions for each n-back task, randomly distributed to avoid sequencing effects. Random distribution of n-back task and speed sequence has been done once for the scenario, and then the same order was re-used for all participants. The scenario had two different traffic situations, half of the scenario had low traffic, the other half had higher traffic. Traffic was always ahead and slow, such that the subjects had to overtake, but no faster traffic was induced from behind. The whole driving experiment lasted for about 60 minutes.

The subjects brain activity was constantly monitored by using a 32 channel neuroNIRX-system (fNIRS). The objective is to use the recorded brain activity data as a source for objective workload measurement, and to correlate it to the CASCaS predictions (similar to (Unni et al., 2015)). The analysis of the brain activity data is still ongoing, nevertheless we analysed the speed driven by the subjects, in order to compare it to CASCaS data on hypothesis 2.

The subjects had a mean r of for their speed of 0.87, and in total over all subjects 17 driving errors where made (mean 1.42 errors per subject). In total, 169 different speed changes had occurred, without initial build-up phase of each n-back task, thus subjects had a error rate of roughly 1%. It could be observed for the subjects, that with the higher n-back levels, also the number of errors increased (1 and 2-back: 2 errors, 3-back 6 errors, 4-back 7 errors). In comparison to that, the model had a mean r for speed adherence of 0.75, and a total of 166 errors (mean 12.7 errors per model run), resulting in an error rate of 7.5%, independent of the n-back task. The decrease in correlation of speed for the model from pre-test to final test can be explained by the added traffic in combination with the probabilistic model for speed control. It can be observed, that the model speeds up for the overtaking (about 10-20 km/h), ignoring also possible speed limits. Especially in high traffic scenarios, overtaking can then take longer then the distance between two speed signs.

Conclusion & Next Steps

Starting from previous work of Wortelen et al. (2016) and Weber et al. (2013), we have integrated a workload model into a closed-loop driving simulation. With that, we extended the workload model to use the phonological loop in CASCaS, so that the auditory workload can predict the n-back level, and thus the workload. For the speed management, a replacement or re-training of the probabilistic model should further improve the model by reducing speed errors.

In addition to that, there are a lot of different workload measures we can implement for the different components in CASCaS, as already introduced by Wortelen et al. (2016). We plan to successively implement more of these measures and validate them against the simulator data, to see if other models can also be used as predictor for the n-back level, which serves as controllable workload indicator.

Acknowledgements

The authors would like to thank their colleagues Bertram Wortelen for providing the initial workload model, and Lars Weber and Mark Eilers for providing the driver model.

This work was supported by the funding initiative *Niedersächsisches Vorab* of the Volkswagen Foundation and the Ministry of Science and Culture of Lower Saxony as a part of the Interdisciplinary Research Centre on Critical Systems Engineering for Socio-Technical Systems 2 (CSE 2).

References

Anderson, J. R. (2000). *Learning and memory: An Integrated Approach* (2nd ed ed.) [Book; Book/Illustrated]. New York; Chichester: Wiley.

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.

Corker, K., & Smith, B. (1993). An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Automation. In *9th Computing in Aerospace Conference* (p. 4660ff). San Diego, CA, U.S.A.: AIAA.

Eilers, M., & Möbus, C. (2014). Discriminative Learning of Relevant Percepts for a Bayesian Autonomous Driver Model. In *Proceedings of the Sixth International Conference on Advanced Cognitive Technologies and Applications, COGNITIVE 2014* (p. 19-25). Venice, Italy: IARIA.

Foyle, D. C., & Hooey, B. L. (2007). *Human Performance Modeling in Aviation*. Boca Raton: CRC press.

Fuller, H. J. (2010). *The virtual driver: Integrating physical and cognitive human models to simulate driving with a secondary in-vehicle task*. Unpublished doctoral dissertation, The University of Michigan.

Gore, B. F. (2011). Workload as a Performance Shaping Factor in MIDAS v5. In *20th Behavior Representation in Modeling and Simulation (BRIMS) Conference.* Sundance, Utah: BRiMS Committee - Curran Associates, Inc.

Juvina, I., & Taatgen, N. A. (2007). Modeling control strategies in the N-Back task. *8th International Conference on Cognitive Modeling*, 73–78.

Kramer, A. F. (1991). Physiological Metrics of Mental Workload: A Review of Recent Progress. In D. L. Damos (Ed.), *Multiple Task Performance* (p. 279-328). London: Taylor & Francis.

Lehman, J. F., Laird, J. E., Rosenbloom, P., et al. (1996). A gentle introduction to Soar, an architecture for human cognition. *Invitation to Cognitive Science*, *4*, 212–249.

Lüdtke, A. (2004). *Kognitive Analyse Formaler Sicherheitskritischer Steuerungssysteme auf Basiseines integrierten Mensch-Maschine-Modells*. Heidelberg, Germany: Akademische Verlagsgesellschaft Aka GmbH.

Lüdtke, A., & Osterloh, J.-P. (2009). Simulating Perceptive Processes of Pilots to Support System Design. In *Proceedings of the INTERACT 2009.* Uppsala, Sweden: Springer LNCS.

Lüdtke, A., Osterloh, J.-P., Mioch, T., Rister, F., & Looije, R. (2009, 10). Cognitive Modelling of Pilot Errors and Error Recovery in Flight Management Tasks. In M. W. Jean Vanderdonckt Philippe Palanque (Ed.), *Proceedings of the 7th Working Conference on Human Error, Safety and Systems Development Systems Development (HESSD)* (1st ed., p. 54-67). Heidelberg: Springer.

Lüdtke, A., Weber, L., Osterloh, J.-P., & Wortelen, B. (2009, 10). Modeling Pilot and Driver Behaviour for Human Error Simulation. In J. Jacko et al. (Eds.), *HCI International 2009* (1st ed.). Heidelberg: Springer.

McCracken, J. H., & Aldrich, T. B. (1984). *Analyses of selected LHX mission fucntions: Implications for operator workload and system automation goals* (Tech. Rep. No. ASI479-024-84). Fort Rucker, AL, USA: U.S. Army Research Institute Aviation Research and Development Activity.

Metzger, U., & Parasuraman, R. (2005). Automation in Future Air Traffic Management: Effects of Decision Aid Reliability on Controller Performance and Mental Workload. *Human Factors*, *47*(1), 35–49.

Miller, K., Price, C., Okun, M., Montijo, H., & Bowers, D. (2009). Is the N-Back Task a Valid Neuropsychological Measure for Assessing Working Memory? *Archives of Clinical Neuropsychology*, *24*(7), 711–717.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *48*(2), 362–380.

Solovey, E. T., Zec, M., Perez, E. A. G., Reimer, B., & Mehler, B. (2014). Classifying driver workload using physiological and driving performance data: two field studies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI14)* (p. 4057-4066). New York, NY, USA: ACM.

Unni, A., Ihme, K., Surm, H., Weber, L., Lüdtke, A., Nicklas, D., et al. (2015). Brain activity measured with fNIRS for the prediction of cognitive workload. In *Proceedings of 6th IEEE Conference on Cognitive Infocommunications (CogInfocom 15)* (p. 349-354). Gyor, Hungary: IEEE Press.

Weber, L., Steenken, R., & Lüdtke, A. (2013, 10). Integrated Modeling for Safe Transportation (IMoST 2) - Driver Modeling & Simulation. In T. Jürgensohn & H. Kolrep (Eds.), *Fortschritt-Berichte VDI* (p. 26-41). Berlin: VDI.

Wortelen, B., Lüdtke, A., & Baumann, M. (2013, 03). Integrated Simulation of Attention Distribution and Driving Behavior. In *Proceedings of the 22nd Annual Conference on Behavior Representation in Modeling and Simulation.* Ottawa, Ontario, Canada: BRiMS Committee.

Wortelen, B., Unni, A., Rieger, J. W., & Lüdtke, A. (2016, 10). Towards the Integration and Evaluation of Online Workload Measures in a Cognitive Architecture. In P. Baranyi (Ed.), *Proceedings of 7th IEEE Conference on Cognitive InfoCommunications* (p. 11-16). Wroclaw, Poland: IEEE.

Comparing the Input Validity of Model-based Visual Attention Predictions based on presenting Exemplary Situations either as Videos or Static Images

Bertram Wortelen (bertram.wortelen1@uni-oldenburg.de)
Cognitive Psychology, C.v.O. University Oldenburg, Ammerländer Heerstr. 69
Oldenburg Germany

Sebastian Feuerstack (feuerstack@offis.de)
Human Centered Design, OFFIS – Institute for Information Technology, Escherweg 2
26121 Oldenburg Germany

Abstract

Functional cognitive models are used to explain observed human behavior. Applying such models to predict behavior requires generalization of the model to be applied in different application domains but also a careful consideration of model input data validity. Visual attention models have already been validated in various domains. But elicitation techniques to collect valid input data that is reproducible by others are still missing. For visual attention prediction model input data is determined mainly based on discussion between experts and individual experience, which is difficult to reproduce. We use a software tool to support input validity. The tool helps users to create attention models. It uses images of the situations that are investigated for stimulating the users to virtually put themselves into these situations. An experiment (n=40) showed that using looping videos instead of static images stimulates imagination in a different way. It has an effect on the models generated by the users and needs careful consideration.

Keywords: Visual Attention; Human Factors; Supervisory Control; Software-supported method; Cognitive Modeling; Safety.

Introduction

Applying cognitive models for predicting human behavior often requires on the one hand expertise in cognitive modeling and on the other hand profound knowledge of the domain for that they are applied. To tackle the former, modeling tools, such as for instance CogTool have been proposed that are based on “zero-parameter models” (i.e. GOMs, KLM and ACT-Simple) (John et al. 2004) that enable human performance prediction based on automatically generated cognitive models. Such approaches can be applied by e.g. a designer to predict human performance for human-machine interface (HMI) design variants.

Most of the cognitive models cannot be generalized and reduced to a predefined fixed set of operators as they depend on experts’ inputs to determine valid input parameters. Visual attention prediction models depend for instance on parameters that are knowledge driven and therefore require application domain knowledge for determining the parameter values. The current process for parameter estimation is to ask domain experts and letting them argue and discuss about the parameter values. And in fact studies in various application domains report high correlations to measured data following such approaches (Wickens et al. 2008, Koh et al. 2011). But a discussion-based parameter-determination is hardly reproducible as the quality depends on individual expertise and on the composition of the expert group (to ensure e.g. that silent voices are also heard).

We use a software tool, the Human Efficiency Evaluator (HEE) to capture the relevant knowledge for visual attention prediction in the specific application domain. The tool implements a structured, repeatable process and is used by the experts individually to capture and aggregate their knowledge. To stimulate the knowledge capturing the tool depends on either images or videos that show exemplary situations for that the model parameters are then estimated by the experts.

In this contribution we explore the experts’ capabilities to abstract from the very specific concrete shown situations and focus on identifying the differences between a video-based and an image-based stimulus for attention modeling.

Model-based Visual Attention Prediction

Model-based visual attention prediction can complement eye-tracking studies, as it does not depend on HMI functional prototypes and simulations but on human experiences and imagination that is captured by discussion and feed as parameters in prediction models. The SEEV model of attention allocation (Wickens et al. 2001) is such a promising model of visual attention. It describes that “the allocation of attention in dynamic environments is driven by bottom up attention capture of *salient* events, which are inhibited by the *effort* required to move attention, and also driven by the *expectancy* of seeing *valuable* events” (McCarley et al. 2002). The SEEV model is used to predict the percentage of time, that someone spends looking at an area of interest (AOI). It is typically applied by HF experts that have a deep understanding of human attentional processes. The SEEV model relates the probability P_s of attending a specific AOI s to four factors:

$$P_s = \underline{S}aliency - \underline{E}ffort + \underline{E}xpectancy \cdot Task\ \underline{V}alue$$

The first two coefficients, *Saliency* and *Effort* are bottom-up factors that describe the saliency of information displayed by an AOI and the effort it takes to obtain the information, e.g., by moving eyes and head or navigating through a menu. *Expectancy* and *Task Value* are top-down factors. They describe how often new information can be expected from an AOI and how valuable the information is for accomplishing the tasks of the human operator.

While the bottom up parameters can be estimated e.g. based on physiological data about the effort for eye and head movements (Gore et al. 2009) or by computing saliency maps (Itti & Koch 2001), the determination of the knowledge-based expectancy and value coefficients often depend on data gained by domain experts for a specific application use case.

SEEV model variants, considering some or all of the four factors, have been used to model and predict attention allocations for a wide variety of tasks in various domains: For instance in aeronautics, to predict monitoring while taxiing on ground (Wickens et al. 2008), or the influence of specific cockpit instruments (Goodman et al. 2003) on monitoring behavior. In the automotive domain the model was applied to evaluate drivers' monitoring behavior while approaching intersections (Bos et al. 2015) and also to evaluate the influence of secondary tasks (Wortelen et al. 2013). Recent studies also demonstrate modeling efforts ending with valid predictions for nurses' experience level when assisting in an operation theater in a hospital (Koh et al. 2011). All SEEV model related studies we are aware of, report moderate up to very high correlations (0.6< R <0.97) between eye tracking studies and the model predictions.

Improving Input Quality

The broad majority of the studies above applied the "least integer ordinal value" heuristic, which estimates parameter values by letting experts systematically compare AOIs between conditions. A recent approach applies the analytic hierarchy process technique for quantifying the informational importance (Ha & Seong 2014).

The results of those methods, the relevant concrete parameter values are stated in most of the studies above and predictions therefore can be reproduced, but only one study we found (Koh et al. 2011) reported insights about the amount of experts, their background and prior knowledge, and the method applied to agree on the model input parameter values. If the attention model is created for instance by only one HF expert, errors made by this HF expert can have a huge impact on the predictions. If the parameter estimation is a result of a discussion of several experts, quiet voices can be missed easily. Finally, if instead several experts are individually applying a method, the often observed evaluator effect might become evident (Hertzum & Jacobsen 2001).

We use a software tool, the Human Efficiency Evaluator (HEE), for input data gathering. We believe that using a well-structured and tool supported process for input data gathering improves documentation and reproducibility of the input data gathering. Prior studies have shown that the tool can be applied in parallel sessions and with very little training by domain experts for visual attention modeling (Feuerstack & Wortelen 2016). Based on a preset set of operator tasks to consider and images of HMI design variants embedded in their environment, the tool guides the domain experts through four major steps: (1) the identification of areas of interest (AOI) relevant for the operator tasks (see Figure 1 for a screenshot), (2) the determination of expectancy, which is performed by ordering the AOIs according to the expected frequency of information events, (3) ordering the importance of the operator tasks, and finally, (4) the specification of relevance of each AOI for the operator tasks. The least integer ordinal value heuristic is used to calculate numeric parameters from the orders defined in step (3) and (4). In (Feuerstack & Wortelen 2017) we observed a high variance in the data we collected from the domain experts, and interestingly also from the HF experts that we evaluated in a separate session. While variance between experts was also observed in earlier studies e.g. in usability evaluation (Hertzum & Jacobsen 2001) it has not been considered to be relevant for model-based attention prediction to the best of our knowledge. The observed variance seems to be capturing well the diversity that people show in general when asked to give estimates. First studies indicate (Feuerstack & Wortelen 2016, Feuerstack & Wortelen 2017) that the diversity prediction theorem (also called Wisdom of the Crowd (Surowiecki 2004)) can be applied also for attention prediction modeling with the HEE: By averaging individual model predictions, individual prediction errors can be eliminated and high correlations with measured eye-tracking data have been observed (Feuerstack & Wortelen 2017).

To gather expert data the tool requires images representing a situation (e.g. a critical traffic situation) for that the operators' (i.e. drivers') visual attention distribution is then modeled. The approach depends on such images to stimulate the capability of the experts to mentally put themselves into this concrete situation (e.g. one specific left lane change situation) and to anticipate all possible situations that could occur (while performing a lane change). While looking at data from a previous study (Feuerstack & Wortelen 2017), we suspected that the models created by the subjects might be affected by the images that were selected to be representative for a specific situation. Therefore, we investigate how the selection of images representing situations for visual attention modeling impacts the identification of AOIs (i.e. where one looks at) by the subjects and how using videos instead of images might reduce potential biases. For an experiment we formulate the following hypotheses:

H_1: *"Experts mark bigger AOIs for information that is moving relative to the position of the human operator if videos are used to present a driving situation in several variations compared to using static images."*

The location of information that is not fixed relative to the operator is moving in a video, while it has a fixed position in an image. Therefore it is hypothesized that participants only mark boundaries of information at a single position when using images instead of marking larger areas when using videos.

H_2: *“The choice between video and image does not affect the expectancy and value parameters of the SEEV model.”*

Although we assume that using videos to represent situations has an effect on the sizes of AOIs compared to using static images, we see no reason, why it should affect the modelling process for expectancy and value parameters of the SEEV model.

H_3: *“More AOIs are marked using looping videos of a situation compared to static images.”*

In previous studies (Feuerstack & Wortelen 2016, Feuerstack & Wortelen 2017) we found high individual differences in how many and what kind of AOIs were marked. We assume that the static image is a reason for this variance. Some AOIs might not be marked by every subject, because the dynamics of the situation are not visible in the static image. Thus, they might fail to identify all areas were information shows up. In contrast, the video shows the dynamics of the situation. Thus we assume that more AOIs are marked using videos.

Experiment

We conducted an experiment and asked subjects to model the distribution of attention for different phases of an overtaking maneuver using the Human Efficiency Evaluator (HEE). We tested two conditions in a between subject design with two groups of subjects. For some subjects the driving situations were represented using videos (V condition) and for some using static images (I condition).

Participants

40 licensed car-drivers were recruited by public announcements in the university and were required to be licensed for at least 3 years (mean: 8.05 median: 7.0), have a minimum driving experience of 3000 km per year (mean: 11450 median: 8000) and received an expense allowance of 10 EUR/h. 23 women and 17 men participated in the study, aged between 20 - 40 years (avg: 25.175 median: 24).

Procedure

The experiment was carried out in groups of 4 to 8 subjects for each session and was done in a computer lab in that every subject had a separate PC workplace with two screens. In total we had 20 randomly assigned subjects for each group and participants of both groups were mixed within the sessions.

A video-tutorial, a scripted subject introduction and a written exercise sheet have been used to reduce potential bias by the instructors. Subjects were allowed to ask questions, which were transcribed in the observation records. The subjects had to start with watching the tutorial video first, which introduced them to the tool and its implemented attention modelling process by a supervision example of a football game. The tutorial video was identical for both groups, with the only exception that for one group the football situations were displayed as static images and for the other group looping videos of several variances of the same football situation (a corner kick) were used. After the tutorial were introduced to an overtaking scenario consisting of three phases: (1) merging into left lane, (2) overtaking, and (3) merging into right lane. All subjects were asked to identify all areas of interest for each phase that they assume are relevant for three given tasks as a car driver: (1) Respect speed limit, (2) Overtake slower vehicles, and (3) Control lateral position. Figure 1 depicts the main screen of the HEE that the subjects used to identify the areas of interest. In the video condition the videos started automatically after starting the tool but could be paused by the participants.

After the experiment the two authors and a co-worker independently identified classes of AOIs based on the 1155 AOIs marked by all subjects. In a group discussion we agreed on 37 classes of AOIs. Subjects used for their models different levels of abstraction. For example, some marked the entire dashboard as an AOI, while others differentiated between speedometer, revolution counter and the blinking arrow of the direction indicator. We reflected this by organizing the AOI classes in a hierarchy shown in Figure 2. Afterwards each of the three persons independently classified all 1155 AOIs with a substantial level of agreement (Fleiss’ κ = 0.83).

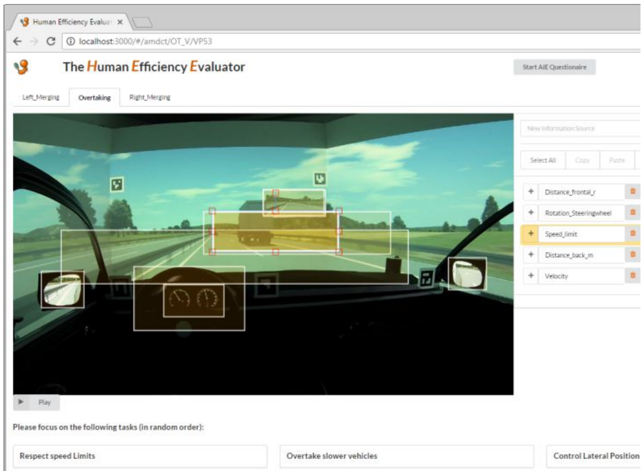
Results and Discussion

Hypothesis H1

To test hypothesis H1 we differentiate between AOIs that have a fixed position relative to the head of the driver (AOI classes with white boxes in Figure 2), and those that move relative to the head of the driver (AOI classes with gray boxes in Figure 2). As expressed in H1, we only expected an effect for moving information sources.

For each AOI class we took all AOIs belonging to the class, including subclasses and calculated the mean size of the information sources in square pixels.

We did this separately for each condition V and I. The results are plotted in Figure 3. The red line is a straight line



Please focus on the following tasks (in random order):

Respect speed Limits

Overtake slower vehicles

Control Lateral Position

Figure 1: Areas of Interest (AOIs) identification with the HEE.

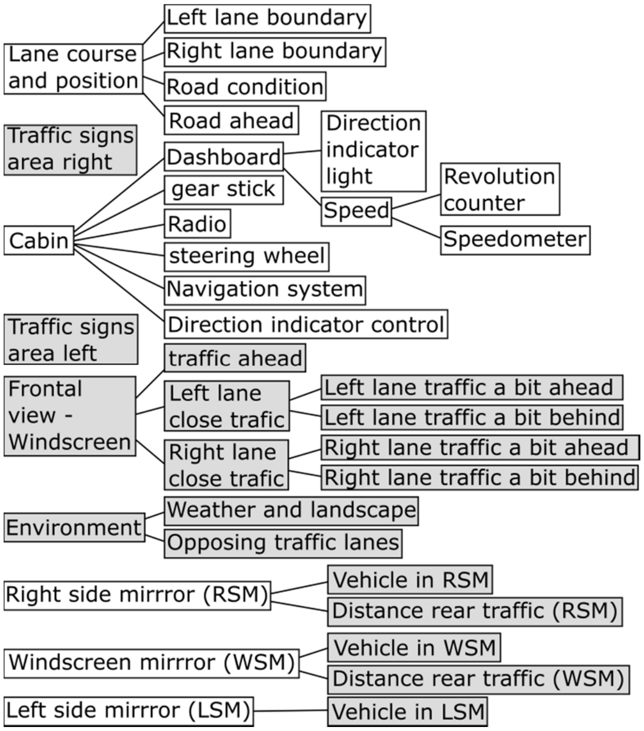


Figure 2: Hierarchy of AOI classes, showing different levels of abstraction.

through the origin with slope 1. For each AOI class the mean AOI size in the V condition is plotted against the mean AOI size in the I condition. According to the H2, AOIs with a fixed position should have similar sizes in both conditions and thus should be located close to the red line (green data points in Figure 3), while moving AOIs should be plotted above the line (blue data points). The sample sizes for all the AOI classes differ, because some AOI classes were marked very often by participants, while others are rarely marked. The size of data points in Figure 3 is proportional to the minimum of the sample size of the V and I conditions (n_{Min}). The Figure seems to support our hypothesis. The two green outliers at (20K, 60K) have a sample size of 1.

For all moving AOI classes with $n_{Min} > 10$, we did Welch two sample t tests with unbalanced sample sizes, to test if the differences are significant. Table 1 shows the p values after Holm-Bonferroni correction for 8 AOI classes with $n_{Min} > 10$. In 4 of the six classes results are significant.

Table 1: Results of the t-tests for differences in AOI sizes between I and V condition for AOI classes with more than 10 AOIs in each condition.

AOI class	p
Frontal view	0.001
Traffic ahead	0.085
Left lane close traffic	0.116
Right lane close traffic	0.003
Right lane traffic a bit ahead	0.011
Traffic signs area right	0.001

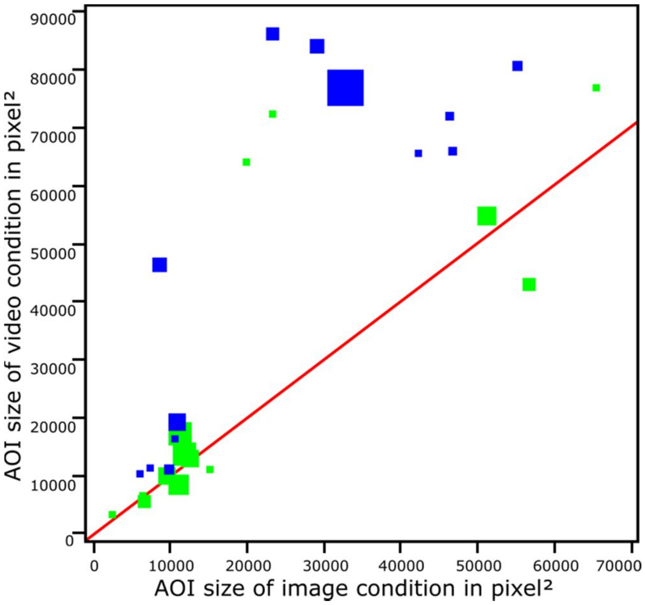


Figure 3. Comparison of average AOI sizes between V and I conditions for each AOI class. AOI classes with moving information are represented with blue data points, fixed AOIs with green data points. Size of data points proportional to sample size. Red line is line through the origin with gradient 1. Blue data points above the line indicate that moving information is marked bigger in the video condition.

Figure 4 illustrates the effect using the AOI class "*Traffic signs area right*" as an example. The top row shows the AOIs marked by subjects of the image group. It mostly shows small areas that just cover the traffic sign in the image, while the areas in the bottom row are from the video condition, were subjects marked the entire region where the traffic sign could be visible. It can also be seen, that participants only mark the information, if it is visible. In the second phase (overtaking) no traffic sign was visible in the image. In this phase only one subject created an AOI in the area were traffic signs are typically perceived.

Hypothesis H2

For testing H2, that there is only an effect on the sizes of AOIs but not on the parameters of the SEEV model, we conducted equivalence tests for these parameters. We used the two one-sided test (TOST) procedure (Schuirmann 1987) to test for equivalence of the parameters between the image and video conditions. For the procedure a margin δ for the difference of the means of the parameters between V and I conditions needs to be defined ($-\delta < \overline{M_V} - \overline{M_I} < \delta$), for which we consider the parameters as equal. [-δ, δ] is the equivalence interval.

Parameters were operationalized using the lowest ordinal heuristic (Wickens et al. 2001). Therefore, the minimum difference between parameters from one modeler is 1. We chose to express the margin δ for the mean of a SEEV parameter for a specific AOI class as a fraction of this minimal individual difference and consider the parameter distribu-

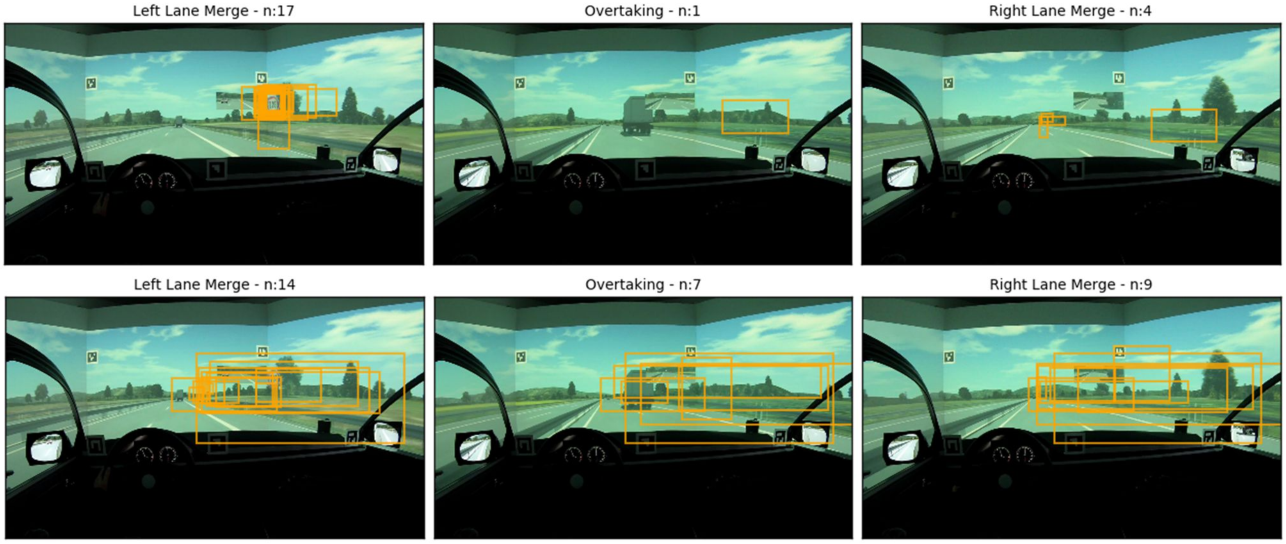


Figure 4: Subjects identification of traffic signs based on static images (top row) and on looping videos (bottom row)

tions equivalent, if the means do not differ more than half of the minimal individual difference ($\delta = 0.5$). The TOST shows equivalence for a chosen α-level, if the (1-2α)-confidence interval is within the equivalence interval. For each AOI class we tested it separately for all three situations, because parameters differ between situations. However, in several cases this resulted in very few data points for an AOI. We excluded AOIs with less than 6 data points.

In Figure 5 all remaining AOIs are listed on the x-axis ordered by the size of the confidence interval. It shows the confidence intervals as red bars and the equivalence interval as blue area. It is easy to see, that we were not able to show parameter equivalence for even a single AOI. For most AOIs the difference of the means is well within the equivalence interval, but the boundaries of the confidence intervals are not. Because we did this test separately for each AOI and each driving situation, the limited number of data points resulted in large confidence intervals and prevents drawing a clear conclusion.

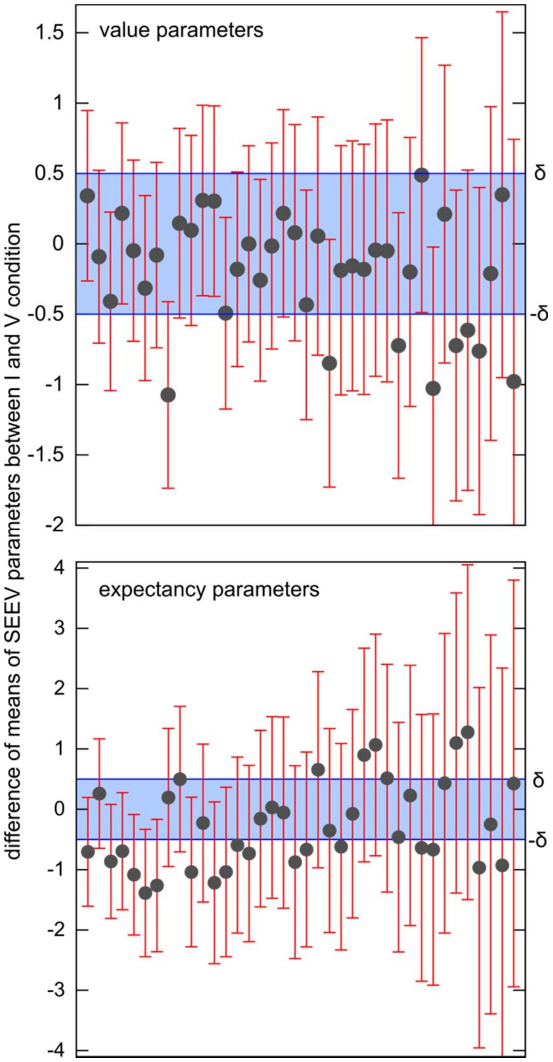


Figure 5. Visualization of the TOST results. Red bars are the 90% confidence intervals. Blue area is the equivalence interval.

Hypothesis H3:

For each participant the identified AOIs for each driving phase were counted resulting in 60=20 × 3 counts. An independent-samples t-test was conducted to compare the counts between video and image condition. There was not a significant difference in the numbers of identified information sources for video (M=6.53, SD=2.13) and image (M=6.23, SD=2.70) conditions (t_{118}=0.68, p=0.50). Subsequent t-tests for each situation alone also found no significant effect.

This result was unexpected. We examined the data in more detail. As we expected, information that is not visible in the image, but is sometimes visible in the video (e.g., indicator lights or road signs) is marked more often in the video compared to the image condition. The opposite case did not occur (information visible in the video, but not in the image). However, we identified another group of AOIs that are visible in the image but only sometimes in the video. This group produces the opposite effect. These AOIs were marked more often in the image compared to the video condition. A t-test showed, that the difference in the effect was significant (p=0.03) between both groups of AOIs 1: visible in image, but only sometimes in the video, and 2: not visible in image, but at least sometimes visible in video.

Although, this is just a post-hoc hypothesis, it indicates, that the choice between using videos or images and selecting what exactly is displayed has an effect on the models generated by the users. It therefore needs careful consideration.

Conclusion

Exemplary situations to stimulate monitoring behavior modeling need to be carefully chosen. One has to distinguish between areas of information with fixed visual borders (e.g. side mirrors), areas without fixed visual borders but fixed location (e.g. road ahead) from the monitoring person's perspective, and those with moving location (e.g. traffic signs). Specifically the AOI identification of AOIs with moving locations benefits from using videos instead of images.

Acknowledgments

The authors acknowledge the financial support by the European Commission (H2020-MG-2014-2015) in the interest of the project AutoMate – GA 690705 and the funding initiative Niedersächsisches Vorab of the Volkswagen Foundation and the Ministry of Science and Culture of Lower Saxony as a part of the Interdisciplinary Research Centre on Critical Systems Engineering for Socio-Technical Systems.

References

Bos, A. J., Ruscio, D., Cassavaugh, N. D., Lach, J., Gunaratne, P. & Backs, R. W. (2015), Comparison of novice and experienced drivers using the SEEV model to predict attention allocation at intersections during simulated driving, in *Proceedings of the Eighth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*.

Feuerstack, S. & Wortelen, B. (2016), AM-DCT: A visual attention modeling data capturing tool for investigating users' interface monitoring behavior, in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, ACM, New York, USA, pp. 252–255. http://doi.acm.org/10.1145/2909132.2909276

Feuerstack, S. & Wortelen, B. (2017), A model-driven tool for getting insights into car drivers' monitoring behavior, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV'17)* (in press).

Goodman, A., L.Hooey, B., Foyle, D. C. & Wilson, J. R. (2003), Characterizing visual performance during approach and landing with and without a synthetic vision display: A part task study., in D. C. Foyle, A. Goodman & B. L. Hooey, eds, *Proceedings of the 2003 Conference on Human Performance Modeling of Approach and Landing with Augmented Displays*, number NASA/CP-2003-212267, Moffett Field, CA: NASA, pp. 71–89.

Gore, B. F., Hooey, B. L., Wickens, C. D. & Scott-Nash, S. (2009), A computational implementation of a human attention guiding mechanism in MIDAS v5, in V. G. Duffy, ed., *Digital Human Modeling*, Vol. 5620/2009 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 237–246.

Ha, J. S. & Seong, P. H. (2014), 'Experimental investigation between attentional-resource effectiveness and perception and diagnosis in nuclear power plants', *Nuclear Engineering and Design* **278**, 758–772.

Hertzum, M. & Jacobsen, N. E. (2001), 'The evaluator effect: A chilling fact about usability evaluation methods', *Int. J. Hum. Comput. Interaction* **13**(4), 421–443. http://dx.doi.org/10.1207/S15327590IJHC1304_05

Itti, L. & Koch, C. (2001), 'Computational modelling of visual attention', *Nature Reviews Neuroscience* **2**(3), 194–203.

John, B. E., Prevas, K., Salvucci, D. D. & Koedinger, K. (2004), Predictive human performance modeling made easy, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, ACM, New York, NY, USA, pp. 455–462. http://doi.acm.org/10.1145/985692.985750

Koh, R. Y. I., Park, T., Wickens, C. D., Ong, L. T. & Chia, S. N. (2011), 'Differences in attentional strategies by novice and experienced operating theatre scrub nurses', *Journal of Experimental Psychology: Applied* **17**(3), 233–246.

McCarley, J. S., Wickens, C. D., Goh, J. & Horrey, W. J. (2002), A computational model of attention/situation awareness, in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 46, SAGE Publications, pp. 1669–1673.

Schuirmann, D. J. (1987), 'A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability', *Journal of Pharmacokinetics and Biopharmaceutics* **15**(6), 657–680. http://dx.doi.org/10.1007/BF01068419

Surowiecki, J. (2004), *The Wisdom of Crowds*, Doubleday.

Wickens, C. D., Helleberg, J., Goh, J., Xu, X. & Horrey, W. J. (2001), Pilot task management: Testing an attentional expected value model of visual scanning, Technical report, NASA Ames Research Center Moffett Field, CA.

Wickens, C. D., McCarley, J. S., Alexander, A. L., Thomas, L. C., Ambinder, M. & Zheng, S. (2008), *Attention-Situation Awareness (A-SA) Model of Pilot Error*, CRC Press/Taylor & Francis Group, chapter 9, pp. 213–239.

Wortelen, B., Baumann, M. & Lüdtke, A. (2013), 'Dynamic simulation and prediction of drivers' attention distribution', *Transportation research part F: traffic psychology and behaviour* **21**, 278–294.

Modeling of Visual Search and Influence of Item Similarity

Stefan Lindner (stefan.lindner@campus.tu-berlin.de), Nele Russwinkel (nele.russwinkel@tu-berlin.de), Lennart Arlt, Max Neufeld, Lukas Schattenhofer
Department of Cognitive Modeling in dynamic Human-Machine Systems, TU Berlin, Marchstr. 23
10587 Berlin, Germany

Abstract

A modeling approach addressing visual search in an array of items of differing similarity is introduced. The model is able to capture the effects found in a study that varies target-distractor similarity (low vs. high), distractor-distractor similarity (low vs. high) of icons, target presence (present vs. absent) and the set size (8, 16 or 24 icons). To be able to simulate human visual search in such a task with original ACT-R mechanisms we implemented a hybrid search strategy that combines parallel and serial search. The presented model can provide useful insight for researchers interested in modeling tasks containing visual icon search.

Keywords: visual search; similarity; ACT-R; cognitive modeling.

Introduction

Visual search is a general requirement for everyday tasks. Especially for user interfaces it is crucial to find the right icon/button/menu item quickly to proceed with the task and to reach the actual goal. The challenge is to find the target item amongst several, often similar distractor items. Performance in such tasks changes with the number of items on screen. Two search paradigms are known, determining whether the number of items influences search time or not. In the case that the target is similar to other items, search time typically increases roughly linearly with the set size (e.g. Wolfe, 1994). Here **serial search** takes place because the person has to actively attend one item after the other in a serial manner.

In case the searched item is distinctive from the other items (a yellow item between blue items) the subjective feeling is that the item literally pops out from its surroundings. Here, reaction time will not differ too much between set sizes – a phenomenon called the "pop-out effect". This **parallel search** relies on preattentive processes that take place before attention is actively drawn to specific items. Whenever a single visual basic feature (such as color or form) differentiates the target from other items this quick process can occur.

The interesting case is the overlap between those two pure paradigms, whenever a heterogeneous field of items has to be searched.

Our aim is on the one hand to understand how people cope with such search demands and what kind of strategies they use. On the other hand we want to model such search behavior to be able to predict the usability and search time of interfaces.

The cognitive architecture ACT-R (Anderson et al., 2004, Anderson, 2007) offers a visual module that is able to address both search paradigms and also a module for motor output to enable realistic predictions about reaction times in visual search tasks. The visual module has two subsystems, the **where system** and **what system**. The where system simulates preattentive processes and relies on well accepted theoretical concepts (Wolfe, 1994; Treisman & Gelade, 1980). Each visual item has features such as type (text, or oval for a button or others), color or width. It is possible to search for items with a specific feature. As a response to such a search request a visual location of such an item is returned. In the next step the visual attention can be directed to this location. The first process needs no time, the second process does need time. A shift of visual attention takes 135ms - 50ms for the production to fire that elicits the request of the shift and 85ms for the shift itself.

But how is visual search executed that is neither purely serial nor parallel in nature? Do people use strategies to find their target item quicker within larger distractor sets, and does an inhomogeneous distractor set regarding similar features (e.g. Duncan & Humphreys, 1989) further influence visual search apart from the above mentioned mechanisms?

The main goal of the paper is to explore the possibilities of accurately modeling visual search in environments with objects of differing similarity in the cognitive architecture ACT-R.

A number of ACT-R models exist that address visual search with different variations (Fleetwood & Byrne, 2006; Everett & Byrne, 2004). Fleetwood and Byrne manipulated set size and quality of icons in a computer-based target identification task. Icon quality was realized by the level of distinctiveness and complexity of icons. Good quality icons were easily distinguishable from others (on a preattentive level). Evidence in the eye tracking data showed that users were able to preattentively discriminate subsets of visual objects in conjunction search tasks, but here the number of similar items were held constant. Fleetwood and Byrne built two ACT-R models to simulate experimental results and managed to achieve a good fit.

There are also a number of ACT-R modules that aim at a more fine-grained modeling of certain aspects of visual cognition. The EMMA-module (Eye Movements and Movements of Attention; Salvucci, 2001) attempts to better model the intricate relationship between eye movements and

The cognitive processes that closely interact with them, while the PAAV module (Nyamsuren & Taatgen, 2012) allows for the incorporation of bottom-up processes. Our model, however, does not make use of any specific bottom up-processes of visual search. Our rational for that is two-fold. On the one hand, owing to the specific structure of the experiment, top-down search of the target item is generally encouraged and then reinforced through practice.

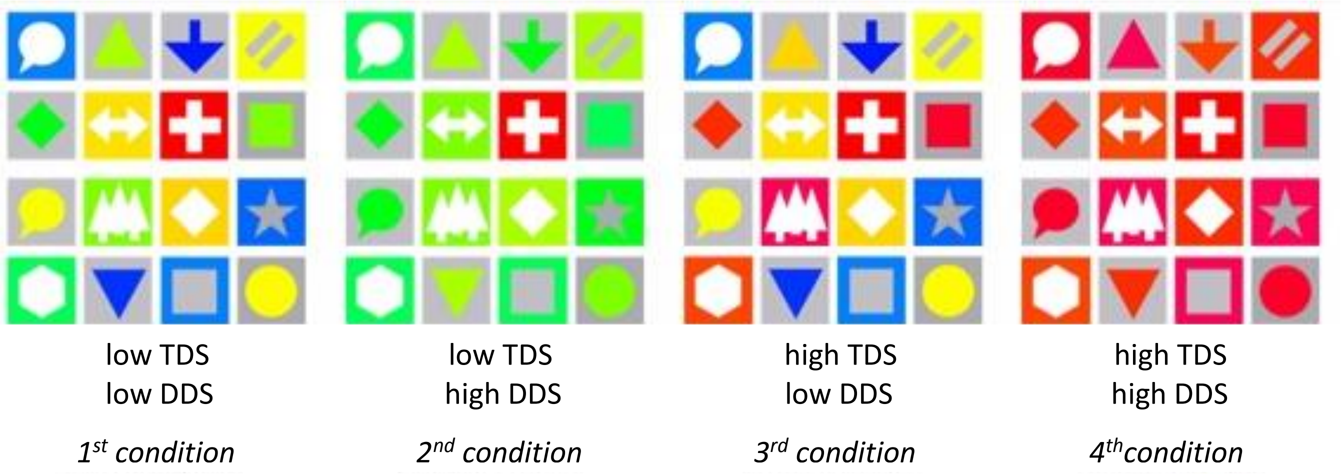


Figure 1: Experimental similarity conditions according to color. Target is the white cross on red ground. (for demonstration; not original icons used)

On the other hand, more importantly even, we are interested in the possibility to model visual search with the core ACT-R mechanisms. While a very fine grained modeling of visual processes has its place, for most task models - especially if they are not primarily focused on the visual aspect of the task – it may be much more realistic and efficient to use a simple model that captures the general behavior reasonably well.

To that end we took an experiment conducted by Trapp & Wienrich (2017) that looks at visual item search independence of four factors: Target-Distractor similarity (TDS), similarity between distractors (DDS), the presence of a target (target presence) and the overall amount of icons present (set size). The experiment is particularly well suited for modeling attempts. It demands the active consideration of both the absolute and relative properties of visual icons such as location, color and form - and therefore tests ACT-R's modeling capabilities in all of these areas, as well. The variation of set sizes also allows for the isolation of invariable mechanisms and those that are dependent on the size of the visual search area.

After presenting the original experiment and its main findings the modeling approach will be introduced. We will first describe the basic model in ACT-R and then move into specific modifications that allowed the final model to capture the experimental results well. To be maximally instructive to future modelers of similar visual mechanisms, we will also shortly discuss several modeling dead ends.

Experiment

The two main independent factors in the experiment by Trapp & Wienrich were Target-Distractor similarity (TDS; low vs. high) and Distractor-Distractor similarity (low vs. high) (see Figure 1). Similarity was realized by the color of the icons. Two further independent factors, target presence (present vs. absent) and the set size (8, 16 or 24 icons) were completely crossed with the similarities, resulting in a 2x2x2x3-factorial setup and a total of 24 experimental conditions. Each participant conducted 12 trials of each condition (for a total of 288 trials per participant), constantly switching between conditions in a fixed blocked fashion. The participants performed a visual search task on a 10" mobile touch device, in which they had to find a specific target icon within a set of distracting icons.

Each trial was performed in the following manner: After the target icon was shown for two seconds, a fixation cross was presented in the center of the screen to ensure a standardized gaze point for all participants. After the fixation cross disappeared, a set of icons was shown. When the target icon was present in the set, the participants had to find and select the target icon as fast as possible. Whenever there was no target, they had to select a specific button at the bottom of the screen to indicate the absence of the target icon. Subsequently, they received feedback on whether their answer was true or false. The reaction time was recorded for each trial and served as a performance measurement. The experiment comprised 18 participants in total (11 male and 7 female) aged between 18 to 30 years.

Both main and interaction effects of TDS, DDS, set size and target presence were consistent with the experimenters' predictions and previous findings. Their main findings were as follows (see also figure 3):

1) The first two conditions (both low TDS) produced low reaction times that showed only a very slight increase with set size.

2) The third condition (high TDS and low DDS) produced moderate reaction times and increased with set size.

3) The fourth condition (high TDS and high DDS) produced high reaction times that increased strongly with set size.

4) The absence of the target item increased reaction times only slightly and by a constant term in the first two conditions. In the third and fourth condition the difference strongly increased with set size.

Model

In order to capture these effects first a basic model in ACT-R was created in a way that required the fewest assumptions while still being able to successfully solve the task in all conditions. Instead of icons the model interacted with oval-objects in the Lisp-GUI with corresponding colors. Instead of the graphic on the icon, text codes were used simulating the visual feature that requires attention shifts. Both this basic and the later, modified model were originally created as part of a student project.

Basic Model

At the beginning of each trial, the model starts by encoding and memorizing the target icon in short term working memory (imaginal buffer). When the fixation cross appears, the visual focus is set on it. Starting with the appearance of the icons the model uses a search routine to scan the graphic user interface (GUI) for the target. Using preattentive perception via the *where* system, it starts a visual-location request for the target color. Its visual attention is then directed to such an item location in order to encode it (text code or icon graphic). The current icon and the target icon (stored in working memory) are compared. Whenever the two items match, the icon is selected. If they do not match, the next item with the target color is picked out and the process repeats until all items with the target color have been attended. If there is no unattended item left, the "not present"-icon at the bottom screen is selected.

While this search routine could plausibly simulate human behavior, this first model had several shortcomings. Most problematically, almost all model behavior was longer than the participants'. This difference was most pronounced in conditions 3 and 4 where many distractor items match the target color and thus the "naive" model had to spend a large amount of time on time-costly fixations of the *what*-system. An additional problem was the fact that the model produced shorter reaction times with no target present (compared to the same condition with target present) in the first two conditions. This was mainly due to the additional visual fixation on the target when the target was present.

Model Changes

To increase the speed, while keeping the model psycho-physiologically plausible, we realized three adjustments: The first adjustment was to move the starting position of the cursor to the center of the screen, assuming that most participants would keep the finger in a click-ready position over the display to be able to react faster. Secondly, as soon as the *where*-system returns a new visual location, two processes start in parallel. While the visual attention is drawn to the location, the manual system prepares to start moving the finger towards the new candidate item. This change was implemented to reflect a routine task handling with subjects constantly anticipating and preparing the next step of the task. Thirdly, while the movement towards an icon takes place, the model already starts to prepare the next motor movement (the pressing of the icon). ACT-R allows for this kind of parallel working of the motor module (here specifically via the "preparation: free" command) as long as the different processes are in different stages of the preparation-initiation-execution sequence that makes up all motor processes. Psychologically, this change can be justified by the assumption that most participants are well-versed in the action of pressing an icon on a touch screen. A procedural acquisition of a combined movement by the participants that does not require several separate preparation and initiation phases is therefore plausible.

Hybrid search strategy

The most important change, however, was the remodeling of the general search in a way that it required fewer attentional fixations, driving down reaction times especially in conditions 3 and 4. Since the fixation of every candidate item (i.e. items of the same color as the target item) was not reconcilable with observed reaction times, the next logical step was to use a strategy that searches an entire cluster of candidate items with one fixation. The new search algorithm (figure 2) thus consists of three main steps (3 productions in ACT-R).

1) A preattentive request (where system) is issued for a previously unattended location with the target color and the lowest x and y values (light blue arrow).

2) If the icon does not match the target icon, the model scans the entire row for the target comparing the "width" of the text (black arrow). Width is processed preattentively, and the target had a unique text width, allowing it to be a search criterion. Psychologically, this much faster search assumes that a visual scan within a short row (here 4 items) allows the shape of the target item to visually "pop out" as well (an assumption that we globally allowed only for colors). When the icon is located in the row, it can be found directly.

3) In case of finding nothing, the model jumps to the nearest oval with the correct color below the current row (red arrow).

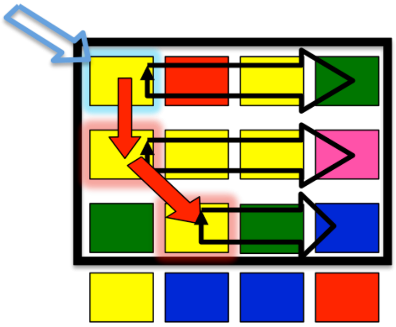


Figure 2: Core visual search algorithm of the final model.

The aggregate of these adjustments allows our model not only to meet the general level of reaction time of the empirical data.

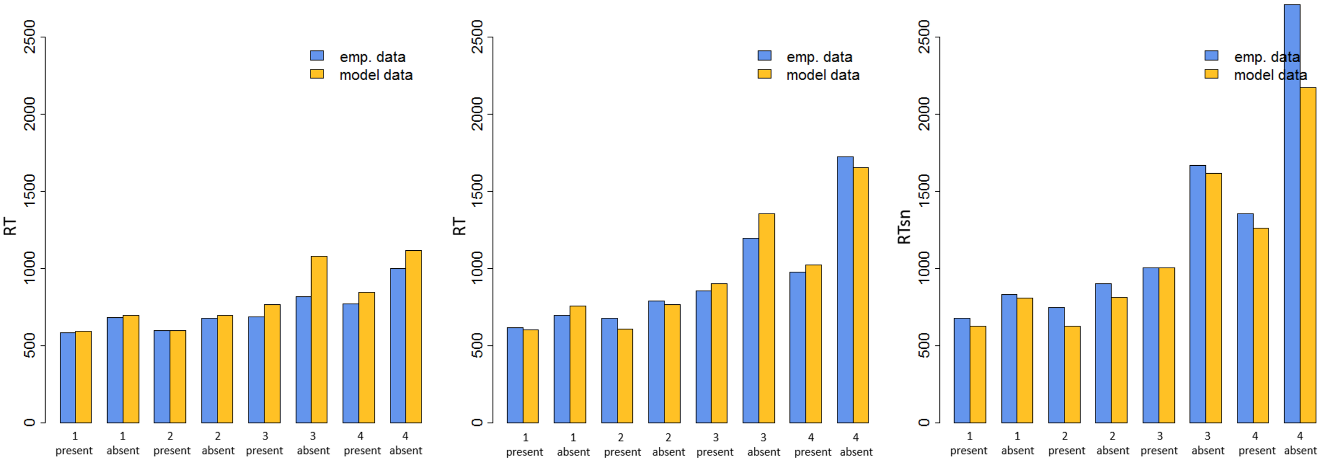


Figure 3: Empirical and Model reaction times for all four principal experimental conditions. Each condition is in turn divided by target present/ not present.

The model now also produced reaction times that are longer in conditions with no target present without any further assumptions.

It should be noted that in this paper, our goal was to recreate reaction times rather than the exact visual paths or fixation patterns. In fact, all models that assume that

1) exactly one fixation per row is needed to scan all candidate items and
2) search is conducted in a structured manner from top to bottom

make the same temporal predictions.

Model Specifications

In the models, no ACT-R parameters were used. The declarative memory consists of a goal chunk and a chunk that stores color, text and width of the current target item. The final model and the GUI are published online at https://depositonce.tu-berlin.de/handle/11303/338. To obtain the simulation results, the model was run 1000 times in each condition.

Results and Model Fit

The following table shows statistical results of the fit. The overall RMSSD (a measure for the absolute distance between model and experimental data; Schunn & Wallach, 2005) of the model is 1.74.

	RMSE	RMSSD	Correlation
Set size 24	102.34 ms	1.34	0.99
Set size 16	172.12 ms	1.20	0.98
Set size 8	179.62 ms	2.42	0.94

Table 1: Statistical model fit (RMSE: absolute fit; RMSSD: absolute fit standardized by the experimental data's standard deviation).

Comparing the empirical data and the model for the set size of 24 icons indicates a good fit over all conditions (figure 3). It captures well the relative trend in all 3 set sizes, both concerning the similarity conditions and the target presence.

The absolute reaction times match reasonably well, although the fit is best for large set sizes. Almost all of the difference between model and experiment results from conditions 3 and 4 when the target is not present. Especially in condition 4 the difference is not within the standard deviation anymore and therefore has a great effect on the RMSSD.

The reaction times reflect the fact that the model is using both the serial and the parallel visual search. While in condition 1 and 2 the target pops out immediately, the model has to use a mixture between parallel and serial search for finding the target fast enough in the other conditions. Despite the fit getting a little less precise in the 3rd and 4th condition the model's searching algorithm seems to be a good approximation to the human visual search behavior.

Discussion

We introduced two modeling approaches. The first one was a simple, reasonable model to address visual search in a task that includes different similarities between target and distractors. This basic model did not capture well the effects found in the experiment. With three adjustments and a new way to describe a mixture of parallel and serial search the new model could capture the empirical data well. The general mechanism used in the model might be helpful to researchers who model visual search in applied tasks, especially for tasks where time is sparse and people try to be as efficient as possible.

To better judge the quality of the current model, it would be useful to compare it to a model that uses both EMMA- and PAAV-module and thus implements more sophisticated mechanisms including those that deal with bottom-up visual processing. Another factor that could also be included in future models is the influence of expectations on visual search patterns, as described in Lindner & Russwinkel (2016).

Furthermore this account is a theoretical concept that should be tested in subsequent experiments. To test the assumption of the pop out of items on one row, a variation of the current experiment could test participants with the screen presented vertically and horizontally. Another variation could address the question of a possible strategy change when the number of distractors similar to the target increases. This can be done by presenting conditions in which only a small number of distractors is clearly different from the target. All future experiments should involve eye-tracking to better track the attentional focus of participants. This in turn should allow for better model construction and evaluation. Furthermore, we would like to test this visual search concept on our model of learning and unlearning of app usage (Prezenski & Russwinkel, 2016).

Acknowledgments

We would like to thank Anna Trapp and Charlotte Wienrich for sharing their experimental data and Lisa-Madeleine Dörr for creating multiple ACT-R GUIs for our model and for providing modeling ideas. We also thank Lou Conradi and Philipp Wolfgang Klemm for valuable ideas regarding model construction.

References

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

Anderson, J. R., Bothell, D., Byrne, M. D.,Douglass, S., Lebiere, C., & Quin, Y. (2004). An integrated theory of mind. Psychological Review, 4, 1036–1060.

Duncan, J., & Humphreys, G. W. (1989). Visual Search and Stimulus Similarity. Psychological Review 96 (3), p. 433–58.

Everett, S., P., & Byrne, D.,B., (2004). Unintended Effects: Varying Icon Spacing Changes Users' Visual Search Strategy. In Proceedings of ACM CHI '04: Conference on Human Factors in Computing Systems, Vienna, Austria: ACM.

Fleetwood, M. D., & Byrne, M. D. (2006). Modeling the visual search of displays: a revised ACT-R model of icon search based on eye-tracking data. *Hum.-Comput. Interact.* 21, 2 (May 2008), 153-197. DOI=http://dx.doi.org/10.1207/s15327051hci2102_1

Lindner, S. and Russwinkel, N. (2016). Modeling of proximity-based expectations. In D. Reitter & F. E. Ritter (Eds.), Proceedings of the 14th International Conference on Cognitive Modeling. University Park, PA: Penn State.

Nyamsuren, E. & Taatgen, N. (2012). Pre-attentive and attentive vision module. In N. Rußwinkel, U. Drewitz & H. van Rijn (eds.), Proceedings of the 11th International Conference on Cognitive Modeling, Berlin: Universitaetsverlag der TU Berlin.

Prezenski, S. and Russwinkel, N. (2016). Towards a general model of repeated app usage. In D. Reitter & F. E. Ritter (Eds.), *Proceedings of the 14th International Conference on Cognitive Modeling*. University Park, PA: Penn State.

Reifers, A. L., Schenck, I. N., & Ritter, F. E. (2005). Modeling pre-attentive visual search in ACT-R. In B. Bara, L. Barsalou & M. Bucciarelli (Eds.), *Proceedings of the 27th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. Cognitive Systems Research, 1(4), 201–220.

Schunn, C. D., & Wallach, D. (2005). Evaluating goodness-of-fit in comparison of models to data. Psychologie der Kognition: Reden und Vorträge anlässlich der Emeritierung von Werner Tack, 115-154.

Trapp, A. K., & Wienrich, C. (2017). App icon similarity and its impact on visual search efficiency on mobile touch devices. *Manuscript submitted for publication.*

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology, 12,* 97–136.

Wolfe, J. M. (1994). Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review, 1(2)*, 202–238.

Spatial relationships and fuzzy methods: Experimentation and modeling

James Lee Ward (robotcycle@gmail.com)
Army Research Office, Research Triangle Park, NC

Robert St. Amant (stamant@ncsu.edu)
Computer Science Department, North Carolina State University, Raleigh, NC

MaryAnne Fields (mary.a.fields22.civ@mail.mil)
U.S. Army Research Laboratory, 28000 Powder Mill Rd. Adelphi, MD

Abstract

This paper describes an experiment and fuzzy set models in the domain of linguistic labels for simple spatial relationships: for example, that one object is "in front of" or "to the right of" another. Input to the models was generated by robot sensors (camera and distance sensors), from a viewer perspective on different configurations of two objects. Performance of the models is is qualitatively similar to human judgments; performance is also quantitatively similar to that of a model working from an environmental bird's-eye view. Such models are part of a robot's interpretation of the context of human activity.

Keywords: spatial relationships; fuzzy sets; cognitive robotics

Introduction

As we attempt to create a new generation of automated helpers to solve problems in the military, elder assistance, transportation, and other areas, we increasingly find that we need robots that can interact naturally with humans and that can move through environments designed for humans. A critical challenge for such robots is the use of context.

Context can help a robot to impose structure on information available to it, in a top-down manner. Some kind of contextual information can be provided by background knowledge or experience of common human activities. For example, "writing a paper" may be associated with scenes such as a "computer lab" or a library, or with object clusters such as a table and chair (Fields, Lennon, Martin, & Lebiere, 2017).

Context is also provided by information about human behavior and performance, which can be exploited in research that integrates cognitive modeling and robotics. Our lab has begun to explore the combination of language comprehension models and information-based search; some of our current research deals with spatial relationships.

Consider the diagram of two objects in Figure 1, one labeled L (for "landmark") and the other T (for "trajector"). The landmark sets the context for the relationship, while the trajector occupies a position—a place in the relationship—with respect to the landmark. If this diagram were a bird's-eye view of a room, a person in the position of observer O_1 would probably say that "T is to the right of L." Would the person also say that "L is in front of T" or that "T is in back of [or behind] L"? Spatial relationships that can be easily diagrammed may be ambiguous when described in language.

The ways that people conceptualize space (and action) have long been a subject of study in psychology. Applying research findings to robot behavior is a more recent development. There are clear advantages in human-robot interaction for a robot that incorporates the ability to take as input, generate as output, or reason about expressions of spatial relationships (Trafton & Harrison, 2011; Guadarrama et al., 2013; Tellex et al., 2011). The work of Regier and Carlson (2001) and others hints at another possibility: a model of human interpretations of spatial relationships may provide information to a robot about what is of interest to individuals or to people in general. For a simple example, people typically attend to what is in front of them; in a classroom full of desks and chairs, it is straightforward to infer the general area a teacher will occupy. We even find spatial directions used in metaphorical language concerning attention: "it's right in front of you" indicates that you should notice whatever it is.

To explore such issues, it will be useful to have a reliable way for a robot to associate spatial relationships with labels such as *left*, *right*, *in front of*, and *in back of*, in the same way that humans do. While there are obvious, canonical examples of such relationships, not all fall crisply into one category or another. Further, robots must deal with noisy sensors and motor movements, which might plausibly interfere with their categorizations of objects in the environment.

In the remainder of this paper we give a brief overview of work on linguistic labels for spatial relationships. We describe an experiment in which participants made judgments about spatial relationships between two objects. We then describe three *a priori* models, from the fuzzy systems literature (Keller & Wang, 1995), that allow a robot to make the same viewer-perspective judgments about the spatial relationships. We compare their performance to the human data and find qualitatively similar model predictions.

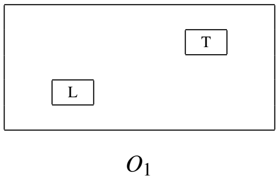


Figure 1: Object/observer relationships

Related work

The literature related to representation, reasoning, and communication concerning spatial relationships is enormous. Mechanisms underlying spatial representation and reasoning have been explored in some depth in the cognitive modeling literature (Harrison & Schunn, 2003; Gunzelmann & Lyon, 2006; Trafton & Harrison, 2011); cognitive models also encompass the language of spatial relationships (Ball, 2015). The research described in this paper is much narrower, however, focusing on spatial relationships that can be expressed in linguistic terms (e.g., *left*, *right*, *above*, *below*, *in front of*, and *behind* or *in back of*) and the extent to which they can be grounded in the perception (of a human or a robot agent).

Regier and Carlson (2001)'s Attentional Vector-Sum (AVS) model is widely accepted as the best model of how a set of spatial expressions can be grounded in perception. Informally, for the *above* relationship, an attentional beam is conceptualized as extending from a trajector object to a landmark object. Attention is strongest at the point on the landmark directly below the trajector, and weaker at other points on the landmark as distance from this point increases; the drop-off is a free parameter in the model. A distribution of vectors is identified, originating at different points on the landmark and directed toward the trajector, the magnitude of each determined by attention. The sum of these vectors is compared with a vertical line, and the deviation determines the extent to which the trajector is above the landmark.

Regier and Carlson (2001) compare the AVS model with others, including a Bounding Box (BB) model and a Proximal and Center-of-Mass (PC) model, both of which it outperforms. For the BB model and the *above* relationship, "a trajector object is above a landmark object if it is higher than the highest point of the landmark and between its rightmost and leftmost points" (Regier & Carlson, 2001). For the PC model, consider a vector from the center of a landmark to the center of a trajector. As this vector deviates from the vertical (roughly, 68° to 72°) ratings of the *above* relationship decrease linearly; further increases cause a much faster drop off, to zero at 90° or greater. Proximity comes into play with a line segment connecting the landmark and trajector at their minimum distance; to the extent that this segment is aligned with the center-of-mass vector, the *above* relationship holds.

Judgments about *above* generalize to comparable relationships, including *left*, *right*, *in front of*, and so forth (Regier & Carlson, 2001). For example, if we interpret the bottom of the box in Figure 1 as being a horizontal surface, then we could ask, "Is *T* above *L*?" and use the same PC model, reinterpreted, to answer the question.

The AVS model and others have been used in computer vision and robotics research, though they typically require some adaptation. In most experiments on labeling spatial relationships, a scene is presented in which the relationship of interest is visible in a plane normal to the participant's line of sight. For example, consider rating the relationships *left*, *right*, *in front of*, and *in back of* for two objects on the floor in a room. A bird's-eye view, along a normal to the plane of the floor, would be a typical presentation, as in Tellex et al. (2011, Figure 4), which we will call an *orthogonal* view. An observer—such as a robot—inside the room with the objects, however, would face a related but slightly different problem. This viewer perspective, in which judgments are required for relationships that are aligned with viewer's line of sight, are part of the experiment described in the next section.

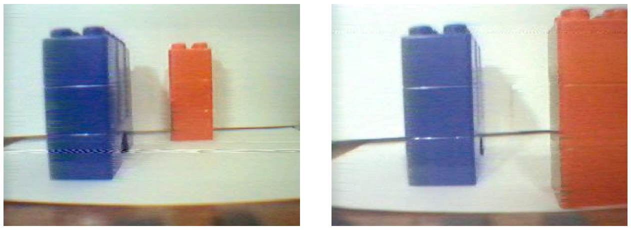


Figure 2: Object configurations (red on the right)

Experiment

This experiment was intended to benchmark performance in answering questions about spatial relationships. Because our eventual goal is a robot that can reproduce human judgment of specific spatial relationships, by reference to a model, the raw data was provided by camera images from a robot: a LEGO Mindstorms NXT robot with customized sensors, including a still camera.

Two stacks of blocks were used for the experiment, as shown in Figure 2. The stack of square red blocks was 10 cm on each side. The oblong stack of blue blocks was 10 cm wide and 30 cm long. Both stacks were about 21 cm tall.

These two stacks were placed in different configurations as follows. The blue stack was initially placed with its narrow side facing the robot, and the red stack was placed to the right and a few cm behind the blue stack, comparable to a configuration in Gapp (1995), in Figure 1b. The red stack was then advanced in increments of 10 cm in a straight line towards the robot. The advances were performed six times until the red stack was about 10 cm in front of the blue stack. After the six advances the red stack was returned to its starting position and the blue stack was rotated by a one-eighth turn. The process was repeated. This continued until the blue stack was at a right angle from its original position.

The robot, approximately 60 cm distant from the centroid of the two stacks in the starting configuration, followed the procedure in Figure 3 after each change in the configuration. Thirty images were collected in total, at six different locations of the red stack and five different rotations of the blue stack. Figure 2 shows two images, the starting configuration on the left and after five steps into the procedure on the right.

The images[1] recorded during this procedure formed the basis of a survey. Twelve participants completed the survey,

[1] Images were used instead of a real environment for consistency across experiment participants.

```
Record compass reading
Record camera image
For target color in {RED, BLUE}
    Identify object of target color
    For target in {LEFT, CENTER, RIGHT}
        CALC: Calculate rotation to center on target
            Rotate
            If within threshold
                Record compass reading
                Record distance reading
            else go to step CALC
```

Figure 3: Measurement procedure

eight men and four women, ranging in age from 28 to 71. The participants received no compensation for participation and were not observed during the task. The sequence of images was randomized; all participants saw the same ordering. For each image, four statements were evaluated by participants, on a scale of 0 to 10; for analysis, all values were linearly transformed to a unit scale.

1. The red blocks are to the right of the blue blocks.
2. The red blocks are in front of the blue blocks.
3. The red blocks are in back of the blue blocks.
4. The blue blocks are to the left of the red blocks.

In other words, we have two independent variables in this experiment. The variable Distance of the red stack to the robot provides for different participant ratings concerning whether the red stack is in front of, in back of, and even to the right or left of the blue stack, in each location. The variable Angle, for the rotation of the blue stack, provides a different cross-section to the viewer as well as a different angle with respect to the red stack.

Note that the experiment excludes the most "obvious" configurations for Front and Back ratings—for example, with one block directly in front of the other, from the position of the camera. There is a sense in which the experiment tests "edge cases" for spatial relationship judgments.

The three plots in Figure 4 show survey ratings for the *right*, *front*, and *back* questions. Values for *left* are not shown, being almost identical to *right*. Each group of six connected dots shows the mean values, scaled from 0.0 to 1.0, the six locations of the red stack, at decreasing Distance from the robot's camera. Five groups are shown, with a graphical icon for each Angle value of the blue stack. Within each group, the sequence shows the red stack moving foward in steps.

No significant effect on Right ratings or on Left ratings was found. The mean values of Left and Right were above 0.9, over all trials, the median equal to 1.0. While a slight inverted U-shaped pattern is visible, we did not analyze the data further.

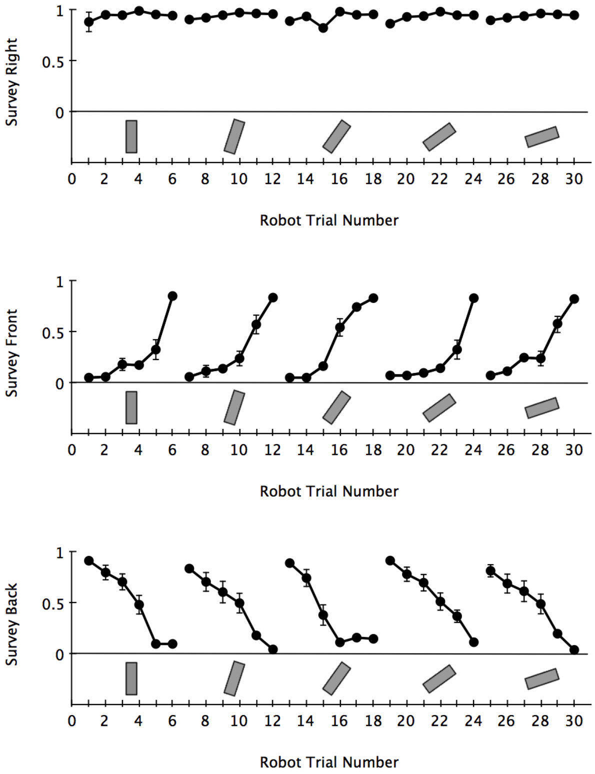


Figure 4: Mean ratings of *right*, *front*, and *back*, with standard error bars; gray blocks show rotation angle of blue stack

An analysis of variance showed a significant overall effect of Distance and Angle on Front ratings, as expected ($F(5,4) = 25.790, p < 0.01$). Distance alone had a significant effect on Front ($F(5,4) = 28.036, p < 0.01$), but Angle did not ($F(5,4) = 0.034$, n.s.); the interaction between Distance and Angle was significant ($F(5,4) = 2.677, p < 0.01$).

Similarly, ANOVA showed a significant overall effect of Distance and Angle on Back ratings ($F(5,4) = 18.973, p < 0.01$). Distance alone had a significant effect on Back ($F(5,4) = 27.505, p < 0.01$), but Angle did not ($F(5,4) = 0.433$, n.s.); the interaction between Distance and Angle was significant ($F(5,4) = 1.812, p < 0.05$).

The general patterns are as expected: experiment participants were able to make plausible judgments about the spatial relationships between the two stacks in different configurations, basing their judgments on the information provided by the robot's camera. There was no influence of the angle of the blue stack, acting as a landmark in the experiment, though the rotations acted to change the "overlap" with the trajector; the lack of an effect is possibly due to the front view and the limited depth information available in the images. Different shapes are noticeable in the Front and Back ratings, with the Front ratings showing a slightly more pronounced curvature with change in Distance.

Modeling

Models such as AVS, PC, and BB have been adapted for use in computer vision and robotics (Guadarrama et al., 2013; Tellex et al., 2011), and new models have been developed Matsakis and Wendling (1999). In such work, however, the models are generally not evaluated directly or compared with human performance, and the models may not take a parameterized form with explicitly identified features, with Gapp (1995) being an exception.[2]

Our work adopts vector-based methods to model the ratings described in the previous section. Regier and Carlson's PC model is a possible candidate, but its four free parameters make it difficult to adapt—specifically, changing to a viewer perspective to evaluate relationships parallel to the viewer's line of sight. Instead, we evaluate three simpler models from the computer vision literature due to Keller and Wang (1995).

These are fuzzy set models, which can deal with membership grades in categorization. In a standard "crisp set" formulation, a predicate is true or false; with a fuzzy set, a membership grade can be a value from 0 (not a member of the set) to 1 (a member of the set). Thus, for example, in Figure 1, T might have a membership grade of 0.8 for the categorization "to the right of L;" it would be greater if it were closer to a horizontal line extending through L. Fuzzy methods can do more than assign a grade for a given label; with several labels, they can be used for categorization, even in cases where a given configuration fall into more than one category.

Keller and Wang's *Centroid* method uses Equation 1 as the membership function for the *right* function; analogous functions are defined for *left*, *front*, and *back*. Let the centroid of L be the origin in a Cartesian coordinate system; let θ be the angle of a vector $\overrightarrow{LT}$ through the centroid of T. The function μ_{right} maps θ to a value between 0 and 1, representing the degree to which T is to the right of L:

$$\mu_{right}(\theta) = \begin{cases} 1 & |\theta| < a\frac{\pi}{2} \\ 0 & |\theta| > \frac{\pi}{2} \\ \dfrac{\pi/2 - |\theta|}{\pi/2(1-a)} & o.w. \end{cases} \quad (1)$$

In words, T is maximally to the right of L when θ is within a small range above or below 0 radians ($a\frac{\pi}{2}$, where a is a free parameter, which we set to 0.05 as a default), along the implicit x-axis in the landmark-based coordinate system. μ_{right} decreases linearly as θ increases or decreases, reaching zero when the center of L falls on or below zero on the x-axis.

Computing the centroids of objects is straightforward with an orthogonal view, but this is more difficult for some relationships from a viewer perspective. As described in the measurement procedure in Figure 3, the robot identified three points on each object in the scene, its left edge, center, and right edge; distances were measured to these points and a centroid computed from the values. This gives the centroid estimation for both objects a strong forward bias, though this is partly alleviated for higher Angle values in cases where the blue stack has been rotated.

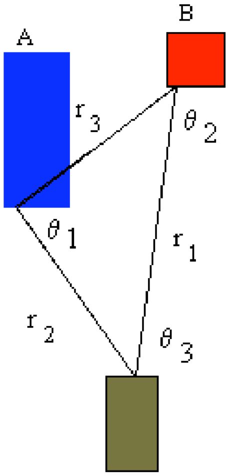


Figure 5: Distance and compass readings, bird's-eye view

The angle θ between the two stacks must be provided as input to μ_{right} but cannot be measured directly; θ is computed, based on the triangle in Figure 5. θ_3, r_1, and r_2 are measured directly by the robot, while θ_1, θ_2, and r_3 are computed:

$$\theta_1 = \cos^{-1}\left(\frac{r_2{}^2 + r_3{}^2 - r_1{}^2}{2r_2r_3}\right); \theta_2 = \pi - \theta_3 - \theta_1, \quad (2)$$

$$r_3 = \sqrt{r_1{}^2 + r_2{}^2 - 2r_1r_2\cos\theta_3}. \quad (3)$$

With measured or computed values for the triangle's sides and angles, plus the assumption that the robot's camera is midway between the two stacks, computing θ for different directions is straightforward trigonometry.

Keller and Wang's second method is *Angle Aggregation*, which samples points from the landmark and trajector objects, computes angles for each pair of points, and aggregates the angles (by a generalized mean operator) into a single value for θ. With three sampled points on each object in the experiment, a total of nine pairs of angles are considered.

The third method is the compatibility method, which we implemented in variant form as a *Histogram/Composition* method. Angles between the two objects are computed as with Angle Aggregation, but instead of computing the mean of the samples, the samples are binned into a histogram, with normalized frequencies for the bins being treated as fuzzy set. This set is composed with the fuzzy sets for *right*, *left*, *front*, and *back*, to compute the membership grade in each category.

[2] Indirect evaluation is carried out, however. Guadarrama et al. (2013) evaluate overall measures of success for experiments with a robot that incorporates a combined PC and BB model to interact with configurations of multiple objects; Tellex et al. (2011) similarly with an AVS model.

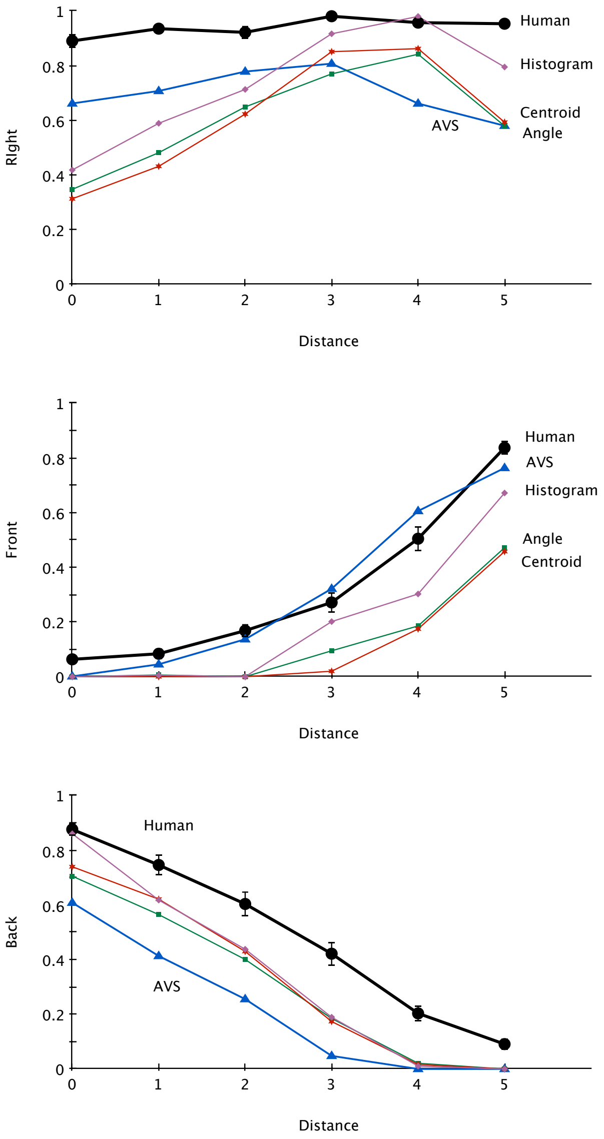


Figure 6: Model predictions for *right*, *front*, and *back*

Model predictions are shown in Figure 6 for the survey data for *right*, *front*, and *back*; as before, *left* is elided, being almost identical to *right*. The parameters of the models were not tuned to fit the data.[3] In these plots, Distance values reflect the ordering in which they were considered by the robot, an inversion of the actual distance from the camera: 0 corresponds to the red stack being farthest away from the robot's camera, i.e., behind the blue stack, with a Distance value 1 being closest to the camera and in front of the blue stack. Predictions are aggregated by Distance value, in that Angle had no effect on the spatial judgments.

The black lines show mean ratings from the survey, with error bars showing the standard error with respect to all participant ratings per Distance value. The other colored lines represent predictions of the Centroid, Angle Aggregation, and Histogram/Composition fuzzy models. The fuzzy models are not separately labeled in the Back plot; their values are almost indistinguishable. For reference, predictions of the AVS model are shown as well. The AVS predictions were based on a virtually constructed orthogonal diagram of each configuration and are given for comparison to a method with access to a perspective not available to the other models.

	Right	Front	Back
AVS	0.891, 0.068 $R^2 = 0.056$	0.046, 0.880 $R^2 = 0.813$	0.285, 0.925 $R^2 = 0.629$
Angle	0.888, 0.083 $R^2 = 0.202$	0.139, 1.449 $R^2 = 0.812$	0.194, 0.946 $R^2 = 0.835$
Centroid	0.887, 0.085 $R^2 = 0.282$	0.164, 1.453 $R^2 = 0.792$	0.193, 0.900 $R^2 = 0.850$
Histogram	0.902, 0.051 $R^2 = 0.138$	0.151, 0.867 $R^2 = 0.698$	0.208, 0.797 $R^2 = 0.817$

Table 1: Model fit statistics

Qualitatively, the fuzzy models are consistent with human ratings, though they are more "conservative" in the sense of assigning lower membership grades than the human participants. This is in part due to the forward bias in the *front* predictions—recall that reference points were identified with the robot's distance sensors, which detect the distance to the front of each object.

The under-predictions of the AVS model for the mean Back rating surprised us; we initially suspected design or implementation errors in the modeling code. Exploration of the participants' data led us to a different conclusion, however.

Leaving out Distance values of 5, the remaining conditions are approximately symmetrical, front-to-back, in geometrical terms. For example, at Distance 0, the front edge of the red stack is aligned with the back edge of the blue stack, while at Distance 4, the back of the red is aligned with the front of the blue, as if reflected in a mirror (though rotation introduces minor asymmetry). The AVS model produces consistent predictions for such symmetrical configurations.

Participant ratings did not show the same consistency between Front and Back ratings. For the symmetrical configurations, Back ratings were higher than Front ratings by about 0.2 on the unit scale. Further experimentation would be needed to verify this bias. It is generally held that models of spatial relationships such as these generalize across orientations; our results suggest that the viewer's perspective may be a factor in the magnitude of ratings.

Table 1 shows how well the models fit the survey ratings, following the approach of Regier and Carlson. Each entry is the y-intercept and slope of a regression that uses the model's output to predict ratings; an R^2 value is below each such entry.

[3] For Keller and Wang's models, $a = 0.05$. The AVS model used parameters given by Regier and Carlson (2001, Table 1): $\lambda = 1.0$ (attentional field width); y-intercept $= 1.007$ and slope $= -0.006$ (alignment function); gain $= 0.131$ (top sigmoid).

For example, we see that AVS gives the best fit for Front: a regression line with a y-intercept of 0.046 and a slope of 0.880 gives the best prediction of survey ratings, with $R^2 = 0.813$. The better the model, the closer the intercept is to zero and the closer the slope is to 1.

Table 1 shows that of the fuzzy models, the Histogram model has the best balance of intercept and slope for the *front* relationship, though a lower R^2. Angle Aggregation is the best for the *back* relationship, though all the fuzzy models are similar. All models perform poorly for *right* and *left* relationships, which is inevitable—Distance and Angle have no predictive value on the participants' ratings.

We tested the sensitivity of the modeling predictions by virtually reconstructing each configuration and running the models on the "theoretical" values for $\theta_1 \dots \theta_3$ and $r_1 \dots r_3$. We found no marked difference between the predictions based on the sensor data and the theoretical data; data is not presented for reasons of space.

To summarize, the fuzzy models' predictions tend to underestimate ratings in all categories of spatial relationships that we tested. The predictions for *front* and *back* are of the same shape as the human ratings with respect to relative distance from the viewer, however, which suggests that an additional constant or linear factor could improve their performance. Poor performance on *right* and *left* is partly due to this underestimate; this does not account for the models' systematic dependence on Distance, however. The AVS model was used for comparison, based on a virtual bird's-eye view. With access to depth information about the stacks, the AVS model outperformed the fuzzy models for the *front* relationship but was considerably worse than the fuzzy models for the *back* relationship, due to asymmetrical ratings by the experiment participants. We leave these issues for future work.

Conclusion

The long-term thrust of this area of research is to give robots a language of spatial relationships that are consistent with human understanding. This can facilitate human-robot interaction and potentially improve a robot's ability to interpret human activity or designed environments. Some of the work cited in this paper goes much farther toward this goal than we have here, and another direction for future work is to determine how best to integrate our results with theirs.

Our results are nevertheless informative. One of the challenges in determining spatial relationships is the uncertainty of data in dealing with an egocentric view; another is in noisy sensor data. A step toward this goal is to identify a technique that gives results in line with human understanding. This paper compared three fuzzy methods with human survey data to find if any of the techniques performed acceptably against human perception. These techniques were developed for judgments about orthogonal presentations and performed approximately as expected from a viewer perspective.

Only four primitive spatial relationships were used in this work; many more would need to be addressed in an effective vocabulary: near, far, surrounding, inside, outside, and so forth. Another direction for future research is to determine the minimum amount of information the robot must sense in the environment before being able to make accurate predictions about the spatial relationships. Our work used three points on each object. With more points some of our models described might have produced better predictions. The work presented here compares relatively straightforward methods of determining spatial relationships given the current scene available. But if the robot moves a new scene is presented and any information from previous scenes is not incorporated into the current calculations of spatial relationships. This could play a part in determining spatial relationships with human robot interaction. A final interesting question is whether people use only available information in the picture to determine the spatial relationships between objects or whether they incorporate background knowledge or previous experience.

Acknowledgments

We would like to thank two anonymous reviewers for suggestions that improved our understanding of our results. This work was partly supported by the Robotics Collaborative Technology Alliance of the U.S Army Research Laboratory.

References

Ball, J. T. (2015). *Toward a logical description of Double-R grammar.* (www.doublertheory.com/double-r-grammar/logical-description.pdf)

Fields, M., Lennon, C., Martin, M., & Lebiere, C. (2017). Priming for autonomous cognitive systems. In *Proc. SPIE 10194.*

Gapp, K.-P. (1995). An empirically validated model for computing spatial relations. In *Proc. KI* (pp. 245–256).

Guadarrama, S., Riano, L., Golland, D., Go, D., Jia, Y., Klein, D., ... Darrell, T. (2013). Grounding spatial relations for human-robot interaction. In *Proc. IROS* (pp. 1640–1647).

Gunzelmann, G., & Lyon, D. R. (2006). Mechanisms for human spatial competence. In *Proc. Spatial Cognition* (pp. 288–307).

Harrison, A. M., & Schunn, C. D. (2003). ACT-R/S: Look ma, no "cognitive-map"! In *Proc. ICCM* (pp. 129–134).

Keller, J. M., & Wang, X. (1995). Comparison of spatial relation definitions in computer vision. In *Proc. Uncertainty Modelling and Analysis* (p. 679).

Matsakis, P., & Wendling, L. (1999). A new way to represent the relative position between areal objects. *IEEE Transactions On PAMI*, 634-643.

Regier, T., & Carlson, L. A. (2001). Grounding spatial language in perception: an empirical and computational investigation. *JEP: General*, *130*(2), 273.

Tellex, S. A., Kollar, T. F., Dickerson, S. R., Walter, M. R., Banerjee, A., Teller, S., & Roy, N. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of AAAI.*

Trafton, G., & Harrison, A. (2011). Embodied spatial cognition. *Topics in Cognitive Science*, *3*(4), 686–706.

Generating Random Sequences For You: Modeling Subjective Randomness in Competitive Games

Arianna Yuan (xfyuan@stanford.edu) and **Michael Henry Tessler** (mtessler@stanford.edu)
Department of Psychology, Stanford University

Abstract

Generating truly random sequences is hard. When participants are engaged in a competitive game (e.g., Matching Pennies), the sequences they generate are surprisingly *more random* than when given explicit instructions to generate random sequences (Rapoport and Budescu, 1992). To explore this phenomenon, we formalized two probabilistic models of Theory of Mind reasoning about subjective randomness. One model (the *Fair-Coin* model) assumes participants predict their opponents' choices by implicitly assuming that their opponents intend to generate binary sequences that simulate the outcome of tossing a fair coin. The other model (the *Markov* model) assumes participants believe that their opponents intend to generate sequences that simulate the outcome of a Markov process with transition probability equal to 0.5. We find that Theory of Mind models of both the Fair-Coin and the Markov definitions of subjective randomness are able to characterize the calibrated subjective randomness that occurs when participants are playing an iterated competitive game (Matching Pennies), but the Markov Model is better than the Fair-Coin Model in simulating the situation where participants need to specify their choice sequences in advance of the game. The current study suggests that the *calibrated* subjective randomness in competitive games can be explained by the online evaluation of sequence randomness with Theory of Mind reasoning.

Keywords: subjective randomness; Theory of Mind; matching pennies; probabilistic models

Introduction

People are relatively poor at generating random sequences (Bakan, 1960). They produce "subjectively random" sequences by switching between heads and tails, but they switch too often (Lopes & Oden, 1987). However, participants are able to generate sequences that are more "truly random" when feedback is available (Neuringer, 1986), as in competitive games (Rapoport & Budescu, 1992). Many theories have been proposed to account for participants' failure to generate random sequences (Griffiths & Tenenbaum, 2003; Hahn & Warren, 2009), but the phenomenon of *calibrated subjective randomness* in the context of feedback has received comparatively little attention by formal models (however, see West and Lebiere (2001); Lee, Conroy, McGreevy, and Barraclough (2004) for some proposals). Here, we propose that calibrated subjective randomness may result from basic Theory of Mind reasoning about others' models of subjective randomness.

The situation we will examine is the competitive game Matching Pennies. In this game, two agents (the "Matcher" and the "Nonmatcher") each make a binary choice (e.g., 0 or 1) secretly. The Matcher wins if their choices are the same and the Nonmatcher wins if their choices are different. Mathematically, the optimal strategy is to choose the two alternatives (0 or 1) with equal probability. Rapoport and Budescu found that participants generated more truly random sequences when playing in the Matching Pennies game.

To understand why feedback helps calibrate subjective randomness, we must first understand the origin of subjective randomness. Several researchers propose that subjective randomness is the result of instructional biases (e.g., when participants are instructed to generate random sequences, they are encouraged to produce sequences that appear to be orderless; Ayton, Hunt, & Wright, 1989). However, even when participants are not explicitly given such instructions, the nature of the task prompts participants to generate sequences that are more "representative" of the output of a random process (Kahneman & Tversky, 1974). Griffiths and Tenenbaum formalized this idea into a probabilistic model of subjective randomness (Griffiths & Tenenbaum, 2001).

Although these theories explain the nature of subjective randomness, they provide no mechanistic explanation as to why this bias is calibrated in the presence of feedback. In the current investigation, we propose that Theory of Mind reasoning can explain participants' behavior in experimental conditions both with and without feedback. The main goal of this paper is to model and explain the data from Rapoport and Budescu (1992). Below we first review the relevant experiments and results. Next, we introduce two computational models, and explore and compare the model predictions in different experimental conditions. Finally, we discuss the implications and limitations of our models.

Calibrated Subjective Randomness

Rapoport and Budescu (1992) ran a subjective randomness experiment with three conditions. In the **Dyad Condition**, participants were paired to form dyads; each dyad played 150 trials of Matching Pennies, generating a response on each trial. The **Single Condition** was the same as the Dyad condition except that the paired dyads were asked to specify their choices in advance of the 150 rounds and were told that the responses would be matched on a trial-by-trial basis to determine the winner. In the **Randomization Condition**, participants were instructed to generate a sequence of 150 random binary responses to simulate the outcome of tossing an unbiased coin in a non-interactive context.

The key results are the patterns of sequential dependencies. The distributions of sequences of length k (i.e., k-tuples; $k = 2, 3, 4$) were not uniform as expected under a "true" random generating process. The authors calculated the frequencies of k-tuples (e.g., 3-tuple [0 1 1], 4-tuple [0 0 0 0]), and found that in the Randomization Condition, participants were more likely to generate [0 1 0 1] and [1 0 1 0] than [0 0 0 0] and [1 1 1 1]. They used two statistics to indicate the extent to which the distributions deviate from the outcome of a truly random generating process: the mean absolute deviation (MAD) from expectation and the standard deviation of the observed proportions around their expectation (SD):

$$\mathrm{MAD} = \sum_{j=1}^{2^k} |p_j - 1/2^k|/2^k, k = 2,3,4.$$

$$\mathrm{SD} = \sum_{j=1}^{2^k} [(p_j - 1/2^k)^2/(2^k-1)]^{1/2}, k = 2,3,4.$$

Here p_j stands for the probability of individual k-tuples.

Rapoport and Budescu (1992) found that MAD and SD in the Randomization Condition were the largest, followed by the Single Condition, and finally the Dyad Condition. Participants deviated from what would be expected with truly random sequences the most in the Randomization Condition and the least in the Dyad Condition.

To model these results, we propose that in the Dyad Condition, participants use their opponents' previous choices to predict their opponents' choices in the current trial, assuming that their opponents intend to generate sequences that are "subjectively random". They then generate their responses accordingly (i.e., Matchers try to match, Non-matchers try to mismatch). In addition, we posit that participants also have a desire to have their own sequences be subjectively random. In the Single Condition, participants consider that their opponents know that they are likely to generate subjectively random sequences and adjust their choices accordingly (though without feedback). Finally, in the Randomization Condition, participants simply generate sequences that are subjectively random.

Note that one of the critical components in our model is how participants define "subjective randomness". Different definitions or models of subjective randomness are possible. In a Bayesian setting, the inference of whether or not a sequence is random will depend on the specification of the alternative hypotheses (i.e., what counts as "non-random"). For example, participants may imagine "random" to mean a Markov process with transition rate of 0.5. In this case, "non-random" corresponds to a Markov processes with transition rates other than 0.5, which will generate sequences with too many or too few alternations between 0 and 1, though the total counts of 0s and 1s would be approximately equal. If instead, participants imagine "random" to mean an unbiased coin, ‘‘‘non-random" corresponds to tossing a biased coin, which would generate sequences with many 1s or many 0s. Given these different possibilities, we formalize both when simulating the calibrated subjective randomness effect reported in Rapoport and Budescu (1992). All models were implemented in the probabilistic programming language WebPPL (Goodman & Stuhlmüller, 2014), and model code can be found online[1].

The Fair-Coin Model

Model Description

Randomization Condition The Randomization Condition corresponds to the same experimental scenario described in Griffiths and Tenenbaum (2001). We incorporate their model of subjective randomness into a probabilistic model of communication described in Shafto, Goodman, and Frank (2012). The integrated model assumes that when participants are instructed to generate a random sequence, they try to *convince* the experimenter that the "weight of the coin" is 0.5. This will result in sequences that are more representative of a random sequence, such as [0 1 0]. We denote a sequence of length k to be S_k, which can be viewed as a random variable, and a specific instance of it to be s_k ($k = 2,3,4$). For instance, S_3 can take values like [0 1 0]. We obtain the probability $\mathrm{P}(S_k = s_k)$ by assuming that participants attempt to convince the experimenter that the sequence is randomly generated. The goal of the model is to maximize the probability that the listener (i.e., the experimenter) would think the sequences are generated by a fair coin. The model returns the probability of specific k-tuples such as $\mathrm{P}(S_2 = [01])$.

Dyad Condition In the Dyad Condition, choices are made incrementally. The model assumes participants generate responses based on the previous choice made by their opponent and themselves. Concretely, participant A first simulates different alternatives (0 or 1) her opponent (participant B) might choose in the current trial, and then combine B's previous responses with the current possible responses. Participant A then predicts B's current response according to the probability that the combined sequence is judged as random. Mathematically, the probability of choosing 0 given the previous responses can be calculated using equation (1).

$$\begin{aligned}
&\mathrm{P}(R_k = 0|S_{k-1} = s_{k-1}) \\
=&\frac{\mathrm{P}(R_k = 0|S_{k-1} = s_{k-1})}{\mathrm{P}(R_k = 0|S_{k-1} = s_{k-1}) + \mathrm{P}(R_k = 1|S_{k-1} = s_{k-1})} \\
=&\frac{\mathrm{P}(R_k = 0 \wedge S_{k-1} = s_{k-1})}{\mathrm{P}(R_k = 0 \wedge S_{k-1} = s_{k-1}) + \mathrm{P}(R_k = 1 \wedge S_{k-1} = s_{k-1})} \\
=&\frac{\mathrm{P}(S_k = (s_{k-1},0))}{\mathrm{P}(S_k = (s_{k-1},0)) + \mathrm{P}(S_k = (s_{k-1},1))}
\end{aligned} \quad (1)$$

With equation 1 and the probability $\mathrm{P}(S_k = s_k)$ derived in the Randomization Condition we can compute $\mathrm{P}(R_k = 0|S_{k-1} = s_{k-1})$. For instance, assuming that a player's most recent two choices are [0 1], the probability that he/she would choose 0 in the current trial is given by equation (2), where $\mathrm{P}(S_3 = [010])$ and $\mathrm{P}(S_3 = [011])$ are computed from the model predictions in the Randomization Condition.

$$\begin{aligned}
&\mathrm{P}(R_3 = 0|S_2 = [01]) \\
=&\frac{\mathrm{P}(R_3 = 0|S_2 = [01])}{\mathrm{P}(R_3 = 0|S_2 = [01]) + \mathrm{P}(R_3 = 1|S_2 = [01])}
\end{aligned} \quad (2)$$

[1] https://web.stanford.edu/~xfyuan/psych204Code.

$$
\begin{aligned}
&= \frac{\mathrm{P}(R_3 = 0 \wedge S_2 = [01])}{\mathrm{P}(R_3 = 0 \wedge S_2 = [01]) + \mathrm{P}(R_3 = 1 \wedge S_2 = [01])} \\
&= \frac{\mathrm{P}(S_3 = ([01], 0))}{\mathrm{P}(S_3 = ([01], 0)) + \mathrm{P}(S_3 = ([01], 1))} \\
&= \frac{\mathrm{P}(S_3 = [010])}{\mathrm{P}(S_3 = [010]) + \mathrm{P}(S_3 = [011])}
\end{aligned}
$$

After participant A predicts B's choice in the current trial, A will make a choice according to his/her assigned role, i.e., if A is a matcher, then A will match B's response, otherwise A will choose the opposite response. In addition, participants might also be motivated to generate sequences that are subjectively random so that their choice is not easily predicted by their opponents. We include a weight term w capturing how participants balance these two concerns. The larger the w, the more weight participants put on their opponents' potential choices. In the current simulation, the value of w is set to 0.6. Mathematically, the probability of choosing 0 given previous responses for a matcher is:

$$
\begin{aligned}
&\mathrm{P}(R_k^M = 0 | S_{k-1}^M = s_{k-1}^M \wedge S_{k-1}^{NM} = s_{k-1}^{NM}) \\
=& w * \mathrm{P}(R_k^{NM} = 0 | S_{k-1}^{NM} = s_{k-1}^{NM}) + \\
&(1-w) * \mathrm{P}(R_k^M = 0 | S_{k-1}^M = s_{k-1}^M),
\end{aligned} \tag{3}
$$

and a non-matcher:

$$
\begin{aligned}
&\mathrm{P}(R_k^{NM} = 0 | S_{k-1}^M = s_{k-1}^M \wedge S_{k-1}^{NM} = s_{k-1}^{NM}) \\
=& w * \mathrm{P}(R_k^M = 1 | S_{k-1}^M = s_{k-1}^M) + \\
&(1-w) * \mathrm{P}(R_k^{NM} = 0 | S_{k-1}^{NM} = s_{k-1}^{NM})
\end{aligned} \tag{4}
$$

Using a concrete example to illustrate how equation (3) and (4) should be applied, we assume that the most recent two choices made by the matcher is [0 1], and those two made by the non-matcher is [1 1]. The probability of choosing 0 as a matcher given her and her opponent's previous responses is computed using equation (5) and the one as a non-matcher is computed using equation (6):

$$
\begin{aligned}
&\mathrm{P}(R_3^M = 0 | S_2^M = [01] \wedge S_2^{NM} = [11]) \\
=& w * \mathrm{P}(R_3^{NM} = 0 | S_2^{NM} = [11]) + \\
&(1-w) * \mathrm{P}(R_3^M = 0 | S_2^M = [01]),
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
&\mathrm{P}(R_3^{NM} = 0 | S_2^M = [01] \wedge S_2^{NM} = [11]) \\
=& w * \mathrm{P}(R_3^M = 1 | S_2^M = [01]) + \\
&(1-w) * \mathrm{P}(R_3^{NM} = 0 | S_2^{NM} = [11])
\end{aligned} \tag{6}
$$

The value of the unknowns can be obtained from the results of equation (1). With those probabilities, we calculate the distribution of all the possible k-tuples and compare the model prediction with the empirical data (Figure 1).

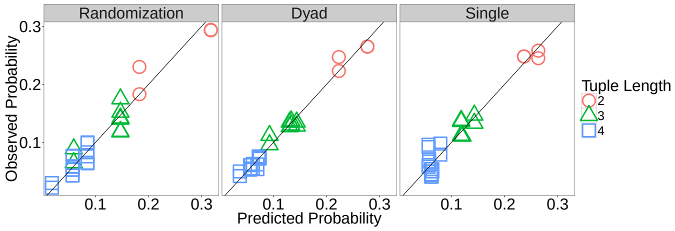


Figure 1: Observed frequencies of k-tuples vs. predicted probabilities of the Fair-Coin Model.

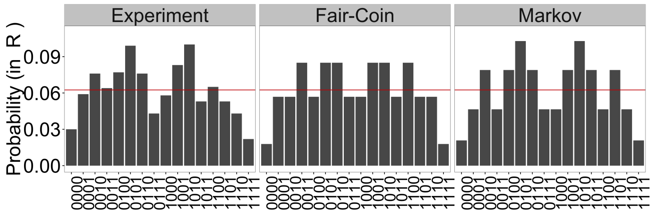


Figure 2: Observed distributions of 4-tuples and model predictions for the Randomization Condition. Left: Empirical data. Middle: Predictions of the Fair-Coin Model. Right: Predictions of the Markov Model. The red line indicates expected probability.

Single Condition For the Single Condition, the model assumes that participants generate responses based on their own previous choices, with the knowledge that their opponent thinks they will generate a random sequence. They extend their previous $k-1$ choices into a k-tuple that is subjectively random. With that subjectively random k-tuple in hand, they make the opposite choice.

Results of the Fair-Coin Model

Randomization Condition Figure 1, Left shows that the model fits the data well, $R^2 = .94$. Figure 2, Middle shows that the model successfully captures the observation that more heterogeneous tuples like [0 1 1 0] are more likely to be generated than less heterogeneous tuples, e.g., [0 0 0 0].

Dyad Condition Although in the Dyad Condition we have different formulae for the matchers and the non-matchers, the simulation results show that the distributions of k-tuples are the same. Therefore, we collapse these two cases (Figure 3, Middle). The model predictions are well aligned with the empirical data (Figure, 1, Middle), $R^2 = .98$. Critically, the model predicts that participants' biased subjective randomness is partially corrected as compared with the Randomization Condition. The MAD and the SD calculated from model predictions in Dyad Condition are much smaller than the ones in the Randomization Condition (Table 1 and 2).

Single Condition In the Single Condition, the formulae for matchers and non-matchers are the same. Therefore, we collapse the two cases. We find that the overall performance

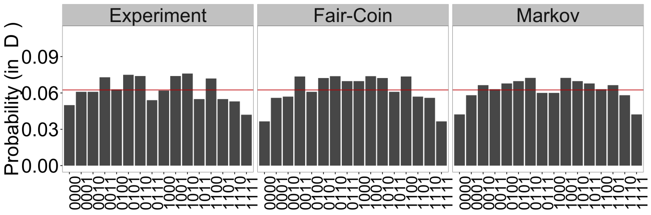


Figure 3: Observed distributions of 4-tuples and model predictions for the Dyad Condition. Plotting conventions are the same as for Figure 2.

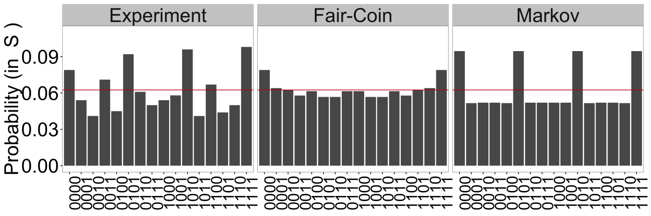


Figure 4: Observed distributions of 4-tuples and model predictions for the Single Condition. Plotting conventions are the same as for Figure 2.

of the model seems to be fine (Figure 1, Right), $R^2 = .94$, though it falls short of capturing the empirical observation that the probability of generating heterogeneous sequences like [0 1 0 1] is high.

The Markov Model

As seen from the model fitting results above, the Fair-Coin Model leaves room for improvement in modeling the Single Condition. Notice that in the empirical data, both sequences that are more representative of the outcome of tossing an unbiased coin (e.g., [0 1 0 1]) and the less representative ones (e.g., [0 0 0 0]) have high probabilities. It is impossible for the Fair-Coin Model to reproduce this effect. We thus explore the possibility that participants adopt a different definition of random sequences (e.g., a Markov process with transition rate of 0.5). This hypothesis has the potential to explain the data in the Single condition because in this case a representative "non-random" sequence would be the outcome of a Markov process with transition rate other than 0.5, thus including too many or too few alternations in the sequences. Indeed, this is what Rapoport and Budescu found. We next explain how this model simulates each condition in the Rapoport and Budescu experiment.

Model Description

Randomization Condition Similar to the Fair-Coin model, the Markov Model assumes that participants try to convince the experimenter that the sequences they give are generated by a random process. However, their notion of a "a random process" is not "tossing an unbiased coin", but rather a generative process with a transition probability $P(R_k \neq R_{k-1})$ equal to 0.5. If there is some bias in the generating process, the transition probability should be less than 0.5, resulting in sequences with fewer alternations, e.g., [0 0 0 0] and [1 1 1 1].[2] As in the Fair-Coin Model, we denote a sequence of length k to be S_k, and a specific instance of it to be s_k ($k = 2,3,4$). Using Bayes' Rule, we obtain the posterior probability of $P(S_k = s_k)$ when participants aim to show the experimenter that the transition probability of the underlying generative process equals to 0.5.

Dyad Condition For the Dyad Condition, the Markov model is very similar to the Fair-Coin Model. It assumes that participants believe that their opponent intends to simulate the outcome of a generative process with transition probability of 0.5. Therefore, they use their opponent's previous choices to predict their opponent's current choice and make a decision according to their prescribed role. At the same time, they are motivated to generate subjectively random sequences so that their own responses are less predictable. Hence, they will try hard to simulate the outcome of a generative process with transition probability of 0.5. Mathematically, the probability of choosing 0 given the previous responses can be calculated using the same equation (1), but now $P(S_k = s_k)$ is obtained from the Markov Model for the Randomization Condition rather than the Fair-Coin Model for the Randomization Condition. We then use Equation (3) and (4) to calculate the probability of choosing 0 as a matcher or a non-matcher conditioned on their own and their opponents' previous responses. w was set to be 0.7 in the Markov Model; since the Markov Model and the Fair-Coin Model have different assumptions, there is no reason that the weight w assigned to the predicted opponents' responses (the Theory of Mind reasoning component) should be equal in these two models.

Single Condition Similar to the Fair-Coin Model, for the Single Condition, the Markov model assumes that participants generate responses based on their own previous choices. Particularly, they know that their opponents think that they intend to generate random sequences. Therefore, in each trial they would make a response so that the opposite of it combined with his/her previous responses will look like an outcome of a Markov process with transition rate equal to 0.5 (see the online code for more detail).

Results of the Markov Model

Randomization Condition Figure 5, Left, shows that the model fits the data well, $R^2 = .95$. Figure 2, Right shows that the model successfully captures the observation that more heterogeneous tuples such as [0 1 1 0] are more likely to be generated than less heterogeneous tuples such as [0 0 0 0]; still, the Markov Model was not statistically significantly

[2] Note that in the Randomization Condition and the Dyad Condition we use this asymmetric prior, whereas in the Single Condition we use a symmetric prior, assuming that a non-random sequence would have either a large transition rate (0.75) or a small transition rate (0.25).

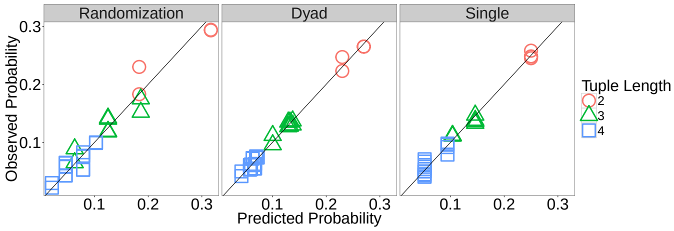


Figure 5: Observed frequencies of k-tuples vs. predicted probabilities of the the Markov Model.

better than the Fair-Coin Model in this condition.

Dyad Condition Although in the Dyad Condition we have different formulae for matchers and non-matchers, the simulation results show that the distributions of k-tuples are the same. Therefore, we collapsed these two cases (Figure 3, Right). The model predictions are well aligned with the empirical data (Figure 5 Middle), $R^2 = .99$. Critically, it predicts that participants' biased subjective randomness is partially calibrated (see the MAD and the SD section).

Single Condition The model fits the data well (Figure 5, Right), $R^2 = .98$. Critically, it captures the findings that both sequences that are more representative of the outcome of tossing an unbiased coin (e.g., [0 1 0 1]) and those that are less representative (e.g., [0 0 0 0]) have higher probabilities. The model works because it assumes that participants avoid producing sequences that are representative of the outcome of a Markov process with transition rate 0.5; thus, participants end up generating sequences with either too many alternations (transition rate larger than 0.5) or too few alternations (transition rate less than 0.5).

Model Comparison From the correlation plot we see that Markov Model seems to fit the data better than the Fair-Coin Model. However, since these two models have different assumptions, traditional statistical tests for model comparison are not applicable. Therefore, we use the R package "cocor" which allows us to directly compare the correlations between the empirical data and the model predictions of these two models (Diedenhofen & Musch, 2014). The results showed that the difference between the two correlations r_{FC} and r_M in the Dyad condition is not significant, Dunn and Clark's $z = -1.49$, $p = .136$. It is also not significant in the Randomization condition, $z = -0.54$, $p = .589$. However, in the Single Condition, the correlation r_{FC} is significantly smaller than the r_M, $z = -3.29$, $p = .001$. Overall the difference between the two correlations r_{FC} and r_M is significant, $z = -2.47$, $p = .014$. In other words, the Markov Model provides a better fit to the data than the Fair-Coin Model. Therefore, in the following section, we only present the MAD and SD calculated from the predictions of the Markov Model.

MAD and SD Consistent with the data, the model predicts the same qualitative results for the MAD and the SD of the three conditions (Table 1 and 2), i.e., the Randomization Condition has the largest deviation from what would be expected with truly random sequences and the Dyad Condition has the smallest one. This suggests that the Markov Model successfully captures the more calibrated subjective randomness in the Dyad Condition.

Table 1: Mean absolute deviation (MAD) of the data and the predictions of the Markov Model. D: Dyad, S: Single, R: Randomization.

	Data			Model		
	D	S	R	D	S	R
2 − tuple	0.0150	0.0043	0.0435	0.0200	0.0000	0.0667
3 − tuple	0.0105	0.0135	0.0273	0.0123	0.0211	0.0309
4 − tuple	0.0087	0.0159	0.0174	0.0068	0.0160	0.0224

Table 2: Standard deviation (SD) around expectations of the data and the predictions of the Markov Model. D: Dyad, S: Single, R: Randomization.

	Data			Model		
	D	S	R	D	S	R
2 − tuple	0.0199	0.0057	0.0534	0.0231	0.0000	0.0770
3 − tuple	0.0141	0.0150	0.0353	0.0155	0.0226	0.0465
4 − tuple	0.0105	0.0194	0.0222	0.0092	0.0191	0.0257

Discussion

Empirical evidence suggests that people generate more truly random sequences in competitive contexts. We explored two probabilistic models to explain the calibrated subjected randomness in a competitive game that was reported in Rapoport and Budescu (1992). We find that Theory of Mind models based on both the Fair-Coin and the Markov formalizations of subjective randomness are able to capture the calibrated subject randomness effects that appear in an iterated competitive game (Dyad Condition vs. Randomization Condition). However, the Markov Model is better than the Fair-Coin Model in explaining the intermediate degree of calibrated subjective randomness that appears in a competitive game where participants must specify their choices ahead of time (the Single Condition).

Why is the Markov Model better than the Fair-Coin Model in simulating the Single Condition? The reason might be that the transition probability of a generative process is more cognitively accessible than "the weight of a coin". When people attempt to generate random sequences, it may be easier to track the transition probability and make sure it approximates 0.5 than to check whether one of the binary responses is made more often than the other. In short, transition probability might be a more convenient heuristic than "the weight of the coin" in evaluating the randomness of sequences.

In addition, it is worth noting that the Markov Model and the Fair-Coin model share the common Theory of Mind reasoning structure. The only difference between these two is the assumption on how people define "random sequences". Kubovy and Gilden (1991) showed that participants attend to multiple numerical properties of the sequence, such as number of alternations, lengths of runs, and imbalance between 0 and 1. The Fair-Coin model focuses on the imbalance between 0 and 1, and the Markov Model focuses on number of alternations and lengths of runs. The results suggest that when online feedback is not available, participants are more likely to rely on number of alternations and lengths of runs to produce unpredictable sequences.

We note some limitations of these models. Both the Fair-Coin Model and the Markov Model assume that participants are probability matching rational agents and generate binary responses in proportion to the interpreted randomness. Therefore, one limitation is that the models cannot predict a player's behavior when his/her opponent does not use the optimal strategy. For example, if matcher "A" plays with a person who chooses "0" more often than "1", A would quickly notice it and choose "0" more (if not always). However, the two models in the current study would not make such predictions because of the assumption that the other agent intends to generate random sequences. Hence, a more complete model may retain uncertainty as to what kind of opponent the participant is playing with. This may also be formalized using a the reinforcement learning algorithm (Lee et al., 2004), and it is worth comparing the assumptions and predictions of the current probabilistic approach with previous reinforcement learning approaches. Another limitation is that we do not explicitly manipulate the number of previous trials the models consider and compare the corresponding performances. However, post-hoc analysis indicates that in both the Fair-Coin model and the Markov model, taking more previous trials into account results in better calibrated subjective randomness, which is consistent with the results of previous connectionist modeling that manipulates the working memory capacity of the models (West & Lebiere, 2001).

In summary, the current investigations suggest Theory of Mind reasoning interacts with participants internal models of subjective randomness in the generation of random sequences in competitive contexts. Future computational approaches should take this into account when modeling subjective randomness.

Acknowledgements

We would like to thank Robert X.D. Hawkins for useful discussions and Noah D. Goodman for his support in teaching Psych 204: Computation and Cognition.

References

Ayton, P., Hunt, A. J., & Wright, G. (1989). Psychological conceptions of randomness. *Journal of Behavioral Decision Making*, *2*(4), 221–238.

Bakan, P. (1960). Response-tendencies in attempts to generate random binary series. *The American Journal of Psychology*, *73*(1), 127.

Diedenhofen, B., & Musch, J. (2014). cocor: A comprehensive solution for the statistical comparison of correlations. *PLOS ONE*, *10*(3), e0121945–e0121945.

Goodman, N. D., & Stuhlmüller, A. (2014). *The Design and Implementation of Probabilistic Programming Languages.* `http://dippl.org`.

Griffiths, T. L., & Tenenbaum, J. B. (2001). Randomness and coincidences: Reconciling intuition and probability theory. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society* (pp. 370–375).

Griffiths, T. L., & Tenenbaum, J. B. (2003). From algorithmic to subjective randomness. In *Advances in Neural Information Processing Systems* (pp. 953–960).

Hahn, U., & Warren, P. A. (2009). Perceptions of randomness: Why three heads are better than four. *Psychological Review*, *116*(2), 454–461.

Kahneman, D., & Tversky, A. (1974). Subjective probability: A judgment of representativeness. In *The concept of probability in psychological experiments* (pp. 25–48). Springer.

Kubovy, M., & Gilden, D. (1991). Apparent randomness is not always the complement of apparent order. In *The perception of structure: Essays in honor of Wendell R. Garner.* (pp. 115–127). American Psychological Association.

Lee, D., Conroy, M. L., McGreevy, B. P., & Barraclough, D. J. (2004). Reinforcement learning and decision making in monkeys during a competitive game. *Cognitive Brain Research*, *22*(1), 45–58.

Lopes, L. L., & Oden, G. C. (1987). Distinguishing between random and nonrandom events. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*(3), 392–400.

Neuringer, A. (1986). Can people behave "randomly?": The role of feedback. *Journal of Experimental Psychology: General*, *115*(1), 62–75.

Rapoport, A., & Budescu, D. V. (1992). Generation of random series in two-person strictly competitive games. *Journal of Experimental Psychology: General*, *121*(3), 352–363.

Shafto, P., Goodman, N. D., & Frank, M. C. (2012). Learning from others the consequences of psychological reasoning for human learning. *Perspectives on Psychological Science*, *7*(4), 341–351.

West, R. L., & Lebiere, C. (2001). Simple games as dynamic, coupled systems: randomness and other emergent properties. *Cognitive Systems Research*, *1*(4), 221–239.

Applying Primitive Elements Theory for Procedural Transfer in Soar

Bryan Stearns (stearns@umich.edu)
John Laird (laird@umich.edu)
Mazin Assanie (mazina@umich.edu)
University of Michigan, 2260 Hayward Street
Ann Arbor, MI 48109-2121 USA

Abstract

Detailed transfer of procedural knowledge has been modeled in Actransfer, an extension of ACT-R, by combining the primitive memory operations of productions (PRIMs) with the architecture's procedural learning mechanism (Taatgen, 2013c). This work explores whether these same principles can be applied to the Soar cognitive architecture, which uses different models of working memory and procedural learning. We confirm that these principles can transfer to an unmodified version of Soar. Our research contributes a novel model of skill learning based upon a deeper level of primitive skill composition than described in the PRIM model that is suitable for unbounded working memory architectures, and which yields transfer profiles similar to those revealed in human studies.

Keywords: cognitive transfer; skill acquisition; cognitive training; cognitive architecture; ACT-R; Soar.

Introduction

For decades, cognitive architectures (Newell, 1990) have been proposed as unified theories for achieving the general capabilities found in the human mind. Transfer of procedural knowledge is one such capability (Taylor, Kuhlmann, & Stone, 2008). The primitive elements theory of cognitive skills, proposed by Niels Taatgen, has recently achieved success in improved modeling of human transfer (Taatgen, 2013a, 2013c). Taatgen implemented his theory in a new cognitive architecture, Actransfer, which extends the ACT-R cognitive architecture (Anderson, 2007) to include this transfer modeling. Taatgen noted that, while his implementation was based upon ACT-R principles, the core ideas could be applied to other theories of cognition (Taatgen, 2013b). The work described here pursues this line of research in the Soar cognitive architecture (Laird, 2012), and briefly compares primitive elements learning in Soar with that of Actransfer and humans performing a common task.

A significant contribution of this work is that we extend Taatgen's theory to include a more general level of learning that supports unbounded, dynamic architectural memory structures and that provides a deeper model of the nature of skill composition.

The following sections first describe Actransfer and the underlying PRIM model before introducing the PROP model, the novel application of these ideas in Soar.

Background

The identical elements model of learning by Thorndike (1922) states that transfer among tasks occurs only inasmuch as there are identical cognitive elements shared in task representation and execution. Singley and Anderson (1987) proposed a more precise definition through the identical productions model, in which complex cognition is controlled by procedural knowledge represented as *if-then* production rules. This representation allows transfer to the extent that different tasks share identical productions.

Singley and Anderson (1985) evaluated the identical productions model using ACT, a precursor to ACT-R. By comparing the model with human performance, they found that in some cases it produced a fairly accurate relative prediction of human data. In other cases, only half the transfer measured in human participants was achieved, indicating that the model was incomplete.

Primitive Elements Theory & PRIMs

Taatgen proposed the primitive elements theory (Taatgen, 2013c) as a modification of the identical productions model of transfer. There are two aspects to the theory. First, primitive elements of transfer are defined not as complete, task-specific productions, but as the individual task-general memory operations used in productions, such as the general action of copying a value from one memory slot to another. Second, the theory outlines a model of human skill acquisition and transfer based on hierarchically composing these primitive operations through practice into task-specific rules, using a procedural learning mechanism. Composed skills will share identical elements if they employ similar memory operations.

Actransfer, the implementation of these ideas, was applied to human transfer experiments (Chein & Morrison, 2010; Elio, 1986; Singley & Anderson, 1987), achieving results that both align with human data and provide deeper theoretical explanations for transfer than earlier models.

Primitive elements theory implemented in Actransfer represents what Taatgen called the ***PRIMitive information processing element*** (PRIM) model. In this model, PRIMs are the fundamental, innate memory operations that can be composed through practice into any skill. Compositions of PRIM sequences are transferable when shared among rules.

Conditions	Action	Other
Compare Equal Compare Unequal Empty Nonempty	Copy	Load Task-Specifics

Table 1: The six types of PRIMs. Loading task-specifics is a PRIM that loads values into memory slots for use by other operations.

```
(PRIM instruction-example
  retrieval.slot2 <> goal.slot2
  retrieval.slot2 <> nil
==>
  query.slot1 := consts.slot1
  query.slot2 := retrieval.slot2
  action.slot1 := consts.slot2
  action.slot2 := retrieval.slot2
)
```

Figure 1: Example pseudo-code primitives of a rule with two compare conditions and four copy actions. Slots are organized under buffers such as `retrieval` or `action`.

Memory Operations

Actransfer working memory is composed of a set of buffers, each having a fixed number of memory slots. All Actransfer skills (rules) are represented as sequences of six types of memory operations, the classes of PRIMs listed in Table 1. This restricted set of compare and copy actions was chosen to reflect previous ACT-R work in basal ganglia modeling (Stocco, Lebiere, & Anderson, 2010).

While there are six PRIM types, there can be many instances of each corresponding to interactions among different memory slots. This is analogous to how computer programs use a finite set of register operations in an assembly language, such as ADD or LOAD, but can apply these among registers in many ways. In Figure 1, while each action is a copy operation, each is a different primitive because it copies using different slots. When Actransfer was configured with 31 working memory slots, this resulted in 1,693 PRIMs for the combinations of these slots with each type of operation (Taatgen, 2013c). Different rules share PRIMs if the same slot operations are used (regardless of the values in those slots).

Independence from values in memory slots makes a PRIM task-general. Instead of using conditions such as `buffer1.slot1 == "foo"`, the architecture preloads constants into a reserved set of slots, effectively making the condition `buffer1.slot1 == consts.slot1`. Thus, a different rule using different constants, such as `buffer1.slot1 == "bar"`, still employs the same primitive. This generalization allowed much of Taatgen's novel transfer across tasks.

Procedural Learning

An Actransfer agent learns a skill by rehearsing it step-by-step according to declarative knowledge recalled from long-term memory. Practice is gradually converted into procedural knowledge. These declarative instructions describe rules as sequences of PRIMs applied with specific constants, as shown in the bottom row of Figure 2. When the agent lacks applicable procedural knowledge, it recalls a list of instructions that describe a single rule, and then sequentially evaluates each instructed condition and action.

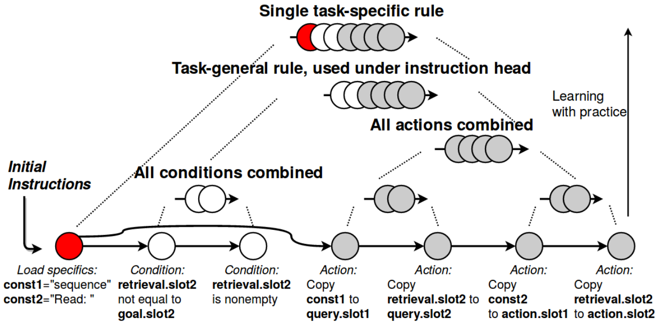

Single task-specific rule
Task-general rule, used under instruction head
Learning with practice
All actions combined
Initial Instructions
All conditions combined
Load specifics:
const1="sequence"
const2="Read: "
Condition:
retrieval.slot2 not equal to goal.slot2
Condition:
retrieval.slot2 is nonempty
Action:
Copy const1 to query.slot1
Action:
Copy retrieval.slot2 to query.slot2
Action:
Copy const2 to action.slot1
Action:
Copy retrieval.slot2 to action.slot2

Figure 2: Hierarchical clustering of PRIMs, adapted from Taatgen (2013c). Task-general conditions are shown in white, task-general actions are in gray, and the instruction head, which includes loading task-specific constants, is shown in red. All instructions begin with loading task-specific constants into memory. ACT-R production compilation combines repeated sequences of task-general condition and action PRIMs, until finally merging with task-specific constants, resulting in a single production. In this example, actions form a query to retrieve the next item in a sequence from declarative memory, while printing the current number to output.

Actransfer employs ACT-R's procedure compilation system to transform this practice into skill. Each PRIM instance is implemented as an innate rule in procedural memory. Each Actransfer decision corresponds to firing a single rule. Whenever two different rules are fired in consecutive decision cycles, the architecture attempts to combine them into a new rule that can perform the work of both in a single decision. Such rules are not used initially the next time the same operations are practiced, but the more the original rules are practiced in sequence, the more likely it is that the combining rule will be used in their place. Once this replacement occurs, the new rule fires alongside other instructed rules, and the combination process repeats. As skills are practiced in this manner, procedure compilation learns an effective binary hierarchical clustering of skill elements, as shown in Figure 2. Compiled operations perform instructions in parallel rather than serially, decreasing execution time with practice and clustering. The final step of learning incorporates any task-specific constants into a single generated rule. With enough practice, all instructions are converted into such procedural knowledge, so that instruction recall becomes unnecessary.

Any intermediate compilations between the original PRIM sequence and the complete task-specific rule can be used for transfer, as the time to learn a new rule is less when portions of its instructions have already been compiled. The PRIM model thus predicts improved performance with repeated practice based both on incremental composition of operations and on reuse of such compositions.

PROPs - Primitive Skill Elements in Soar

Soar's working memory is not a fixed set of slots, but is an unbounded directed cyclic semantic graph rooted in a *state* ID, as in Figure 3, with each possible attribute path through the graph referencing a unique memory element. It would seem impossible to use primitive elements in Soar, since an unbounded set of memory locations would define infinitely many PRIMs. The solution to applying these concepts to Soar lies in recognizing that Soar's information processing

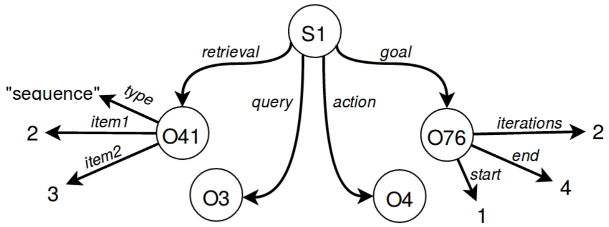


Figure 3: Example Soar working memory. Working memory is a directed graph rooted in a single state ID (shown as S1). Nodes can have any number of edges each pointing to a single value. Values can be more IDs or literals such as numbers or strings.

also differs from that of Actransfer.

Operators

A Soar decision cycle corresponds to the selection of a single *operator*, an architectural construct created, selected, and applied by rules to guide decision making. Selection of a single operator can involve firing several rules in parallel and/or serial, and rules can use wildcard variables to match a potentially dynamic state and shape of working memory.

An important distinction exists between an operator and its abstract definition. The set of rules describing an operator form the abstract definition, and an individual operator is the application of this definition in a decision to specific memory elements. Because Soar rules use variables, an abstract definition can be applied through an unbounded number of operators corresponding with the possible applications of the rule logic to states of the unbounded working memory graph.

We define the set of primitive memory operations within Soar decisions as the *PRimitive OPerator processing elements* (PROPs) that correspond to the most primitive operators from which all Soar agent processing can be composed. PROPs are defined through a fixed set of innate rules corresponding to each type of PROP listed in Table 2, which in turn correspond to the basic conditions and actions usable in Soar rules. While all Soar information processing can be composed from operators of these 22 abstract types, the number of possible PROPs is unbounded, since each corresponds to specific memory elements. For example, in Figure 4 there are four copy action operators, and these would be implemented through the same defining rules, but they remain distinct operators because they use different memory elements.

PROPs can only be applied if given specific memory references as arguments. The *memory referencing* PROP is what allows true support for unbounded memory by tracing a declaratively-known path through the memory graph and supplying the located element as an argument to another operator.[1] The number of memory references required to define each type of PROP is also shown in Table 2.

PROPs correspond to PRIMs as the primitive elements of decision making in Soar, but unlike PRIMs they are not innate. Initially, memory references must be reconstructed to redefine a PROP any time it is used. However, with practice, procedural knowledge to use memory elements is learned (see below), providing rules similar to the innate PRIMs of Actransfer. Thus, primitive memory access skills are based upon references actually used by the agent rather than the space of all possible memory operations. PRIMs in Actransfer may be considered a special case of the PROP model in which the working memory graph elements are arranged to match Actransfer slots and the agent is already trained in their use.

Conditions	Actions	Preference Actions	Other
Equal (2)	Copy (2)	Acceptable(1)	*Memory referencing*
Unequal (2)	Remove (1)	Indifferent (1)	
Exists (1)	Add ID (1)	Better (2)	
Not Exists (1)		Worse (2)	
Type Equal (2)		Best (1)	
Greater (2)		Worst (1)	
Greater/Equal (2)		Reject (1)	
Less (2)		Require (1)	
Less/Equal (2)		Prohibit (1)	

Table 2: Table of PROP types. In parentheses after each PROP is the number of memory element arguments required to apply the abstract definition into an operator.

```
pp {PROP-instruction-example
  (s1.retrieval.item2 <> s1.goal.end)
  (s1.retrieval.item2)
-->
  (s1.query.type := s1.consts.slot1)
  (s1.query.item1 := s1.retrieval.item2)
  (s1.action.out1 := s1.consts.slot2)
  (s1.action.out2 := s1.retrieval.item2)
}
```

Figure 4: PROPS instruction logic mirroring Figure 2. Because each primitive is self-contained, the full path from S1 must be specified for each working memory element.

Procedural Learning

Soar's procedural learning mechanism also creates rules by combining the results of decision cycles, and can be used to compose skill elements hierarchically as in Figure 2. However, Soar does not compose rules that subsume pairs of sequential decisions, but instead summarizes any number of decisions and rules that are used to resolve a subgoal impasse.

An impasse is an event that arises when normal decision making cannot proceed, such as when no operator is available for selection or no procedural knowledge carries out a selected operator. When an impasse arises, a new substate is automatically created in working memory. Operators are selected in the substate to resolve the impasse. When the results of this work allow processing in the original state to resume, Soar automatically creates rules that summarize the rule firings and decision making that led to resolving the impasse. This learning process is called *chunking*. In similar future sit-

[1] This includes referencing any task-specific constants, and thus there is no separate primitive for that operation.

uations, the learned rules (*chunks*[2]) fire to avoid the impasse, replacing the substate processing. Soar chunks do not take effect gradually with practice as do ACT-R rule compositions, but fire whenever their conditions are met.

The PROP model is implemented through standard Soar rules that can be loaded into any Soar agent. As with Actransfer, declarative instructions describing the skills being taught are initially loaded into the agent's long-term memory, and are rehearsed during agent operation whenever the agent has no known operators to select. However, where Actransfer employed an architectural modification to automatically recall instructions when no decisions could be made, this behavior comes naturally in Soar through agent reactions to impasses. A Soar agent is also not restricted to only respond to an impasse with instruction practice, but could choose from available strategies according to the situation at hand. The PROPs agent by default begins instruction practice by recalling and following instructions within the new substate. Further impasses during instruction evaluation allow the agent to compile pairs of instructed procedures through chunking.

The amount of practice taken by a PROPs or Actransfer agent before compiling procedures into higher skills can affect whether those skills transfer across tasks (Anderson, 1982). Consider two rules, one composed of primitives A, B, C and the other of B, C, D. Ideally, the skill (BC) is learned that reduces training time for both rules, as opposed to (AB) and (CD), which cannot be shared. Actransfer does not replace element pairs with their combination until experience determines that they co-occur often across tasks. Once a combination replaces the original components, use of that rule prevents the architecture from further sampling co-occurrence of the old component rules in that context.

For Soar to combine skills based on co-occurrence, a declarative representation of experience is used to mediate the chunking process. The skill hierarchy of Figure 2, which implicitly reflects the ACT-R learning approach, is represented literally in the PROPs agent's long-term declarative memory, along with declarative measures of how often instruction items are experienced together. When two elements in this hierarchy co-occur beyond some threshold, T, the agent chunks them into a new skill element.[3] This co-occurrence reasoning is not integral to the PROP model, and would be unnecessary if Soar defined gradual confidence-based chunking.

Levels of Skill Composition

Through chunking, a PROPs agent learns three different types of knowledge that vary in complexity and provide speedup in different ways.

The first level of learning is of the use of memory through practice in applying PROPs from their abstract definitions, and is unique to this model. This processing is chunked into procedural knowledge when PROPs are compiled into new skill elements (the first level of composition in Figure 2).

The next type of learning is the normal hierarchical skill compilation, which is also the core of learning in Actransfer. Gradually improved performance comes from repeatedly composing instructed decisions into fewer, more task-specific rules through chunking, and from transferring such knowledge from previous compositions.

The third, outer-most level of learning is that which achieves independence from declarative instruction look-up. As with Actransfer, once general conditions and actions in an instruction set are merged as far as possible, a final learning step summarizes the instruction set into a task-specific rule that is executed when needed without fetching or evaluating instructions (the final stage shown in Figure 2).

By comparing these stages with the corresponding mechanisms of Actransfer, we can predict that a PROPs agent performing level-one learning should have a steeper initial performance curve as it learns to use its memory references. However, the main learning profiles of the architectures should be similar, including the amounts of transfer they provide, since they share the same core level-two learning process. We can also predict that the more aggressive nature of Soar chunking compared to gradual ACT-R rule compilation should result in the PROPs agent displaying a slightly more discrete and complete independence from instructions upon completion of its third level of learning.

Evaluation

Testing these predictions, we gave a PROPs agent declarative instructions to perform in a simulation of the Elio (1986) study that measured human transfer. Instruction logic and memory organization copied an Actransfer simulation by Taatgen (2013c), so that both model implementations learned to compose equivalent sequences of memory processing.

We ran two experiments. First, we tasked the agent to learn from scratch all three levels of knowledge described. Then, we repeated the experiment, but bypassed level-one learning by initially supplying the agent with all procedural knowledge necessary for memory referencing, mirroring Actransfer's use of innate PRIMs. We varied the learning threshold T, the number of times two skill elements must be seen together to be chunked into a new skill, and found values in the range of 16 to 24 provide comparable behavior to human and Actransfer results. $T = 16$ is used in data shown below.

Because the PROP model is implemented through rules rather than architectural modifications, maintaining instruction co-occurrence knowledge requires agent decision cycles, and this manifests as performance overhead during training. For a better comparison of the models, this overhead is omitted from this evaluation, as it is a reflection of implementation rather than part of the theory. All data are averaged over 8 samples, as in Taatgen's originally reported results.

Actransfer performance was originally reported in simulated time, but decision cycles are shown here to allow a more meaningful comparison across architectures. [4]

[2] Unrelated to ACT-R *chunks*.

[3] Different metrics can be easily substituted for linear co-occurrence, but this simple measure works sufficiently well here.

[4] Actransfer follows ACT-R in assuming 50 ms per decision by

Step	Calculation	Operation Type
Particulate rating	$\text{Solid} \times (\text{lime}_4 - \text{lime}_2)$	Component
Mineral rating	greater of $(\text{algea}/2)(\text{solid}/3)$	Component
Index 1	$\text{Particulate} + \text{Mineral}$	Integrative
Marine hazard	$(\text{toxin}_{\text{max}} + \text{toxin}_{\text{min}})/2$	Component
Index 2	$\text{Index1}/\text{Marine}$	Integrative
Overall quality	$\text{Index2} - \text{Mineral}$	Integrative

(a) Example procedure. Component steps only reference inputs. Integrative steps require remembering results of previous calculations.

SOLID	ALGAE	LIME	TOXIN
6	2	3 5 1 9	4 8 7 2

(b) Participants look up hypothetical water sample data from among ten values provided per trial. For lime or toxin values, procedure instructions either specify the row index to look up or instruct to find the max or min value.

Figure 5: The Elio task

The Elio task involved calculating hypothetical pollution rates based on water samples. Subjects repeatedly performed mental calculations using given input values. In the human study, subjects were trained in an initial procedure until they achieved perfect recall, and then tasked with performing it 50 times on various inputs (see Figure 5). Following this, subjects were assigned to 50 trials of one of three transfer conditions: transferred integrative, transferred component, and a control. The first two of these shared different types of calculations with the training, but the control did not. A basic ACT-like identical productions model would predict transfer from the training procedure to procedures that shared calculations, but would not predict transfer to the control. Yet transfer to the control was evident in the human results, as shown in Figure 6 through the faster initial performance of the transfer tasks compared to the training. Elio's transfer condition data measure the mean performances from the first and last 25 trials per subject. Depicted human training data shows Elio's power-law fit to human performance. In the original study, results for component and integrative calculations were reported separately. Only performance on component steps is shown here for brevity, as integrative results are comparable.

Control transfer is also reflected in the Actransfer agent. The *transferred component* procedure shows much additional transfer as well. This is because it shares component calculations with the training, allowing classic identical productions transfer in addition to primitive skill composition transfer.

We first ran our PROPs agent on the Elio task performing full learning of knowledge levels one through three. We then ran the agent with level-one procedural knowledge predefined for all relevant *memory referencing* PROPs. Performance for both experiments is shown in Figure 7.

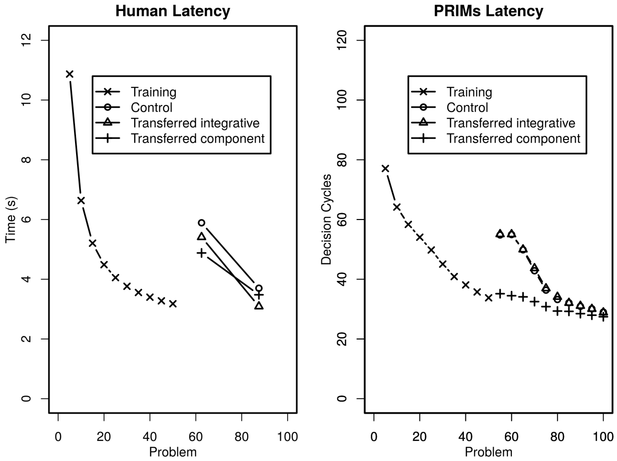


Figure 6: Human and Actransfer performance for component steps. Data for problems 1-50 show training performance. Data for problems 50-100 show performance for each of the three transfer conditions. Actransfer data were generated using supplementary materials from Taatgen (2013c).

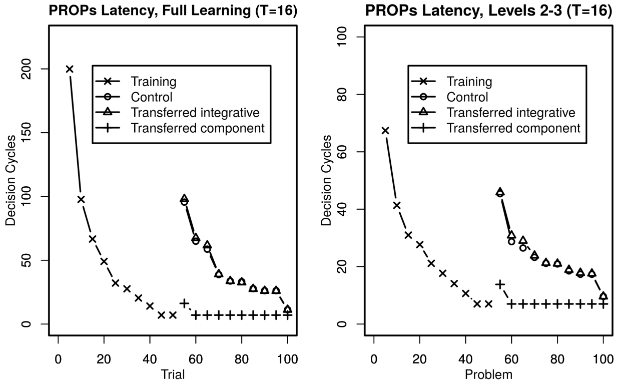


Figure 7: PROPs agent performance of the Elio task, measured in decision cycles. Hierarchy management overheads are not included. Left: Learning all levels. Right: Learning with predefined PROPs.

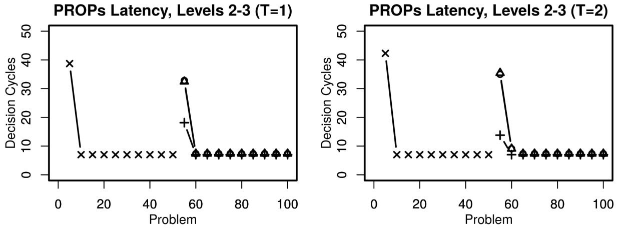


Figure 8: The progression from $T = 1$ to $T = 2$.

Four results stand out. First, as expected, the initial transfer condition performances in the PROP model indicate the same rates of transfer as in the Actransfer agent. Second, unlike Actransfer, in both experiments the PROPs agent performance sharply converges to maximal performance, which is just under ten decision cycles. This is due to the discrete nature of Soar chunking, particularly with level-three learning, when independence from instructions is permanently achieved in a single chunking step. Third, the PROPs agent that only performs level-two and level-three learning roughly

default, with additional time for operations such as long-term memory retrievals. We similarly assume 50 ms per Soar decision.

shows the same power-law performance as Actransfer, as is expected since they perform similar learning processes. Finally, one notes that if modeling simulated time using 50 ms per decision cycle, the PROPs agent with full learning of levels one through three performs at similar simulated time scales to the human model, beginning at 10 s, with the exception of then converging to maximal performance as discussed, ending at about 0.5 s.

Learning threshold T controls the sharpness of the learning curve, as well as the transferability of composed skill elements. A threshold of $T = 1$ causes near-instant skill acquisition, but makes blind combinations that might not be transferable. Through an analysis of the number of chunks transferred across procedures with varying T (not shown), we found that even a threshold as low as $T = 2$ allowed sufficient co-occurrence sampling for achieving near-optimal transfer in this task. Figure 8 demonstrates the improved initial latency of the *transferred component* case, which is not further improved with the higher threshold of Figure 7.

In summary, our experimentation indicates that the PROP model not only provides the same transfer as the PRIM model, but that deeper learning with memory references to suit dynamic memory also aligns with human performance.

Discussion

Primitive elements theory distinguishes among three types of skills: innate, task-general, and task-specific (Taatgen, 2013c). Innate skills are single primitives, task-general skills are combinations of primitives, and task-specific skills are the combinations of general skills with specific constants. These correspond to the three levels of *learning* in the PROP model. We theorize that level-one memory management knowledge would be learned (possibly developmentally) by human subjects prior to participating in the Elio task. Actransfer by contrast assumes a fixed configuration in which memory slots and their use are innate.

Actransfer's fixed set of PRIMs is useful in that they *must* in some respect be shared across any use of the architecture, just as registers must be used in any normal processor logic. In that model, the number of innate PRIMs expands combinatorially with working memory capacity, though only a subset might be used. In the PROP model, however, while transfer likewise depends on using a common set of memory references, PROPs only reflect skill elements used in practice.

ACT-R's and Soar's procedural learning mechanisms differ in many ways, yet provide similar models of learning with practice. However, as Soar chunking does not currently define gradual skill acquisition under uncertainty. To support such learning requires decision-making overheads that are not part of the PROP model, suggesting that architectural support for gradual confidence-based chunking provides a better fit to this sort of learning, and might be worth pursuing in Soar.

We have shown that despite differences between ACT-R and Soar models of working memory and learning, the primitive elements theory can be implemented in both to achieve similar results. In so doing, we introduced the PROP model for information processing in unbounded memory spaces through memory reference learning. The PROP model builds upon Taatgen's original PRIM model to provide a deeper and more general theory for the acquisition of cognitive skills.

Acknowldgments

The work described here was supported by the Office of Naval Research under grant number N00014-15-1-2058. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the ONR or the U.S. Government.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review*, *89*(4), 369 - 406.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.

Chein, J. M., & Morrison, A. B. (2010). Expanding the mind's workspace: Training and transfer effects with a complex working memory span task. *Psychonomic Bulletin & Review*, *17*(2), 193–199.

Elio, R. (1986). Representation of similar well-learned cognitive procedures. *Cognitive Science*, *10*(1), 41 - 73.

Laird, J. E. (2012). *The soar cognitive architecture*. Cambridge, MA: MIT Press.

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

Singley, M. K., & Anderson, J. R. (1985). The transfer of text-editing skill. *International Journal of Man-Machine Studies*, *22*(4), 403 - 423.

Singley, M. K., & Anderson, J. R. (1987). A keystroke analysis of learning and transfer in text editing. *Human-Computer Interacttion*, *3*(3), 223–274.

Stocco, A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: A model of the basal ganglias role in cognitive coordination. *Psychological Review*, *117*(2), 541 - 574.

Taatgen, N. A. (2013a). Diminishing return in transfer: A PRIM model of the Frensch (1991) arithmetic experiment. In *International conference on cognitive modeling.*

Taatgen, N. A. (2013b). *The gap between architecture and model: Strategies for executive control* (Tech. Rep. No. FS-13-03). AAAI.

Taatgen, N. A. (2013c). The nature and transfer of cognitive skills. *Psychological Review*, *120*(3), 439–471.

Taylor, M. E., Kuhlmann, G., & Stone, P. (2008). Transfer learning and intelligence: An argument and approach. In *Proceedings of the first conference on artificial general intelligence (AGI)* (Vol. 171, pp. 326–337). IOS Press.

Thorndike, E. L. (1922). The effect of changed data upon reasoning. *Journal of Experimental Psychology*, *5*(1), 33.

Cognitive Modelling with Term Rewriting

Ivica Milovanović (i.milovanovic@uu.nl)
Utrecht University, Princetonplein 5
3584 CC Utrecht, Netherlands

Johan Jeuring (J.T.Jeuring@uu.nl)
Utrecht University, Princetonplein 5
3584 CC Utrecht, Netherlands

Abstract

Term rewriting is a well established formal method used for defining semantics of programming languages, program transformations, automatic theorem proving, symbolic programming, intelligent tutoring system development etc. In this paper, we present a language based on term rewriting as an alternative formalism for modelling cognitive skills. We show how the language overcomes some deficiencies of production systems (compositionality, readability, control-flow etc.) and how, as a consequence, it can help with addressing practical problems raised by the cognitive modelling community.

Keywords: ACT-R; Language for Cognitive Modelling; Production Systems; Term Rewriting

Introduction

Since the pioneering work of Simon and Newell, production systems have been a dominant formalism for cognitive modelling of human behaviour. A production system is a set of condition-action pairs, called production rules or simply productions. Production conditions test against a collection of facts, commonly called working memory. A typical production system interpreter is a loop that continuously selects and fires a production whose condition matches the current working memory content. Firing a production executes its action which can add, remove or modify working memory facts. If more than one production matches at a time, a conflict resolution strategy is used to select and fire a single one. In addition to cognitive modelling, production systems have been used for implementing Knowledge-Based expert Systems, Intelligent Tutoring Systems (ITSs), Business Rule Engines etc.

ACT-R Cognitive Architecture

The ACT-R cognitive architecture (Anderson, 2007; Anderson, Byrne, Douglass, Lebiere, & Qin, 2004; Taatgen & Lee, 2003) is similar to general production systems. It distinguishes procedural knowledge represented as production rules from declarative knowledge represented as a set of untyped key-value pairs called chunks. Keys are commonly called slots. A distinctive characteristic of ACT-R is that the main goal of its development has been to faithfully reproduce human behaviour. For that reason the architecture introduces a number of constraints not present in general-purpose production rule engines. The architecture is divided into a number of modules, each one specialised for performing specific brain functions. The central procedural module stores and interprets production rules and coordinates the function of all the other modules. The visual, aural, motor and speech modules communicate with the outside world. The goal module (also called control state module), the imaginal module (also called problem state module) and the declarative module are specialised for different aspects of internal cognitive processing. The goal module keeps track of the current state in a task, the imaginal module keeps track of the current internal problem representation and the declarative module is responsible for storing long term declarative knowledge and making it available when needed. The modular nature of the architecture reflects the modular nature of the brain itself, and each module can be mapped to a particular brain region (Anderson, 2007). A module is capable of fast internal parallel processing, but can only communicate with other modules through a buffer that can contain a single chunk at a time. The procedural module does not have its own buffer. It can read buffer content of all the other modules, test the content against LHSs of many productions in parallel, but it can only fire a single production at a time. A cognitive skill is a sequence of productions each of which can test against the buffers and perform actions that directly or indirectly modify the buffers.

Cognition is not a deterministic but rather a stochastic process. Knowledge or lack of a skill is not a binary presence or absence of a given production. It is rather a continuous quantity. For example, a skill may be present, but not trained. All the new skills get better with time. To model that, the ACT-R symbolic system is augmented with a neural-like subsymbolic layer. Each chunk has an activation value which determines whether or not it will be retrieved from declarative memory. If its activation is lower than a certain threshold value, a chunk is not retrieved even if it matches the declarative request. Each production has a utility value that is used in conflict resolution. When multiple productions match, the one with highest utility is selected. Both chunk activations and production utilities are evolving quantities. Activations and utilities start low for newly formed chunks and productions respectively, and get higher with practice.

The Programming Model of ACT-R

An ACT-R program consists of a set of production rules and chunks. As a programming language, ACT-R inherits all the characteristics of the production system paradigm. In isolation, productions can be seen as similar to imperative procedures which modify some global state or to functions that take the state as an argument and return a new modified state as a result. Different from procedures and functions, a production is never called directly in code. A programmer writes rules, and the interpreter decides which rule to fire when. Control flow in production systems is implicit. The production system paradigm is well suited for programming solutions to problems with a large number of independent states and a set of relatively independent actions that can be performed on those states, and not so well suited for programming solutions to problems in well-structured domains, where problem states are related by some formal laws and actions are organised into complex control flows (Buchanan & Shortliffe, 1985). A typical example of the former class are business rule engines, and of the latter class are typical problems that both students and experts solve in STEM fields. Production systems are less suitable for such a domain because of the implicit control flow. As productions can communicate only indirectly through working memory, the only way to establish a desired control flow is to write and read control information to and from that (global) memory. Doing so results in a goto-like coding style which is error-prone and produces code that is difficult to follow, reason about, compose, reuse and maintain. Nevertheless, production systems have often been used for modelling problem solving procedures in STEM fields, particularly in ITSs (Anderson, Corbett, Koedinger, & Pelletier, 1995). The reason for that choice may be partly historical - ITSs emerged from the ACT-R research.

One might argue that implicit control flow in the context of cognitive modelling reflects the flexibility of the brain itself. While that is true, we are investigating the programming aspect of cognitive modelling here. Many published cognitive models[1] are based on complex control flow. This holds in particular for models of learning from instructions or models of metacognitive processes. Both explicit learning and metacognition require complex control flow and planning. The production compilation mechanism (Taatgen & Lee, 2003) eventually produces efficient productions with implicit control flow, but the modeller has to write complex productions with complex control flow. Using ACT-R terminology, production systems become less suitable as a programming model with increased usage of the goal (control state) module. Interestingly, Taatgen (2005) formulated a minimal control principle, which states that humans tend to use a strategy with minimal number of control states while learning a new task. What is more difficult for humans to express as a production system is also more difficult for the brain to execute. Many tasks do require at least some amount of control information and some tasks require complex control. Hence, cognitive modelling can benefit from a programming formalism in which both flexible and complex control can be expressed naturally.

One more difficulty with ACT-R as a programming language, is that modellers have to think explicitly about which modules to use for given actions. Using an appropriate module is important as different combinations of modules for performing the same task can give different predictions of timing, fMRI features etc. Ideally, we would like a modeller to think about task models at a high-level of abstraction and let an interpreter map different actions from a given model to the appropriate modules. An interpreter should know as much as possible about the ACT-R theory and the modeller should focus on the domain-specific features of a particular task.

Heeren and Jeuring have developed a language based on strategic term rewriting (Heeren, Jeuring, & Gerdes, 2010) and used the language to create a number of ITSs and serious games. Besides addressing the software engineering issues of production systems mentioned above, the approach uses well established and rigorous formal methods from formal language theory and practice to analyse and automatically calculate feautures important for intelligent tutoring, such as instructions and feedback. Strategic term rewriting as a computational paradigm allows for writing high-level declarative task models using domain-specific notations. For example, reducing a logic expression to disjunctive normal form is modelled as a set of rewrite rules corresponding to laws of propositional logic (e.g. `not(not(p)) -> p`). A flexible strategy applies all rules extensively until no rule can be applied any more. Such a strategy is similar to strategies used for interpreting production systems. Contrary to production rules, more complex strategies can be expressed naturally. Rewrite rules can be composed to build complex first-class procedures which themselves can be reused in other procedures and composed to build complex hierarchical domains.

Most of the current cognitive architectures do not offer an easy way to reuse knowledge from a model (Taatgen, van Vugt, Borst, & Mehlhorn, 2016). As a result, most published models are isolated theories sharing only common architectural features, and „few cognitive modellers ever use, or even look at, models built by other modellers". Taatgen et al claim that using tools for interactive programming, similar to iPython notebooks, can make cognitive models more accessible and understandable. Term rewriting as a computational paradigm can help addressing all of these problems. Firstly, we can easily compose and reuse terms and rewrite rules, both automatically and by programmers. Secondly, the high-level nature of term rewriting allows for writing cognitive models that are easier to understand (compared to low-level ACT-R models) even by modellers who are not familiar with classical AI programming. Finally, term rewriting is an excellent and natural technique for interactive development.

This paper is organised as follows: we first briefly describe a symbolic language based on term-rewriting. Then we show how the language can be used to develop a cognitive model of the pyramid task (Tenison, Fincham, & Anderson, 2016), and we discuss which characteristics an interpreter for the language should have so that it can give cognitive predictions. We conclude by comparing the introduced approach with related work and giving future work. Although we base our work on ACT-R, the language itself can be used for any cognitive theory by augmenting the interpreter with knowledge of that particular theory.

[1] http://act-r.psy.cmu.edu

Language

The design of our language is inspired by Mathematica (Wolfram Research, Inc., 2017) and Pure[2]. The accompanying example is implemented in Mathematica[3]. Fundamental constructs of the language are expressions (terms) and rewrite rules. For rules of the form `f(e1,e2...)->rhs` we use the terms rule and function interchangeably. Table 1 gives examples of these and other language constructs used in this paper. The interpreter of the language is a higher-order conditional term rewriter with a programmable evaluation strategy. A cognitive model is a set of rewrite rules in the language. A cognitive task is a (potentially complex) term. The interpreter traverses the term following an evaluation strategy and applies rewrite rules to sub-terms. A single traversal step, followed by the application of one or more rewrite rules to a visited sub-term, is roughly equivalent to the three stages in a cognitive task, encoding, solving and responding, as defined in (Tenison et al., 2016). We leave a detailed discussion of the interpreter and evaluation strategies for future work. Fundamental aspects of the interpreter important for cognitive modelling are described in the next section. The model used in this paper is encoded as a simple small term, hence no traversal strategy is necessary.

Model

We use the cognitive model of the pyramid task (Tenison et al., 2016) as our main example in this paper. It requires relatively complex control flow, is expressed in a relatively recent version of ACT-R (6.0), and uses most of the modern ACT-R features. The pyramid task is a simple arithmetic task of the form `base$height` where `base` and `height` are numbers. To evaluate a pyramid, we start from `base` and recursively add `base-1` to it until we reach the total of `height` addends. For example, `8$3` is evaluated as `8+7+6=21`. The (Tenison et al., 2016) model predicts three learning phases, which were corroborated by neural imaging experiments. A learning phase consists of three learning stages - encoding, solving and responding. In the first learning phase there is a long solving stage. With enough practice with a given pyramid, people learn to retrieve the answer for that pyramid without doing the arithmetics, shortening the solving stage significantly. Finally, with even more practice with the same pyramid, people transition to the third learning phase in which they automatically respond without even retrieving the answer. Here we describe the published ACT-R pyramid model informally, using a self-explanatory pseudo-syntax for chunks only occasionally. Rather than trying to follow every technical detail of the model from (Tenison et al., 2016), we use the general theoretical ideas from the model, namely the ACT-R theory of learning by following instructions (Salvucci, 2013; Taatgen & Lee, 2003). We sometimes use slightly more generic productions than those in the published model. For example, instead of having a special production retrieving the sum of two numbers, we assume the existence of more general productions that can retrieve results of any unary, binary or ternary operation respectively. In addition, we use simplified declarative knowledge of arithmetics by treating two-digit numbers as atomic in the same way single-digit numbers are. This is a simplification of a real model, but we want to keep the example model simple. As the entire approach we describe is highly compositional, adding more complicated representation for multi-digit numbers is just a matter of more work. Table 2 gives an informal description of representative parts of the ACT-R pyramid model.

We treat the pyramid task as a formal language with a grammar and semantics. A cognitive model of the pyramid task is an interpreter of that language. To interpret an expression from the language (i.e. to solve a particular pyramid problem), the expression is parsed into an abstract syntax term. After parsing, we apply a series of transformations to the term to obtain a desired solution. Finally, the solution is used somehow, e.g. it is written or pronounced or used as an input to some other sub-problem. The three stages of the interpreting process roughly correspond to the encoding, solving and reporting stages. A typical interpreter of a programming language parses the entire source code into a large abstract syntax tree and performs a series of in-memory traversals and transformations on that tree. Doing the same in interpreters that serve as cognitive models would generally be psychologically unrealistic due to limited capacity of the working memory. Cognitive models are more like sets of small interpreters each of which interprets a small portion of, say, a visual scene (the equivalent of source code). To perform an entire task, small interpreters are composed using a traversal strategy. In case of the pyramid task, however, the entire problem can be parsed and solved in the head. We leave implementing the parsing stage for future work. Here, we assume that the parsing (encoding) stage is already performed, resulting in an abstract syntax term. For example, parsing `8$3` produces the following term:

```
solvePyramid(base->8,height->3) (T1)
```

The above term has a straight-forward interpretation in the context of ACT-R theory - a goal to solve a pyramid is stored in the goal buffer, and base and height are stored in the imaginal buffer. More generally, the contents of the goal and imaginal buffers are represented as a function call, and a list of arguments of that function, respectively. Different ways of achieving a goal are modelled with different definitions of the function representing that goal. The simplest way of achieving a goal is the automatic response to a stimulus which occurs in the third phase of learning. For example, if there is a goal to solve the pyramid with base 8 and height 3, a production (P1) can fire to immediately store the solution of that pyramid in the imaginal buffer. Solving a particular pyramid is represented by a single procedural skill, i.e. a single production specific for that pyramid. Such productions are formed by the production compilation mechanism and strengthened by subsequent practice with the same pyramid.

[2] https://purelang.bitbucket.io

[3] https://github.com/IvicaM/CogEx

Table 1: Examples of Language Constructs Used in the Paper

expression	rewrite rule	anonymous function	subscript notation
`add,subtract,f`	`add(2,3)->2`	`|add($1,$2)|`	get:`pyramid[base]`
`add(2,3),p(b->8,h->3)`	`b->8`	`|add($,3)|`	set:`pyramid[sum->8]`

Table 2: Representative Portion of the ACT-R Pyramid Model

Productions	Facts	Operators
P1: automatically respond to a goal	(add 8 7 15)	O1/O3:(set base/2 sum/count)
P2: retrieve a fact	(decrement 3 2)	O2:(subtract base 1 term)
P3: harvest retrieved fact	(subtract 7 1 6)	O4:(add term sum sum)
P4: retrieve an operator	(increment 2 3)	O7/O8:(decrement/increment term/count)
P5: interpret retrieved operator		O5:(say sum); O6:iterate or respond

In phase 2 of learning, when there is no specialised production for a given pyramid, a solution for that pyramid is retrieved from declarative memory. In that case, two productions, (P2) and (P3) fire, one which sends a declarative request and one which harvests it. Different from specific productions from phase 3, productions from phase 2 are general and can retrieve and harvest any declarative fact. In phase 1 of learning, retrieval fails as the given pyramid fact does not yet have sufficient activation, and the pyramid task is solved step by step by following instructions. We now show how we model all three phases using functions defined as rewrite rules. As automatic procedural skills and declarative knowledge are fundamental and the simplest building blocks in ACT-R theory necessary to model learning from instructions, we start from phase 3 and gradually move towards phase 1.

Rewrite rules that transform a pyramid to its solution represent procedural knowledge used in phase 3. For example:

`solvePyramid(base->8,height->3)->21` (R1)

Analogously to ACT-R production rules, rewrite rules such as (R1) are stored in procedural memory and have utility values. Given a term, such as (T1), the interpreter matches it against LHSs of all the rules in procedural memory. A matching rule transforms the term. If multiple rules match, the one with highest utility value is selected. As with production firing, it takes 50ms to apply a rewrite rule stored in procedural memory. Up to this point, the main difference of the approach relative to ACT-R is syntactic - rewrite rules are much like overloaded functions from ordinary programming languages. Contrary to ordinary interpreters, which are deterministic and usually select the most specific overload of a function, our interpreter selects an appropriate overload according to the assumptions of the ACT-R theory. We show that using functions instead of productions that operate on buffers with chunks allows us to write cognitive models in a compositional way.

To model phase 2, we represent declarative knowledge. In ACT-R, chunks represent declarative facts, e.g. `(solvePyramid 8 3 21)` (F1). Fact (F1) and its procedural counterpart production (P1) contain exactly the same knowledge. Conceptually, both are functions that take some arguments and return a result. Procedural knowledge is stored in procedural memory and interpreted directly by the ACT-R interpreter while declarative knowledge is stored in declarative memory and interpreted indirectly, through productions similar to (P2) and (P3). Productions (P2) and (P3) act as embedded (in ACT-R) interpreters of declarative knowledge. We use the same representation for both declarative and procedural knowledge. For example, the declarative fact equivalent to (F1), which can be retrieved in phase 2, is represented by the same function (R1) as its procedural counterpart. Contrary to procedural knowledge, declarative knowledge is stored in declarative memory and rules in that memory have activations, analogously to ACT-R chunks. When there is a goal, such as the one represented by the term (T1), the interpreter first matches it against all the rules in procedural memory. If no rule matches or if the utility of any matched rule is low (because the model is not yet in phase 3), the interpreter matches the term against the rules in declarative memory. If no rule matches or if the activation of any matched rule is low (because the model is still in phase 1), a model starts interpreting instructions. Thus we free a modeller from writing generic rewrite rules, equivalent to productions (P2) and (P3), whose only purpose is to interpret declarative knowledge. We do not imply that the skills represented by (P2) and (P3) are innate, i.e. a part of the architecture (term rewriting interpreter in this case). We simply consider the equivalent rewrite rules boilerplate code from the programming perspective. As the ability to retrieve and harvest declarative knowledge is essential for almost any cognitive skill, most modellers would probably want those rules. In a future implementation, we will expose that part of the interpreter as rules that can be removed or customised, if desired.

To summarise, we represent both declarative and the corresponding procedural knowledge as functions stored in the two distinct memories. In learning phase 2, the solution of a problem is stored in declarative memory as a function that can be fetched and applied. In phase 3, the same function is present in procedural memory and can be applied directly, reflecting automatisation of the given skill. In ACT-R, transition between phases 2 and 3 is modelled as a production compilation that collapses rules (P2) and (P3) into (P1). Note that no such compilation is necessary in our approach as both kinds of knowledge share exactly the same representation. The given declarative knowledge is converted into procedural by copying a corresponding function to the procedural memory. After copying, its utility can increase with subsequent practice, as described in ACT-R theory.

In learning phase 1, a given problem is solved by following the instructions encoded as declarative knowledge. In addition to instructions, declarative knowledge of other facts is usually necessary, as shown in Table 2. We represent those facts with functions stored in the declarative memory. For example:

```
add(8,7)->15 (R2)
decrement(3)->2  (R4)
subtract(7,1)->6 (R4)
```

Operators encode instructions. A typical operator consists of an operation to be done, names of the slots from which arguments are obtained and name of a slot that stores the result. An argument may be a concrete value instead of a slot name. Table 2 shows the operators encoding the instructions for solving a pyramid task. Production (P4) requests an operator of the declarative module that can achieve the current goal. The retrieved operator is executed. As there are different possible kinds of actions, multiple productions are necessary for executing each of those actions. We call this set of productions (P5). Even though productions (P5) are specific to particular actions, they are still fairly generic as they can be used in different domains. Dynamic pattern matching (Anderson, 2007) is necessary to follow the instructions as slot names to be matched against are not known in advance and have to be read from the operators. In the same way (P2) and (P3) can be seen as embedded interpreters of declarative facts, (P4) and (P5) can be seen as embedded interpreters of declarative operators. As with declarative facts, we implement declarative operators as functions stored in declarative memory and implicitly include rules that fetch and interpret operators. Operators are higher order functions that return a function. Primitive generic operators perform basic actions corresponding to reading, comparing and writing slot values. These operators abstract over dynamic pattern matching. Non-primitive operators are composed from primitive operators and fact functions. A cognitive model is programmed as a set of operator functions composed in different ways to achieve given goals. The acquisition of a skill can be seen as „the composition of already-known component skills in novel ways to enable the performance of new skills and tasks" (Salvucci, 2013). The approach we introduce here can thus be seen as an implementation of that view.

We abstract over dynamic pattern matching by using `get`, `set` and `equal` functions. Function `get` takes a list of subscripts and returns a function (getter) that takes an expression and applies the subscripts to the expression. For example, `get(base)` returns `|$[base]|`. Function `set` also takes a list of subscripts, but it returns a function (setter) that takes a list of values and an expression and returns an expression with values stored at the subscripts. For example, `set(term)` returns `|$2[term->$1]|`. Function `equal` takes a list of subscripts and returns a function that tests whether the values at given subscripts are equal or not. Function `not` is a standard logic negation operator. Operator >> is a left-associative generalised function composition operator overloaded to work with different kinds of functions and constants. Finally, we define function `update(s,f)` as `get(s)>>f>>set(s)`. Although the implementation of the described constructs may seem involved, writing cognitive models using these constructs is relatively easy. We express almost all the operators from Table 2 as follows:

```
get(base)>>set(sum) (O1)
get(base)>>|subtract($,1)|>>set(term) (O2)
2>>set(count) (O3)
get(term,sum)>>add>>set(sum) (O4)
get(sum)>>say (O5)
update(term,decrement) (O7)
update(count,increment) (O8)
```

`add`, `subtract` etc. are exactly the same functions used to encode declarative knowledge. The main advantage of using the unified function representation for all kinds of knowledge is that we can build a cognitive model incrementally by hierarchically composing functions starting from the most primitive ones. Each intermediate function can be easily understood and tested in isolation and reused in other cognitive models. The only operator missing from the previous list is (O6), which does not encode any action. Instead, it encodes an instruction to test whether `height` and `count` are equal and decides to respond (fetch and execute (O5)) if they are or repeat the sequence (O7-O8-O4) if they are not. We express that kind of iteration by the `while(condition,f)` function which repeatedly applies `f` as long as the `condition` is `true`. With `while`, we express the remaining operator (O6) as `while(not(equal(height,count)),O7>>O8>>O4)`.

Finally, we compose the operators to express the entire pyramid task: `solvePyramid->O1>>...>>O4>>O6>>O5`. The resulting model is a function that can be further reused and composed too. The `solvePyramid` function is not stored in declarative memory as that would not be psychologically plausible. Instead, it reflects the order in which instructions are received, and we use the order to distribute the activation to the individual operators. Fetching the operators based on their activation is similar to the PRIMS approach (Taatgen, 2013). As we use rewrite rules to implement functions, we can trace the execution steps of the model. By applying cognitive interpretation to each step, using the principles described above, we effectively get the same simulation as by running the ACT-R model (Tenison et al., 2016), but with all the benefits of functions. The model starts with rules similar to (R2-R4) and operators (O1-O8) stored in declarative memory, much like the ACT-R model. When there is a goal encoded by the term (T1) and there is no function (R1) in the declarative memory, the model solves the problem by fetching and interpreting the appropriate operators. Some operators, such as (O1), correspond to single productions and can be interpreted in a single step. Other operators require sub-goalling and hence correspond to multiple productions that create sub-goals, fetch additional declarative facts, and return to the main goal by trying to fetch it from declarative memory after the subgoal is achieved. For example, after execution of (O1-O3), the working memory contains `solvePyramid(base->8,height->3,term->7)` (T2) and the operator (O4) is fetched. The operator reads the arguments supplied by `get` and passes them to `add` resulting in the term `add(8,7)`. This interprets as a subgoal and results in fetching (R2) from declarative memory and storing (T2) in declarative memory. If (R2) is fetched and executed, the interpreter retrieves and restores (T2) and stores the result of addition as encoded by `set`. To completely reproduce the results of (Tenison et al., 2016), we interpret setting the count and saying the result as motor actions performed by the corresponding modules. We leave the details of implementing motor actions as compositional functions for future work. When a pyramid is solved, its solution is stored in declarative memory as a function similar to (R1). Subsequent solving of the same pyramid increases the activation of its declarative function. When the activation is high enough, the model transitions to phase 2 and subsequently to phase 3. These transitions are similar to the equivalent transitions in the ACT-R model. However, as we start from fine-grained primitive functions rather than coarse-grained production rules, the composition process also produces many intermediate functions that can be reused in different models, as in PRIMs (Taatgen, 2013). Behaviour similar to PRIMs arises naturally as a consequence of using compositional functions and not as a consequence of our intentional design.

Concluding Remarks

A number of alternatives to using low-level production systems in cognitive modelling have been suggested. (Paik, Kim, & Ritter, 2010) divides those approaches in two groups: reimplementing existing languages in higher-level general purpose languages (1), and creating completely new higher-level languages specialised to writing cognitive models (2). A more recent approach is to program cognitive models directly in a general purpose high-level language by using a library implemented in that language (Salvucci, 2016) (3). Our approach is somewhere between 2) and 3). It is based on a particular higher-level language and in that sense similar to 2). Rather than defining special constructs for abstracting over the low level features of a cognitive architecture (ACT-R in this case), we model all aspects of cognition by first-order and higher-order functions, implemented as rewrite rules. As a result, programming feels much like programming in a general-purpose functional programming language, hence the similarity with 3). The work reported here is preliminary and its purpose is to introduce the approach to the cognitive modelling community. We plan to further develop the language and explore its strengths and weaknesses by using it to develop real world cognitive models. We also want to examine mentioned similarity with PRIMS (Taatgen, 2013) in more detail. Cognitive models implemented as term rewriting systems are actually algebraic structures. An interesting research question we want to explore is if we can utilise those structures and a number of formal methods based on term rewriting, to reason about cognitive models on a more fundamental mathematical level. Term rewriting is closely related to formal languages. We want to further explore theoretical and practical aspects of that relation in the context of cognitive modelling, what can be of special interest to computational and cognitive linguists.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* (1st ed.). Oxford University Press.

Anderson, J. R., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036-1050.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4*(2), 167–207.

Buchanan, B. G., & Shortliffe, E. H. (Eds.). (1985). *Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project*. Addison-Wesley.

Heeren, B., Jeuring, J., & Gerdes, A. (2010). Specifying rewrite strategies for interactive exercises. *Mathematics in Computer Science*, *3*(3), 349–370.

Paik, J., Kim, J., & Ritter, F. (2010). Building large learning models with Herbal. In *ICCM'10.*

Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science*, *37*(5), 829–860.

Salvucci, D. D. (2016). Cognitive code : An embedded approach to cognitive modeling. In *ICCM'16* (pp. 15–20).

Taatgen, N. A. (2005). Modeling parallelization and flexibility improvements in skill acquisition: from dual tasks to complex dynamic skills. *Cognitive Science*, *29*(3), 421–455.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological review*, *120*(3), 439–71.

Taatgen, N. A., & Lee, F. J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. *Human Factors*, *45*(1), 61-76.

Taatgen, N. A., van Vugt, M. K., Borst, J. P., & Mehlhorn, K. (2016). Cognitive modeling at ICCM: state of the art and future directions. *Topics in Cognitive Science*, *8*(1), 259–263.

Tenison, C., Fincham, J. M., & Anderson, J. R. (2016). Phases of learning: How skill acquisition impacts cognitive processing. *Cognitive Psychology*, *87*, 1–28.

Wolfram Research, Inc. (2017). *Mathematica 11.1.* Champaign, Illinois.

Warm (for winter): Comparison class understanding in vague language

Michael Henry Tessler[1], **Michael Lopez-Brau**[2], and **Noah D. Goodman**[1]
mtessler@stanford.edu, lopez_mic@knights.ucf.edu, ngoodman@stanford.edu
[1]Dept. of Psychology, Stanford University, [2]Dept. of Electrical & Computer Engineering, University of Central Florida

Abstract

Speakers often refer to context only implicitly when using language. The utterance "it's warm outside" could signal it's warm relative to other days of the year or just relative to the current season (e.g., it's warm for winter). *Warm* vaguely conveys that the temperature is high relative to some contextual *comparison class*, but little is known about how a listener decides upon such a standard of comparison. Here, we formalize how world knowledge and listeners' internal models of speech production can drive the resolution of a comparison class in context. We introduce a Rational Speech Act model and derive two novel predictions from it, which we validate using a paraphrase experiment to measure listeners' beliefs about the likely comparison class used by a speaker. Our model makes quantitative predictions given prior world knowledge for the domains in question. We triangulate this knowledge with a follow-up language task in the same domains, using Bayesian data analysis to infer priors from both data sets.

Keywords: comparison class; pragmatics; Rational Speech Act; Bayesian cognitive model; Bayesian data analysis

If it's 75 °F (24 °C) outside, you could say "it's warm." If it's 60 °F (16 °C), you might not consider it warm. Unless it's January; it could be warm for January. *Warm* is relative, and its felicity depends upon what the speaker uses as a basis of comparison—the *comparison class* (e.g., other days of the year or other days in January). Comparison classes are necessary for understanding adjectives and, in fact, any part of language whose meaning must be pragmatically reconstructed from context, including vague quantifiers (e.g., "He ate a lot of burgers."; Scholler & Franke, 2015) and generic language (e.g., "Dogs are friendly"; Tessler & Goodman, 2016a). The challenge for listeners is that the comparison class often goes unsaid (e.g., in "It's warm outside.").

The existence of comparison classes for understanding vague language is uncontroversial (Bale, 2011; Solt, 2009). Four-year-olds categorize novel creatures (*pimwits*) as either "tall" or "short" depending on the distribution of heights of *pimwits* and not the heights of creatures that are not called *pimwits*, suggesting the comparison class in that context is *other pimwits* (Barner & Snedeker, 2008). Adult judgments of the felicity for adjectives like "dark" or "tall" similarly depend upon fine-grained details of the statistics of the comparison class (Qing & Franke, 2014b; Schmidt, Goodman, Barner, & Tenenbaum, 2009; Solt & Gotzner, 2012).

Any particular object of discourse, however, can be conceptualized or categorized in multiple ways, giving rise to multiple possible comparison classes. A day in January is also a day of the year; if it's warm, it could be *warm for winter* or *warm for the year*. Why should one comparison class be preferred over another? To our knowledge, this question has not been addressed formally or empirically.[1] We propose that listeners actively combine category knowledge with pragmatic considerations to infer the comparison class implicitly used by the speaker. We introduce a minimal extension to the Rational Speech Act (RSA) model for gradable adjectives (Lassiter & Goodman, 2013) to allow it to flexibly reason about the implicit comparison class.

We derive two novel **qualitative predictions** from this model. Saying "it's warm" in winter should signal it's warm *for winter* (as opposed to *for the year*) more so than saying "it's cold". The opposite relationship should hold in summer, where "it's cold" should signal it's cold *for summer* more so than "it's warm". This prediction is driven by the *a priori* probability that the adjective could apply to the class (e.g., the probability that a given day in winter is warm; Prediction 1). In addition, regardless of the season and the adjective form (e.g., "warm" or "cold"), listeners who expect speakers to be informative will prefer classes that are relatively specific (e.g., relative to *the current season* as opposed to *the whole year*), as they carry more information content (Prediction 2). We test these predictions by eliciting the comparison class using a paraphrase dependent measure (Expt. 1).

As with any Bayesian cognitive model, explicitly specifying relevant prior knowledge (e.g., beliefs about temperatures) is necessary for the model to make **quantitative predictions**. The current methodological standard is to measure beliefs by having participants estimate quantities or give likelihood judgments (Franke et al., 2016). We pursue a different methodology. The RSA model captures a productive fragment of natural language; thus, it makes predictions about a related natural language task (Expt. 2). Critically, we can use the model to predict natural language judgments that require the *same prior knowledge* as in Expt. 1 and use Bayesian data analysis to jointly infer the shared priors. This approach harnesses the productivity of language into experiment design and allows us to reconstruct priors without having participants engage in challenging numerical estimation tasks.

Understanding comparison classes

Adjectives like *warm* and *cold* are vague descriptions of an underlying quantitative scale (e.g., temperature). The vagueness and context-sensitivity of these adjectival utterances can be modeled using threshold semantics ($[\![u]\!] = x > \theta$, for utterance u, scalar degree x, and threshold θ), where the threshold is probabilistically set with respect to a comparison class c via pragmatic reasoning (Lassiter & Goodman, 2013; see also Qing & Franke, 2014a):

[1] Theoretical work in semantics has instead focused on how information from a comparison class is used and what representations might be preferred (Bale, 2011; Solt, 2009).

$$L_1(x,\theta \mid u) \propto S_1(u \mid x,\theta) \cdot P_c(x) \cdot P(\theta) \tag{1}$$
$$S_1(u \mid x,\theta) \propto \exp\left(\alpha_1 \cdot \ln L_0(x \mid u,\theta)\right) \tag{2}$$
$$L_0(x \mid u,\theta) \propto \delta_{[\![u]\!](x,\theta)} \cdot P_c(x) \tag{3}$$

This is a Rational Speech Act (RSA) model, a recursive Bayesian model where speaker S and listener L coordinate on an intended meaning (for a review, see Goodman & Frank, 2016). In this framework, the pragmatic listener L_1 tries to resolve the state of the world x (e.g., the temperature) from the utterance she heard u (e.g., "it's warm"). She imagines the utterance came from an approximately rational Bayesian speaker S_1 trying to inform a naive listener L_0, who in turn updates her prior beliefs $P_c(x)$ via an utterance's literal meaning $[\![u]\!](x)$. Lassiter & Goodman (2013) introduced into RSA uncertainty over a semantic variable: the truth-functional threshold θ (Eq. 1). θ comes from an uninformed prior and is resolved by the listener by reasoning about the likely states of the world $P_c(x)$ (e.g., possible temperatures) and the likelihood that a speaker would say the adjective given a state and a threshold $S(u \mid x,\theta)$. The prior distribution over world-states $P_c(x)$ is always relative to some comparison class c (Eqs. 1 & 3) but where does the comparison class come from?

When a listener hears only that "it's warm outside" without an explicit comparison class (e.g., "...for the season"), we posit the listener infers the comparison class using her world knowledge of what worlds are plausible given different comparison classes $P(x \mid c)$, what comparison classes are likely to be talked about $P(c)$, and how a rational speaker would behave in a given world and comparison class $S_1(u \mid x,c,\theta)$ (Eq. 4). As a first test of this idea, we consider an idealized case where the comparison class can be either a relatively specific (subordinate) or relatively general (superordinate) categorization (e.g., warm relative to *days in winter* or relative to *days of the year*). Crucially in this situation, the listener is aware that the target entity is a member of the subordinate class (e.g., aware that it is winter) and draws likely values of the degree (e.g., temperature) from the subordinate class prior $P(x \mid c_{sub})$. With these assumptions, the model becomes:

$$L_1(x,c,\theta \mid u) \propto S_1(u \mid x,c,\theta) \cdot P(x \mid c_{sub}) \cdot P(c) \cdot P(\theta) \tag{4}$$
$$S_1(u \mid x,c,\theta) \propto \exp\left(\alpha_1 \cdot \ln L_0(x \mid u,c,\theta)\right) \tag{5}$$
$$L_0(x \mid u,c,\theta) \propto \delta_{[\![u]\!](x,\theta)} \cdot P(x \mid c) \tag{6}$$

We are interested in the behavior of the model with the underspecified utterance (e.g., "It's warm"), and we assume the speaker has two alternative utterances in which the comparison class is explicit (e.g., "It's warm relative to other days in winter." and "It's warm relative to other days of the year."). The predictions of this model depend on the details of the listener's knowledge of the subordinate and superordinate categories: $P(x \mid c_{sub})$ and $P(x \mid c_{super})$, as well as the prior distribution on comparison classes $P(c)$ in Eq. 4.

Comparison class prior $P(c)$ reflects listeners' expectations of what classes are likely to be discussed. As a proxy for comparison class usage frequency, we use empirical frequency $\hat{f}$ estimated from the Google WebGram corpus[2], and scale it by a free parameter β such that $P(c) \propto \exp(\beta \cdot \log \hat{f})$.

Degree priors (World knowledge) Only the relative values for $P(x \mid c_{sub})$ and $P(x \mid c_{super})$ affect model predictions. Hence we fix each superordinate distribution to be a standard normal distribution $P(x \mid c_{super}) = \mathcal{N}(0,1)$ and the subordinate priors to also be Gaussian distributions $P(x \mid c_{sub}) = \mathcal{N}(\mu_{sub}, \sigma_{sub})$; the subordinate priors thus have standardized units. We will eventually infer the parameters of the subordinate priors from experimental data.

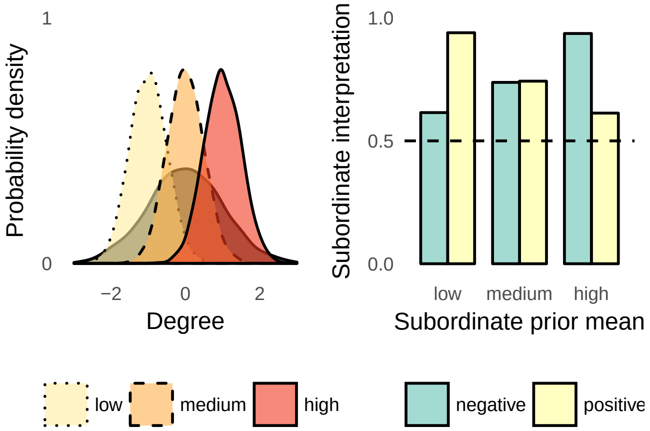


Figure 1: Left: Three hypothetical subordinate class prior distributions over a degree (fixing the superordinate class to be a unit-normal distribution, in grey). Right: Predicted listener inferences for an intended subordinate class interpretation given positive and negative form adjectives with different subordinate degree priors.

Qualitative model predictions Figure 1 (left) shows schematic superordinate and subordinate priors; e.g., temperatures over the whole year (super), in winter (low), fall (medium), and summer (high). The subordinate distributions have lower variance than the superordinate, and the "low" and "high" distributions have different means (e.g., temperatures in winter are expected to be lower and have lower variance than temperatures over the whole year).

Two intuitions explain the inferences of the pragmatic listener model (shown in Figure 1 right). First, certain classes are more or less likely to have an adjective felicitously apply. For example, any given day in winter is *less* likely to be warm than cold. Thus, hearing "it's warm" (a positive-form adjective) in winter (low prior) will signal it's warm *for winter* (the subordinate class) more so than hearing "it's cold" (negative-form), because it's more likely to be true (Prediction 1).

[2] Corpus accessed via `https://corpora.linguistik.uni-erlangen.de/cgi-bin/demos/Web1T5/Web1T5_freq.perl`. Due to potential polysemy and idiosyncracies of our experimental materials (Table 1), we made the following substitutions when querying the database for emprical frequency: produce → "fruits and vegetables"; things you watch online → "online videos"; days in {season} → "{season} days"; dishwashers → "dishwashing machines"; videos of cute animals → "animal videos".

Scale (adjectives)	Subordinate classes	Superordinate
Height (tall, short)	(professional) gymnast, soccer player, basketball player	people
Price (expensive, cheap)	bottle opener, toaster, dishwasher	kitchen appliances
Temperature (warm, cold)	winter, fall, summer (day in Maryland)	days in the year
Time (long, short)	video of a cute animal, music video, movie	things you watch online
Weight (heavy, light)	grape, apple, watermelon	produce

Table 1: Items used in Experiments 1 and 2. Subordinate categories were designed to fall near the low end, high end, and somewhere in the middle of the degree scale

Second, the amount of information conveyed by a vague utterance depends upon the variability in the comparison class. Comparison classes that have higher variance will result in relatively less information gain by the listener. All else being equal, listeners will prefer lower variance (e.g., subordinate) comparison classes because they are more informative (Prediction 2). Figure 1 (right) shows that subordinate class interpretations are above baseline regardless of the adjective polarity (positive or negative) or the mean of the subordinate prior (low, medium, high).

In sum, we see two predictions: The pragmatic listener overall prefers subordinate comparison classes, though the extent of this preference is modulated by the *a priori* probability that the adjective is true of the subordinate category. We test these two predictions in our first experiment.

Overview of data analytic approach As described above, specifying the relevant prior knowledge yields two free parameters per subordinate class. We will put priors over these parameters and infer their likely values using Bayesian data analysis. The data from the comparison class experiment (Expt. 1) would be insufficient, however, to reliably estimate all of the parameters of this data analytic model. To alleviate this, we use the same RSA model to predict additional data about related language use in the same domains (Expt. 2). Specifically, we gather judgments about adjectives when the comparison class is explicit: whether or not an adjective would apply to a subordinate member explicitly relative to the superordinate category (e.g., Is a day in winter warm relative to other days of the year?).

To model Expt. 2 data, we remove comparison class uncertainty by setting $P(c_{super}) = 1$, since the sentences provide an explicit comparison to the superordinate class. We model sentence endorsement using a pragmatic speaker (following Qing & Franke, 2014a; Tessler & Goodman, 2016a, 2016b):

$$S_2(u \mid c_{sub}) \propto \exp\left(\alpha_2 \cdot \mathbb{E}_{x \sim P_{c_{sub}}} \ln L_1(x \mid u)\right) \qquad (7)$$

Note that $L_1(x \mid u)$ is defined from Eq. 4 by marginalization.

Eqs. 4 and 7 define models for the data we will gather from Expts. 1 and 2, and depend on the same background knowledge $P(x \mid c)$. We can thus use data from both experiments to jointly reconstruct the shared prior knowledge and generate predictions for the two data sets. Experimental paradigms, computational models, preregistration report, and data for this paper can be found at `https://mhtess.github.io`.

Behavioral experiments

Experiment 1 tests the qualitative predictions of the model. Experiment 2 collects further data about adjective usage in order to constrain the quantitative predictions of the RSA model, which will be used to predict data from both experiments. The materials and much of the design of the two experiments are shared. Participants were recruited from Amazon's Mechanical Turk and were restricted to those with U.S. IP addresses with at least a 95% work approval rating. Each experiment took about 5 minutes and participants were compensated $0.50 for their work.

Materials We used positive- and negative-form gradable adjectives describing five scales (Table 1). Each scale was paired with a superordinate category, and for each superordinate category, we used three subordinate categories that aimed to be situated near the high-end, low-end, and intermediate part of the degree scale (as in Figure 1 left). This resulted in 30 unique items ({3 subordinate categories} x {5 scales} x {2 adjective forms}). Each participant saw 15 trials: one for each subordinate category paired with either the positive or negative form of its corresponding adjective. Participants never judged the same subordinate category for both adjective forms (e.g., cold and warm winter days) and back-to-back trials involved different scales to avoid fatigue.

Experiment 1: Comparison class inference

In this experiment, we gather human judgments of comparison classes in ambiguous contexts, testing the two predictions described in **Qualitative Model Predictions**.

Participants and procedure We recruited 264 participants and 2 were excluded for failing an attention check. On each trial, participants were given a context sentence to introduce the subordinate category (e.g., *Tanya lives in Maryland and steps outside in winter.*). This was followed by an adjective sentence, which predicated either a positive- or negative-form gradable adjective over the item (e.g., *Tanya says to her friend, "It's warm."*). Participants were asked "What do you think Tanya meant?" and given a two-alternative forced-choice to rephrase the adjective sentence with either an explicit subordinate or superordinate comparison class:

> {She / He / It} is ADJECTIVE (e.g., warm) relative to other SUBORDINATES (e.g., *days in winter*) or SUPERORDINATES (e.g., *days of the year*)

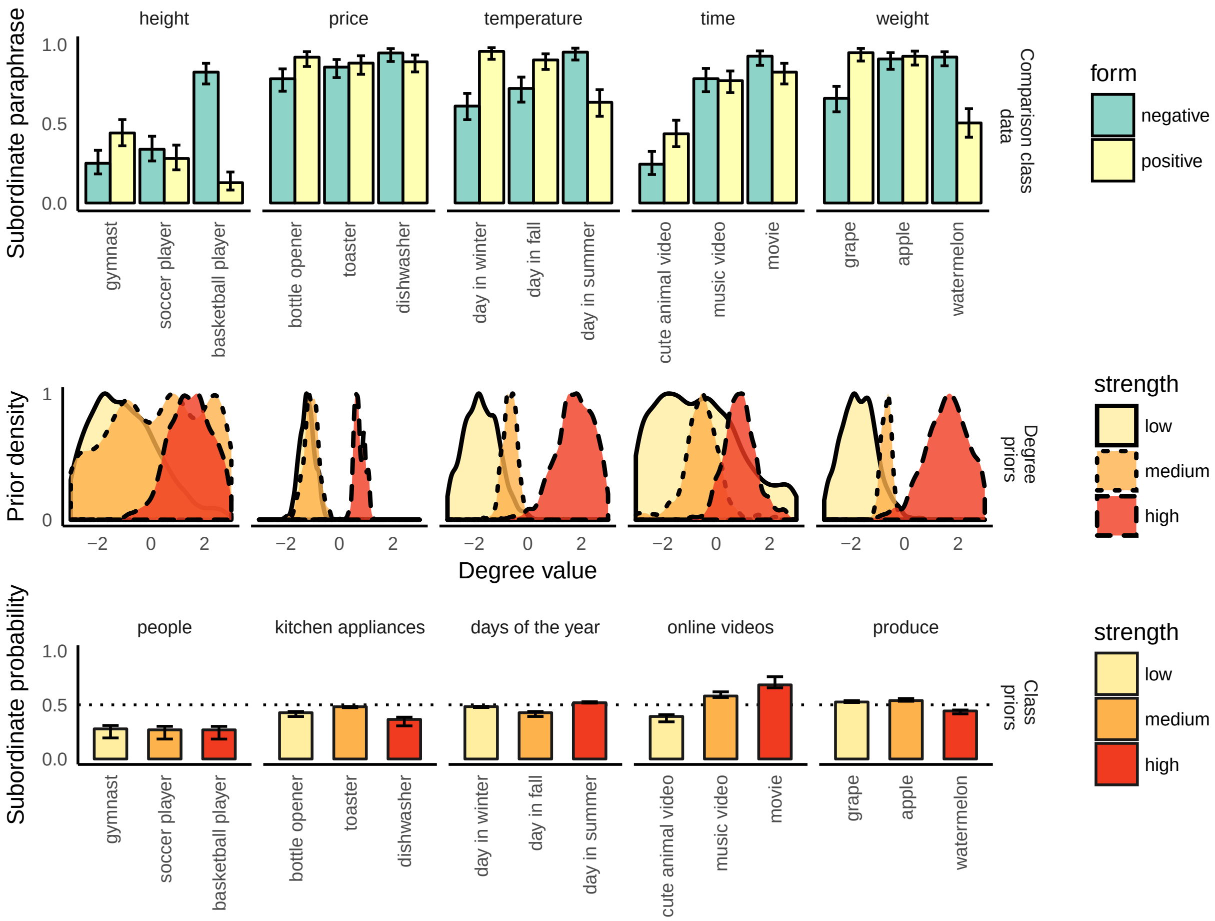


Figure 2: Empirical comparison class data, inferred world priors, and empirically derived comparison class priors. Top: Experiment 1 results. Comparison class judgments in terms of proportion judgments in favor of subordinate comparison class. Middle: Inferred prior distributions of world knowledge used to model Experiment 1 and 2 data. Bottom: Inferred prior probability of the subordinate comparison classes based on Google WebGram frequencies. Error bars correspond to 95% Bayesian credible intervals (for bottom plot, derived from the posterior on the β scale parameter).

In addition to all of the above design parameters, half of our participants completed trials where an additional sentence introduced the superordinate category at the beginning (e.g., *Tanya lives in Maryland and checks the weather every day.*), with the intention of making the superordinate paraphrase more salient.

Results We observed no systematic differences between participants' responses when the superordinate category was previously mentioned in the context and those when it was not; thus, we collapse across these two conditions for all analyses. Figure 2 (top) shows the proportion of participants choosing the *subordinate* paraphrase for each item, revealing considerable variability both *within*- and *across*- scales. The predicted effects are visually apparent within each scale (compare with Figure 1 right).

Our qualitative predictions are confirmed using a generalized linear mixed effects model with main effects of adjective form (positive vs. negative) and the *a priori* judgment by the first author of whether the sub-category was expected to be low or high on the degree scale, and of critical theoretical interest, the interaction between these two variables. In addition, we included by-participant random effects of intercept and by-subordinate category random effects of intercept and iteraction between form and strength[3]. Confirming our two qualitative model predictions, there was an interaction between form and strength ($\beta = -3.75$; $SE = 0.58$; $z = -6.49$) and there was an overall preference for subordinate category paraphrases ($\beta = 1.21$; $SE = 0.37$; $z = 3.27$). The main effects of form and strength were not significant.

We then test the simple effects. For items low on the degree scale (e.g., temperatures in winter), positive form adjectives were significantly more likely to imply subordinate comparison classes ($\beta = 1.41$; $SE = 0.15$; $z = 9.43$), while the opposite is true for items high on the scale (e.g., summer days; $\beta = -2.5$; $SE = 0.19$; $z = -13.15$). Participants reason pragmatically to resolve the comparison class, combining world knowledge with informativity as predicted by our model.

[3] This was the maximal mixed-effects structure that converged.

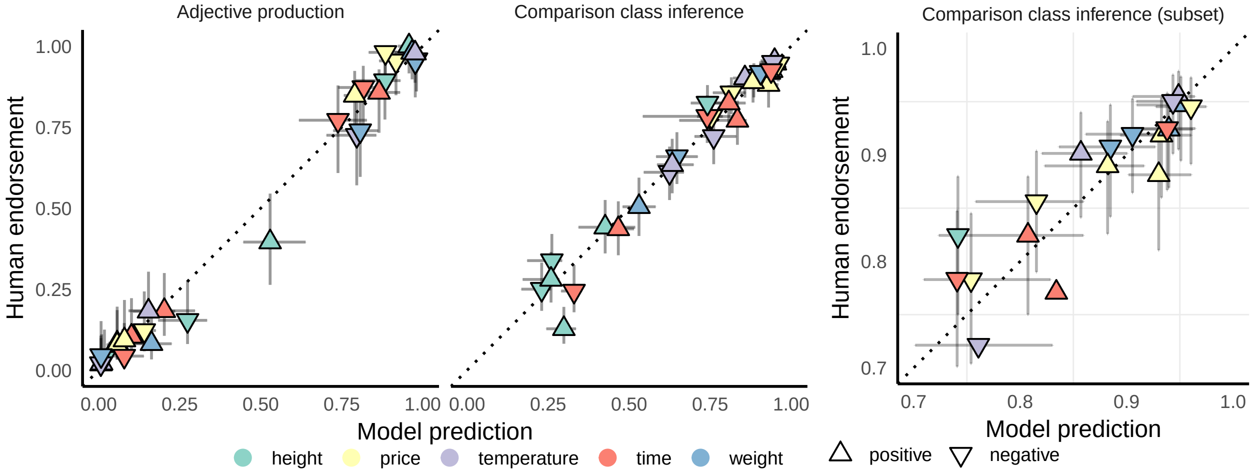


Figure 3: Human endorsement of subordinate comparison class paraphrases (middle; Expt. 1) and adjective sentences (left; Expt. 2) as a function of listener model L_1 and speaker model S_2 predictions, respectively. The right facet displays a subset of the paraphrase data (Expt. 1) to reveal good quantitative fit even in a small dynamic range. Error bars correspond to 95% Bayesian credible intervals.

Experiment 2: Adjective endorsement

In this experiment, we collected data about adjective endorsement that would require the same prior knowledge relevant for Expt. 1. We use this data to further constrain the RSA model's quantitative predictions.

Participants and procedure We recruited 100 participants and 5 were excluded for failing an attention check. On each trial, participants were given a sentence introducing the subordinate category (e.g., *Alicia lives in Maryland and steps outside in winter.*). This was followed by a question asking if the participant would endorse an adjective explicitly relative to the superordinate category (e.g., *Do you think the day in winter would be warm relative to other days of the year?*).

Results The judgments in this experiment were consistent with the *a priori* ordering of the subordinate categories on the degree scale. On the y-axis of Figure 3 (left), we see that the endorsement of adjectival phrases in these domains is markedly more categorical than the comparison class inference task (compare vertical spread of left and middle facets).

Full model analysis and results

The RSA listener (Eq. 4) and speaker (Eq. 7) models make quantitative predictions about comparison class interpretation and adjective endorsement, respectively. We construct a single data-analytic model with each of these RSA components as sub-models in order to make quantitative predictions about the data from both of our experiments.

The listener and speaker sub-models share their prior world knowledge $P(x \mid c)$ (e.g., temperatures in winter), described in the **Degree Priors** section. We put the same priors over the parameters of each subordinate distribution: $\mu \sim \text{Uniform}(-3,3)$, $\sigma \sim \text{Uniform}(0,5)$, since they have standardized units. The comparison class prior $P(c)$ in Eq. 4 scales the empirical frequency $\hat{f}$ by a free parameter, which we give the following prior: $\beta \sim \text{Uniform}(0,3)$.

The full model has three additional parameters not of direct theoretical interest: the speaker optimality parameters α_i^{expt}, which can vary across the two tasks. The pragmatic listener L_1 model (Eq. 4) has one speaker optimality: α_1^1. The pragmatic speaker S_2 model (Eq. 7) has two speaker optimality parameters: $\{\alpha_1^2, \alpha_2^2\}$. We use priors consistent with the previous literature: $\alpha_1 \sim \text{Uniform}(0,20)$, $\alpha_2 \sim \text{Uniform}(0,5)$

We implemented the RSA and Bayesian data analysis models in the probabilistic programming language WebPPL (Goodman & Stuhlmuller, 2014). To learn about the credible values of the parameters, we collecting 2 chains of 50k iterations (after 25k burn-in) using an incrementalized version of MCMC (Ritchie, Stuhlmuller, & Goodman, 2016).

Results The full model's posterior over the RSA and data-analytic parameters were consistent with prior literature and intuition. The maximum a-posteriori (MAP) estimate and 95% highest probability density (HPD) intervals for model parameters specific to the L_1 model used for Expt. 1 were $\alpha_1^1 = 1.6[1.1, 2.5]$, $\beta = 0.13[0.11, 0.19]$. Model parameters specific to the S_2 model used for Expt. 2: $\alpha_1^2 = 3.5[0.6, 13.2]$, $\alpha_2^2 = 3.2[2.6, 3.8]$. The inferred distributions corresponding to subordinate class priors were consistent with the *a priori* ordering of these subordinate classes (low, medium, high) used in these tasks (Figure 2 middle).

Finally, the full model's posterior predictive distribution does an excellent job at capturing the quantitative variability in responses for Expt. 1: $r^2(30) = 0.965$, and Expt. 2: $r^2(30) = 0.985$ (Figure 3). Because of the overall preference for the subordinate comparison class, many of the data points are distributed above 0.5. Even for these fine-grained differences, the model does a good job at explaining the quantitative variability in participants' data (Figure 3 right).

Discussion

The words we say are often too vague to have a single, precise meaning and only make sense in context. Context, however, can also be underspecified, as there are many possible dimensions or categories that a speaker might be implicitly referring to or comparing against. Here, we investigate the flexibility in the class against which an entity can be implicitly compared.

We introduced a minimal extension to an adjective interpretation Rational Speech Act model to allow it to flexibly reason about the comparison class. This model made two novel predictions about how listeners should prioritize one class over another. It also made quantitative predictions about how background knowledge about the degree scale should inform this inference in a graded fashion. Both qualitative predictions of the model were borne out in our first experiment, and the quantitative predictions were confirmed using a novel data analytic technique. To our knowledge, this is the first experiment to demonstrate how reference classes for adjective interpretation can adjust based on world knowledge.

We observe in our modeling results for Expt. 1 that a uniform prior distribution over the experimentally supplied comparison class alternatives is unlikely (Figure 2 bottom). For example, the comparison class of "people" for heights of individuals is relatively more salient than the class of "produce" for the weights of fruits and vegetables. We used the frequency of the class in a corpus as a proxy for their prior probability $P(c)$, which was sufficient to account for differences in baseline class probability both *between*- and *within*-scales.

Corpus frequency is a composite measurement of factors relevant for speech production. Its utility in this model suggests that utterances without an explicit comparison class (e.g., "It's warm outside") may in fact be incomplete sentences, in a way analogous to sentence fragments studied in noisy-channel models of production and comprehension (Bergen & Goodman, 2015). Another (non-mutually exclusive) possibility is that the comparison class prior reflects basic-level effects in categorization (Rosch & Mervis, 1975). Future work should attempt to understand these factors to construct a more complete theory of the comparison class prior.

The second contribution of this paper is a novel data-analytic approach, where prior knowledge used in the Bayesian language model is reconstructed from converging evidence gathered from related language experiments. In previous work, we have attempted to measure prior knowledge by decomposing what would be a single, implicitly multi-layered, numerical estimation question into multiple simpler questions. Then, we construct a Bayesian data analytic model to back out the prior knowledge (Tessler & Goodman, 2016a, 2016b). We extend this approach by using the same core RSA model to model behavior across two language experiments. The major feature of this method is that participants respond only to simple, natural language questions rather than estimating numerical quantities for which complicated linking functions must be designed (e.g., Franke et al., 2016). The fully Bayesian language approach we pioneer here also provides a further constraint on the language model, which must predict data from two similar but distinct language experiments. The productivity of natural language can thus be harnessed to productively design experiments that further constrain and test computational models of language and cognition.

Acknowledgements

The authors would like to thank Ali Horowitz for help in stimuli design. This work was supported in part by NSF Graduate Research Fellowship DGE-114747 to MHT, a Stanford CSLI Summer Internship for MLB, and a Sloan Research Fellowship, ONR grant N00014-13-1-0788, and DARPA grant FA8750-14-2-0009 to NDG.

References

Bale, A. C. (2011). Scales and comparison classes. *Natural Language Semantics*, *19*(2), 169–190.

Barner, D., & Snedeker, J. (2008). Compositionality and statistics in adjective acquisition: 4-year-olds interpret tall and short based on the size distributions of novel noun referents. *Child Development*, *79*(3), 594–608.

Bergen, L., & Goodman, N. D. (2015). The strategic use of noise in pragmatic reasoning. *Topics in Cognitive Science*, *7*(2), 336–350.

Franke, M., Dablander, F., Scholler, A., Bennett, E., Degen, J., Tessler, M. H., ... Goodman, N. D. (2016). What does the crowd believe ? A hierarchical approach to estimating subjective beliefs from empirical data. In *Proceedings of the 38th annual meeting of the cognitive science society*.

Goodman, N. D., & Frank, M. C. (2016). Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, *20*(11), 818–829.

Goodman, N. D., & Stuhlmuller, A. (2014). The Design and Implementation of Probabilistic Programming Languages. `http://dippl.org`.

Lassiter, D., & Goodman, N. D. (2013). Context, scale structure, and statistics in the interpretation of positive-form adjectives. In *Semantics and linguistic theory* (Vol. 23, pp. 587–610).

Qing, C., & Franke, M. (2014a). Gradable adjectives, vagueness, and optimal language use: A speaker-oriented model. In *Semantics and linguistic theory* (Vol. 24, pp. 23–41).

Qing, C., & Franke, M. (2014b). Meaning and Use of Gradable Adjectives: Formal Modeling Meets Empirical Data. In *Proceedings of the 36th annual conference of the cognitive science society*.

Ritchie, D., Stuhlmuller, A., & Goodman, N. D. (2016). C3: Lightweight incrementalized mcmc for probabilistic programs using continuations and callsite caching. In *AISTATS 2016*.

Rosch, E., & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, *7*(4), 573–605.

Schmidt, L. A., Goodman, N. D., Barner, D., & Tenenbaum, J. B. (2009). How Tall Is Tall? Compositionality, Statistics, and Gradable Adjectives. In *Proceedings of the 31st annual conference of the cognitive science society*.

Scholler, A., & Franke, M. (2015). Semantic values as latent parameters: Surprising few & many. In *Semantics and linguistic theory* (Vol. 25, pp. 143–162).

Solt, S. (2009). Notes on the Comparison Class. In *International workshop on vagueness in communication*.

Solt, S., & Gotzner, N. (2012). Experimenting with degree. In *Semantics and linguistic theory* (Vol. 22, pp. 166–187).

Tessler, M. H., & Goodman, N. D. (2016a). A pragmatic theory of generic language. *ArXiv Preprint ArXiv:1608.02926*.

Tessler, M. H., & Goodman, N. D. (2016b). Communicating generalizations about events. In *Proceedings of the 38th annual meeting of the cognitive science society*.

Degrees of Separation in Semantic and Syntactic Relationships

Matthew A. Kelly (matthew.kelly@psu.edu), David Reitter (reitter@psu.edu)
The Pennsylvania State University, University Park, PA

Robert L. West (robert.west@carleton.ca)
Carleton University, Ottawa, ON, Canada

Abstract

Computational models of distributional semantics can analyze a corpus to derive representations of word meanings in terms of each word's relationship to all other words in the corpus. While these models are sensitive to topic (e.g., tiger and stripes) and synonymy (e.g., soar and fly), the models have limited sensitivity to part of speech (e.g., book and shirt are both nouns). By augmenting a holographic model of semantic memory with additional levels of representations, we present evidence that sensitivity to syntax is supported by exploiting associations between words at varying degrees of separation. We find that sensitivity to associations at three degrees of separation reinforces the relationships between words that share part-of-speech and improves the ability of the model to construct grammatical sentences. Our model provides evidence that semantics and syntax exist on a continuum and emerge from a unitary cognitive system.

Keywords: semantic memory; mental lexicon; distributional semantics; word embeddings; holographic models; cognitive models; semantic space; part-of-speech; language production

Introduction

How do humans acquire, produce, and comprehend language? To what extent does language require a specialized cognitive capacity? And to what extent do humans learn language the same way that humans learn any other skill, whether it is learning to play chess or to play a piano piece?

Computational cognitive models provide a means of investigating the extent to which basic cognitive functions play a role in language. Computational models of learning and memory have been able to account for a variety of psycholinguistic phenomena without any *a priori* linguistic knowledge.

Linguistics distinguishes *lexical* knowledge (describing words) from *syntactic* processes (describing how words are combined to form sentences). We modify an existing computational model of the acquisition of lexical knowledge to enhance its ability to provide an integrated account of the acquisition of syntactic knowledge.

Our model, the Hierarchical Holographic Model (HHM), is based on BEAGLE (Jones & Mewhort, 2007). BEAGLE is a distributional semantics model that uses holographic memory (Plate, 1995). Distributional models infer the meaning of words from how the words co-occur in a corpus. BEAGLE's algorithm is not specific to language and has been applied to recognition memory (Kelly, Kwok, & West, 2015), learning a decision-making task, math cognition, and playing simple games (Rutledge-Taylor, Kelly, West, & Pyke, 2014).

Building on work by Grefenstette (1994), we define *orders of association* as a measure of the relationship between words. This notion is related to *degrees of separation*, a measure of the distance between two nodes in a connected graph.

First-order (direct) associations are useful for detecting words that are related in topic (e.g., *tiger* and *stripes*) and second-order associations are useful for detecting words that have a degree of synonymy (e.g., *tiger* and *lion*). Distributional semantics models, such as BEAGLE, are sensitive to both first and second-order associations.

Distributional models are weakly sensitive to part-of-speech (e.g., *book* and *shirt* are nouns). In the semantic space of distributional models, words tend to cluster by part-of-speech, such that, using a classifier, these models can be used for automated part-of-speech tagging (e.g., Tsuboi, 2014).

Distributional models are not, strictly speaking, sensitive to these clusters, it is the work of the classifier to detect them. While all words in a cluster will be similar to some other words in the cluster, there may be words in the cluster that are entirely dissimilar to each other. This is because similarity is not transitive. These clusters are evidence of higher-order associations that all words in the cluster have to all other words in the cluster. Thus, we propose a variant of BEAGLE that is sensitive to arbitrarily indirect associations. This allows us to explore how higher-order associations can be utilized to improve on the ability of computational models of distributional semantics to infer syntactic information from a corpus.

Our Hierarchical Holographic Model is not a model of syntax or semantics *per se*, as it does not produce or comprehend utterances. However, HHM generates representations that capture knowledge of how a word is used, what words it can be used with, and how those words should be sequenced to form a grammatical utterance. HHM's representations can be situated in and utilized by a model that operates at the utterance level (e.g., Johns, Jamieson, Crump, Jones, & Mewhort, 2016). The objective of this research is to provide a foundation for a single system account of the acquisition of semantic and syntactic lexical knowledge that is based on a general-purpose computational model of human memory.

In this paper, we explain the theory and mechanics of the Hierarchical Holographic Model and show how the model can be used to learn part of speech relations between words and to order words into grammatical sentences. In sum, we present contributions to a theory of human memory, describe a computational model based on that theory, and evaluate the model on human linguistic behavior.

Theory

In what follows, we define *orders of association* as a measure of the relationship between a pair of words in memory. We describe the BEAGLE model of distributional semantics

Table 1: Example of a third order association between *eagles* and *birds*.

Sentences			
eagles *soar over trees*	airplanes *soar* through skies	dishes are *over* plates	squirrels live in *trees*
birds *fly above forest*	airplanes *fly* through skies	dishes are *above* plates	squirrels live in *forest*

(Jones & Mewhort, 2007), based on the holographic model of memory (Plate, 1995). We then propose the Hierarchical Holographic Model (HHM), a variant of BEAGLE capable of detecting arbitrarily high orders of association.

Orders of Association

Imagine a graph where each word in the lexicon is a node connected to other words. A pair of words are connected once for each time they have occurred in the same context. In human cognition, that context is defined by the limited capacity of working memory. In our model, the context is a window of 5 to 10 words to the left and right of the target word. *Order of association* is the length of a path between two words in the graph. The *strength* of that order of association is the number of paths of that length between the two words.

First order association is when two words appear together. In the sentence "eagles soar over trees", the words *eagles* and *trees* have first order association. Words with strong first order association (i.e., frequently appear together) are often related in topic, such as the words *tiger* and *stripes*.

Second order association is when two words appear with the same words. In the sentences "airplanes soar through skies" and "airplanes fly through skies", *soar* and *fly* have second order association. Words with strong second order association are often synonyms.

Third order association is when two words appear with words that appear with the same words. Given the sentences in Table 1, the words *eagles* and *birds* have neither first nor second order association, but do have third order association.

Fourth order and higher One can keep abstracting to higher orders of association indefinitely. Eventually, all words are related to all other words in the language.

No association A pair of words with no path between them have no association of any order. For an agent that knows only the eight sentences in Table 1 as well as a ninth sentence "cars drive on streets", the words *car* and *eagle* have no association. In real language data, two words will only have no association if they belong to two different languages.

The definition of *orders of association* that we provide here is an application of the concept of *degrees of separation* in a network to words in a language, and is a generalization of Grefenstette (1994)'s first-order, second-order, and third-order affinities between words.

According to Barceló-Coblijn, Corominas-Murtra, and Gomila (2012), the point at which a child transitions from speaking in utterances of one or two words to speaking in full sentences is the point at which the child's knowledge of the relationships between words forms a dense ”small world" graph, typical of an adult vocabulary, where all words are several steps from all other words in the graph. We hypothesize that learning these longer range connections between words is necessary to construct novel syntactic utterances.

To define orders of association, we have described the lexicon as a connected graph. This graph is not explicitly represented by the computational models we use. The BEAGLE model defines a space rather than a graph, where words are points in space. Words close together in BEAGLE's space have strong second-order association. Our Hierarchical Holographic Model (HHM) extends BEAGLE by defining a space for each order of association. Level 1 of HHM is BEAGLE, Level 2 represents third-order associations as distance, Level 3 represents fourth-order associations, and so on.

Previous computational models that detect third-order associations (or higher) have been clustering or classification algorithms applied to words organized in a space of second-order associations (e.g., Grefenstette, 1994; Tsuboi, 2014). Conversely, HHM recursively applies the memory and learning principles it uses to detect second order associations to detect higher order associations. As such, even at higher-orders, HHM does not produce discrete categories corresponding to noun, verb, adverb, etc., but instead produces graded representations of lexical syntactic relationships.

We expect that fourth-order associations may be sufficient to capture syntactic relationships. In a semantic network constructed from English word co-occurrence, the average minimum path length between any pair of words is between 3 and 6, depending on how the network is constructed (Steyvers & Tenenbaum, 2005). As such, we expect that by Level 3 of HHM, many words will be related to half the lexicon.

The BEAGLE Model

In the BEAGLE model of semantic memory (Jones & Mewhort, 2007), each word is represented by two vectors: an environment vector that represents the percept of a word and a memory vector that represents the concept of a word.

An environment vector (denoted by **e**) stands for what a word looks like in writing or sounds like when spoken. For simplicity, we chose not to simulate the visual or auditory features of words (but see Cox, Kachergis, Recchia, & Jones, 2011 for a version of BEAGLE that does simulate these features). Instead, we generate the environment vectors using random values, as in (Jones & Mewhort, 2007). In our simulations, environment vectors are generated by randomly sampling values from a Gaussian distribution with a mean of zero and a variance of $1/n$, where n is the dimensionality. These dimensions are meaningless, only the relationships between vectors are meaningful. The number of dimensions, *n*, determines the fidelity with which BEAGLE stores the word co-occurrence data, such that smaller *n* yields poorer encoding.

Memory vectors (denoted by **m**) represent the associations a word has with other words. Memory vectors are constructed as the model reads the corpus. Memory vectors are holographic in that they use circular convolution (denoted by $*$) to compactly encode associations between words (Plate, 1995). Given a sentence, for each word in the sentence, vectors representing all sequences of words in the sentence (or grams) that include the target word are summed together and added to the target word's memory vector.

For example, given the sentence, "eagles soar over trees", we update the memory vectors for each word in the sentence: *eagles, soar, over,* and *trees*. Each memory vector is updated with a sum of grams. The memory vector for the word *soar*, $\mathbf{m}_{\text{soar}}$, is updated with the bigrams "eagles soar" and "soar over", the trigrams "eagles soar over" and "soar over trees", and the tetragram "eagles soar over trees".

Each gram is constructed as a convolution of the environment vectors of the constituent words, except for the target word, which is represented by the placeholder vector (denoted by ϕ). The placeholder vector is randomly generated and serves as a universal retrieval cue. With the placeholder substituted for the target word, each gram can be understood as a question to which the target word is the answer. So, rather than adding a representation of "eagles soar over" in $\mathbf{m}_{\text{soar}}$, we instead add "eagles ? over", i.e., "What was the word that appeared between *eagles* and *over*?". Each memory vector can be understood as the sum of all questions to which that memory vector's word is an appropriate answer.

For example, given "eagles soar over trees", we add "eagles ?", "? over", "eagles ? over", "? over trees", and "eagles ? over trees" to $\mathbf{m}_{\text{soar}}$ as follows:

$$\begin{aligned}\mathbf{m}_{\text{soar},t+1} = \mathbf{m}_{\text{soar},t} &+ \mathbf{P}_{\text{before}}(\mathbf{e}_{\text{eagles}}) * \phi + \mathbf{P}_{\text{before}}(\phi) \\ &* \mathbf{e}_{\text{over}} + \mathbf{P}_{\text{before}}(\mathbf{P}_{\text{before}}(\mathbf{e}_{\text{eagles}}) * \phi) \\ &* \mathbf{e}_{\text{over}} + \mathbf{P}_{\text{before}}(\mathbf{P}_{\text{before}}(\phi) * \mathbf{e}_{\text{over}}) * \mathbf{e}_{\text{trees}} \\ &+ \mathbf{P}_{\text{before}}(\mathbf{P}_{\text{before}}(\mathbf{P}_{\text{before}}(\mathbf{e}_{\text{eagles}}) * \phi) * \mathbf{e}_{\text{over}}) * \mathbf{e}_{\text{trees}}\end{aligned} \tag{1}$$

where *t* is the current time step, all vectors **m**, **e**, and ϕ have *n* dimensions, and $\mathbf{P}_{\text{before}}$ is a permutation matrix used to indicate that a word occurred earlier in the sequence. $\mathbf{P}_{\text{before}}$ is constructed by randomly permuting the rows of the *n* x *n* identity matrix. Multiplying a vector **v** by $\mathbf{P}_{\text{before}}$ results in the permuted vector $\mathbf{P}_{\text{before}}\mathbf{v}$.

While BEAGLE is a model of lexical semantics, variants of BEAGLE have been applied to non-linguistic memory and learning tasks, such as learning sequences of actions for strategic game play (Rutledge-Taylor et al., 2014). We previously proposed a variant of BEAGLE (Kelly et al., 2015) that learns sets of property-value pairs (e.g., *colour:red shape:octagon type:sign label:stop*) of the kind used by the ACT-R cognitive architecture (Anderson & Lebiere, 1998). Thus, the BEAGLE algorithm can be applied to any problem domain that can be translated into discrete symbols. This holds true for the Hierarchical Holographic Model (HHM). While we evaluate HHM in this paper in terms of its ability to account for properties of natural language, HHM is intended as a general model of learning and memory.

Hierarchical Holographic Model

The Hierarchical Holographic Model (HHM) is a series of BEAGLE models, such that the memory vectors of one model serves as the environment vectors for the next model. Level 1 is a standard BEAGLE model with randomly generated environment vectors. Once Level 1 has been run on a corpus, Level 2 is initialized with Level 1's memory vectors as its environment vectors. Level 2 is run on the corpus to generate a new set of memory vectors, which in turn are used as the environment vectors for the next level, and so on, to generate as many levels of representations as desired.

To use the memory vectors of a previous level as the environment vectors for the next, one must normalize and randomly permute the vectors (Kelly, Blostein, & Mewhort, 2013). For level $l+1$, and all words *i*, the environment vectors for that level are:

$$\mathbf{e}_{l+1,i} = \mathbf{P}_{\text{group}}\left(\frac{\mathbf{m}_{l,i}}{\sqrt{\mathbf{m}_{l,i} \bullet \mathbf{m}_{l,i}}}\right) \tag{2}$$

where $\mathbf{P}_{\text{group}}$ is a random permutation used to transform memory vectors into environment vectors and $\bullet$ is the dot product.

The levels in HHM are virtual mental constructs that could all be represented within a single fully distributed neural structure. There is no limit to the number of such levels that could exist in the mind, as they are not physical constructs.

The levels in HHM can be understood as the products of memory re-consolidation, the process of revisiting experiences and recording new information about those experiences. The different levels of representation are stored separately from each other in the model for the purpose of examining the differential effects of representations that encode lower and higher orders of associations. The different levels are not necessarily separate memory systems.

Experiments

In what follows, we show that the Hierarchical Holographic Model (HHM) is able to detect third-order associations using a small example data set (Experiment 1). Running HHM on a corpus of novels from Project Gutenberg, we show that sensitivity to third or fourth order associations strengthens similarity between words that are the same part of speech (Experiment 2) and improves the ability of the model to order words into grammatical sentences (Experiment 3). These results show that HHM works as intended and that higher-order associations provide useful language data.

Experiment 1: Small Example Data Set

Higher levels of the model are sensitive to higher orders of association, as demonstrated by an example data set consisting of the eight sentences in Table 1 as well as an unrelated control sentence, ”cars drive on streets”. This is a toy example chosen to provide a clear illustration of how HHM works. We believe this toy example is important because understand how HHM behaves in this example is critical to understanding how HHM behaves on real language data.

HHM was run with 1024 dimensional vectors and three levels of representations. In the nine sentences of this example, there are 21 unique words, and therefore 210 unique pairs of words. We can characterize the behavior of HHM by how the word pairs change in similarity across levels. In Figure 1, of the 210 word pairs, we graph the 24 word pairs that have non-negative similarity by Level 3. Of those 24 pairs, we label the 10 pairs with the most similarity.

The memory vectors for words with second order association, such as *soar* and *fly*, are close on Level 1 (cosine = 0.51) and closer by Level 3 (cosine = 0.67). Words *eagle* and *bird*, which have only third order association, are unrelated on Level 1 (cosine = -0.01) but are the fifth most similar word pair by Level 3 (cosine = 0.33).

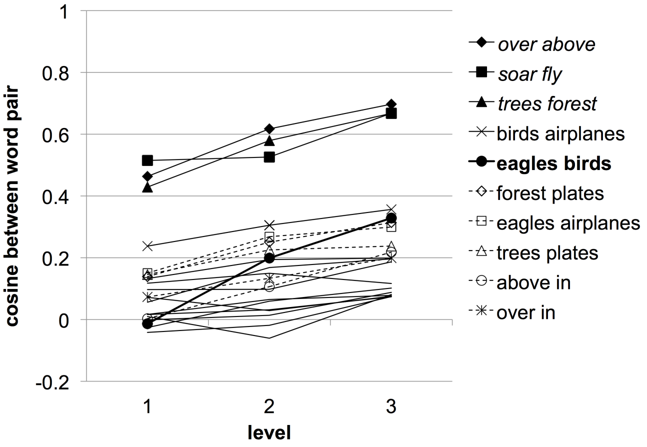


Figure 1: Cosines between word pairs across levels.

These results provide a simple example of the effect of the higher levels. Each memory vector at Level 1 is constructed as a sum of convolutions of environment vectors. As such, the memory vectors at Level 1 encode first order associations with respect to the environment vectors, measuring the frequency with which each word co-occurs with other words and sequences of words. The cosines between memory vectors are a measure of second-order association, the degree to which the two words co-occur with the same words. The algorithm that produces Level 1 transforms data that captures first-order association (co-occurrence) into data that captures second-order associations. The algorithm is a step, and by repeating it to produce higher levels, we can build a staircase.

Level 1 of the model cannot detect third-order associations. A pair of words with third-order association, but not first or second, do not appear together in the same sentence and do not co-occur with the same words. As such, the memory vectors for a pair of words with only third-order association will be constructed from disjoint sets of vectors. At Level 1, $\mathbf{m}_{1,\text{eagles}}$ is a sum of convolutions of $\mathbf{e}_{1,\text{soar}}$, $\mathbf{e}_{1,\text{over}}$, $\mathbf{e}_{1,\text{forest}}$, whereas $\mathbf{m}_{1,\text{birds}}$ is a sum of convolutions of $\mathbf{e}_{1,\text{fly}}$, $\mathbf{e}_{1,\text{above}}$, $\mathbf{e}_{1,\text{trees}}$. As Level 1 environment vectors are approximately orthogonal, the memory vectors constructed from them will also be approximately orthogonal. As a result, $\mathbf{m}_{1,\text{eagles}}$ and $\mathbf{m}_{1,\text{birds}}$ are approximately orthogonal (cosine = -0.01).

But at higher levels, the environment vectors are no longer orthogonal. Level 2 environment vectors are the Level 1 memory vectors. As a result, $\mathbf{e}_{2,\text{soar}}$ is similar to $\mathbf{e}_{2,\text{fly}}$ (cosine = 0.51), $\mathbf{e}_{2,\text{over}}$ is similar to $\mathbf{e}_{2,\text{above}}$ (cosine = 0.46), and $\mathbf{e}_{2,\text{forest}}$ is similar to $\mathbf{e}_{2,\text{trees}}$ (cosine = 0.43). Even though $\mathbf{m}_{2,\text{eagles}}$ and $\mathbf{m}_{2,\text{birds}}$ are still constructed from disjoint sets of environment vectors, because the vectors that they are constructed from are similar, $\mathbf{m}_{2,\text{eagles}}$ and $\mathbf{m}_{2,\text{birds}}$ are somewhat similar (cosine = 0.20). As the memory vectors for the pairs *soar* and *fly*, *above* and *over*, and *forest* and *trees* are more similar at Level 2 than at Level 1 (see Figure 1), the environment vectors for them will be more similar at Level 3 than Level 2, which increases the similarity between *eagles* and *birds* at Level 3 (cosine = 0.33).

Experiment 2: Part of Speech

We trained HHM on a corpus of novels from Project Gutenberg. The corpus is 10 238 600 sentences with 145 393 172 words and 39 076 unique words. HHM read the corpus one sentence at a time. Within each sentence, HHM used a moving window of 21 words, 10 words to the left and right of a target word. In that window, all grams that included the target word, from bigrams up to 21-grams, were encoded as convolutions of environment vectors and summed into the target word’s memory vector. We used 1024 dimensional vectors.

Using WordNet (Princeton University, 2010) and the Moby Part-Of-Speech list (Ward, 1996), we assigned a part of speech tag to each word in the 39 076 word vocabulary. Here we use similarity between words that are the same part-of-speech (noun, verb, adjective, etc.) as a proxy measure for knowledge that those words can be used in similar ways.

To examine the effect of third-order associations, we compare Levels 1 and 2. We limit our analysis to words with at least 1000 occurrences in the corpus, as these words will have the most robust vector representations, and to word pairs that increased or decreased in similarity the most between levels.

As shown in Table 2, of the 1000 word pairs that increased the most in similarity from Level 1 to 2, 71% of those words have matching part-of-speech: 48% are partial matches (e.g., *associated* and *searching* are both verbs, but *searching* is also an adjective) and 23% are exact matches (e.g., *focused* and *emerging* can both be an adjective or a verb).

In total, 13% of all pairs of words in the lexicon are exact matches (see Table 2). Among the 1000 word pairs that increased the most from Level 1 to Level 2, there are significantly more (23%) exact matches than would be expected in a random sample from the set of all word pairs ($p < 0.0001$).

Of the 1000 word pairs that decreased in similarity the most from Level 1 to 2, only 1% are exact matches (e.g., both *local*

Table 2: Top 1000 word pairs that changed in similarity the most at each level, categorized by part-of-speech match.

Level	Change	Exact	Partial	Mismatch
total	-	*13%*	*45%*	*42%*
1 to 2	increase	23%	48%	29%
1 to 2	decrease	1%	53%	46%
2 to 3	increase	26%	44%	30%
2 to 3	decrease	0%	1%	99%

and *wizard* can be used as an adjective and a noun), which is significantly fewer than chance ($p < 0.0001$).

From Level 2 to 3, we find that 26% of the word pairs that increased in similarity the most are exact matches, which is significant ($p < 0.0001$). Of the word pairs that decreased in similarity from Level 2 to 3, zero were exact matches and only 1% were partial matches (e.g., *never* and *oh* can both be exclamations, but *never* is more commonly an adverb), which, again, was significantly less than chance ($p < 0.0001$).

In sum, we find that the sensitivity to third order (Level 2) and fourth order associations (Level 3) strengthens similarities between words with matching part of speech and weakens similarities between words with mismatching part of speech.

Experiment 3: Word Ordering Task

Do higher-order associations provide additional useful information about how to sequence words into a sentence? When given an unordered set of words that can be arranged into a sentence, are higher levels of HHM better able to find the grammatical ordering? We replicate a task from Johns et al. (2016). In this task, the model is given a set of n words from an n-word sentence that is not present in the exemplar set. The model must discern which of the n! possible word orderings is the grammatical, original ordering.

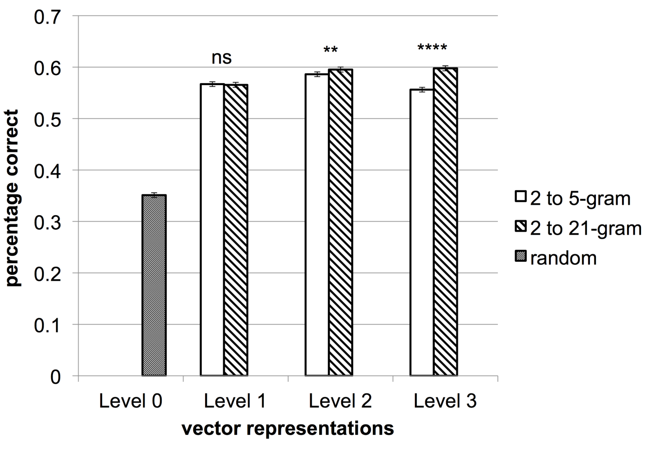


Figure 2: Percentage of test sentences correctly ordered by model as a function of vectors used to represent words.

The exemplar set consists of 125 000 seven-word sentences randomly sampled from the Project Gutenberg corpus. Sentences in the exemplar set have no words with frequency less than 300. All test set sentences and permutations thereof are excluded from the exemplar set.

We embed the word representations generated by each level of HHM in a minimal exemplar model of syntax based on Johns et al. (2016)'s work. Each sentence in the exemplar set is represented as a pair of vectors in the model. One vector is an unordered set of words constructed as a sum of the vectors representing each word in the sentence. The second vector is the ordered sequence of the words in the sentence, constructed as a holographic representation (Plate, 1995).

Test items are a set of 200 seven-word sentences taken from Johns et al. (2016). Test items have simple syntactic construction and consist of words that occur at least 300 times in the corpus. Test items are presented to the model as an unordered set of words. The model first selects the exemplar sentence most similar to the test item, as measured by cosine between the vectors for the unordered sets. Then, of the 7! possible orderings of the words in the test item, the model selects the ordering most similar to the selected exemplar sentence, as measured by the cosine between the vectors representing the ordered sequences of words. The ordering is judged correct if it matches the original ordering of the words in the test item.

HHM is trained on the full Project Gutenberg corpus. We trained HHM twice: once using a 21 word window, computing bigrams to 21-grams within that window, and once using an 11 word window, computing bigrams to 5-grams within that window. The 5-gram window is standard for the BEAGLE model. Words are represented by either random vectors (Level 0), BEAGLE memory vectors (Level 1), Level 2 memory vectors, or Level 3 memory vectors. At Levels 1, 2, and 3, we test both the 5-gram and 21-gram variants.

To ensure that results are not contingent on a particular sample of 125 000 exemplar sentences, results are averaged across 50 random samples. Mean percent correct across the 50 samples is shown in Figure 2 (Error bars indicate standard error). To test for statistical significance across the seven conditions, we used a repeated measures permutation test.

Level 0 gets a mean of 35.1% of the sentences correct using random vectors, i.e., by selecting the exemplar sentence with the most words in common with the test item.

At Level 1, we find no effect of window size ($p > 0.05$). Level 1 outperforms Level 0 ($p < 0.0001$) with a mean of 57% correct. Level 1 uses BEAGLE memory vectors, i.e., selects the exemplar sentence which has the most semantic similarity to the test item.

Level 2 outperforms Level 1 ($p < 0.0001$), demonstrating the value of third-order associations. Here we find an effect of window size ($p < 0.01$). The 21-gram window gets 59.5% correct to the 5-gram window's 58.6% correct.

At Level 3, we find the 21-gram window again outperforms the 5-gram window ($p < 0.0001$). With the 21-gram window, Level 2 and Level 3 are not significantly different (59.5% vs. 60.0%, $p > 0.05$). With the 5-gram window, Level 3 gets only

55.6% correct, significantly worse than Level 2 ($p < 0.0001$).

Our results show that for the task of ordering words into grammatical sentences, a model that uses third or fourth order associations between words outperforms a model that uses first or second order associations. Our results also show that higher levels of HHM benefit from *n*-grams larger than 5-grams (whereas 5-grams may be sufficient for BEAGLE).

Conclusions

We find that the higher levels of the Hierarchical Holographic Model (HHM) exploit higher-order associations to gain syntactic information. Sensitivity to third order (Level 2) or fourth-order associations (Level 3) reinforces relationships between words that share part-of-speech and improves the model's ability to order words into grammatical sentences.

However, we find that higher levels of HHM are more useful when using larger *n*-grams. At higher levels, HHM progressively loses the ability to make fine distinctions between small *n*-grams as the representations for the words that compose the *n*-grams become increasingly similar. For example, ”she grinned” and ”he smiled” may be represented by identical or nearly identical bigrams at higher levels.

At the same time, higher levels begin to be able to make use of large *n*-grams. At lower levels, large *n*-grams are unique, and thus do not provide useful information about the relationships between words. At higher levels, large *n*-grams are similar to other large *n*-grams. For example, while the 7-gram ”you are as gregarious as a locust” may occur only once in a corpus, at higher levels of HHM, this 7-gram comes to resemble other 7-grams, such as ”he was as strong as an ox”.

Gruenenfelder, Recchia, Rubin, and Jones (2016), modeling word association norms, find that a hybrid model that uses both first and second order associations better matches human data. We note that on the word ordering task, while, on average, Levels 2 and 3 with the 21 word window produced the best results, Level 1 often correctly ordered sentences that Levels 2 or 3 got wrong. We speculate that a model that uses all three levels could outperform a model that uses only one level at a time. We hypothesize that human memory is able to use relations between concepts at varying levels of abstraction as needed to meet task demands.

The Hierarchical Holographic Model is not intended as strictly a language model but as a model of human memory with the ability to detect arbitrarily abstract associations. The present work is a proof of concept of the utility of HHM as a model and preliminary evidence that higher-order associations are relevant to understanding human cognition.

Acknowledgments

We thank Kevin D. Shabahang and D. J. K. Mewhort for the use of their BEAGLE code and Project Gutenberg corpus file. We also thank D. J. K. Mewhort for the use of his server, funded by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC: APA 318). This research has been funded by an Ontario Graduate Scholarship to M. A. Kelly, a National Science Foundation grant (SES-1528409) to D. Reitter, and a grant from Natural Sciences and Engineering Research Council of Canada to R. L. West.

References

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

Barceló-Coblijn, L., Corominas-Murtra, B., & Gomila, A. (2012). Syntactic trees and small-world networks: syntactic development as a dynamical process. *Adaptive Behavior*, *20*(6), 427-442.

Cox, G. E., Kachergis, G., Recchia, G., & Jones, M. N. (2011). Towards a scalable holographic representation of word form. *Behavior Research Methods*, *43*, 602-615.

Grefenstette, G. (1994). Corpus-derived first, second and third-order word affinities. In *Proceedings of the sixth euralex international congress* (p. 279-290). Amsterdam, The Netherlands: Association for Computational Linguistics.

Gruenenfelder, T. M., Recchia, G., Rubin, T., & Jones, M. N. (2016). Graph-theoretic properties of networks based on word association norms: Implications for models of lexical semantic memory. *Cognitive Science*, *40*(6), 1460–1495.

Johns, B. T., Jamieson, R. K., Crump, M. J. C., Jones, M. N., & Mewhort, D. J. K. (2016). The combinatorial power of experience. In *Proceedings of the 38th annual meeting of the cognitive science society* (p. 1325-1330). Austin, TX: Cognitive Science Society.

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*, 1-37.

Kelly, M. A., Blostein, D., & Mewhort, D. J. K. (2013). Encoding structure in holographic reduced representations. *Canadian Journal of Experimental Psychology*, *67*, 79-93.

Kelly, M. A., Kwok, K., & West, R. L. (2015). Holographic declarative memory and the fan effect: A test case for a new memory model for act-r. In *Proceedings of the 13th international conference on cognitive modeling* (p. 148-153). Groningen, the Netherlands: University of Groningen.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*, 623-641.

Princeton University. (2010). About wordnet. *WordNet*. Retrieved from `http://wordnet.princeton.edu`

Rutledge-Taylor, M. F., Kelly, M. A., West, R. L., & Pyke, A. A. (2014). Dynamically structured holographic memory. *Biologically Inspired Cognitive Architectures*, *9*, 9-32.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science*, *29*, 41–78.

Tsuboi, Y. (2014). Neural networks leverage corpus-wide information for part-of-speech tagging. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (p. 938-950). Doha, Qatar: ACL.

Ward, G. (1996). *Moby part-of-speech*. University of Sheffield. Retrieved from `http://icon.shef.ac.uk/Moby/mpos.html`

Linking Memory Activation and Word Adoption in Social Language Use via Rational Analysis

Jeremy R. Cole, Moojan Ghafurian, and David Reitter
jrcole,moojan,reitter@psu.edu
The Pennsylvania State University
University Park, PA USA

Abstract

This paper investigates how cognition facilitates the adoption of new words through a study of the large-scale Reddit corpus, which contains written, threaded conversations conducted over the internet. Parameters for the cognitive architecture are estimated. Using ACT-R's account of declarative memory, the activation of memory chunks representing words is traced and compared to usage statistics sampled from a year of data. Potential values for decay and retrieval threshold are identified according to model fit and growth rates of word adoption. The resulting estimate for the decay parameter, d, is 0.22, and the estimate for the retrieval threshold parameter, rt, lies between 3.4 and 4.5.

Keywords: neologisms, retrieval threshold, decay

Introduction

Language is a communication system that varies among speakers and is constantly changing. Naturally, language occurs in the context of social interaction, and large-scale datasets reflecting language use are a good opportunity to study individual cognition in the social context. It is this context that the cognitive architecture may have evolved to serve.

The aspects of the architecture most linked to the adoption of new words among individual language users is are declarative memory formation retrieval. English is a productive language: new words are invented frequently. In fact, the rate of new word formation has increased in the past century (Lehrer, 2006). Newly introduced words might be used for only a short period of time or may last longer and contribute to large-scale language change. This process relies on speakers taking liberties with their word choice and on speaker communities that facilitate and accept the use of novel words.

In this paper, we model word choice and exposition to words as the result of declarative memory activation (Anderson and Schooler, 1991). This lets us study the cognitive architecture in the context of the social environment, as it presents itself in a very large corpus of web-forum dialogue. As a result, we are able to derive rational parameters for the ACT-R declarative memory module.

Lexical change has been studied experimentally. For example, *naming games* have proven to be a fruitful way to elicit change (e.g., Baronchelli 2011). The dispersion of new ideas has also been observed in large-scale data as well. Hashtags in Twitter are a good example of neologisms that represent memes. Their dispersion dynamics can be surprising in that they appear to be different depending on the topic (Romero et al., 2011). For controversial topics, e.g. in politics, repeated exposure keeps achieving additional adoption (*complex contagion*).

However, to our knowledge, little work has studied word adoption at an individual level through cognitive modeling. We take this as an opportunity to employ rational analysis to fit architectural parameters. While Anderson and Schooler (1991) touched on this, determining certain features of memory that must be true in order to process newspaper headlines. Relatedly, we model the state of memory directly to determine the optimal fit of parameters based on the data.

For a word to be used spontaneously, it must have high enough *activation* to be retrieved. This presents a bit of a conundrum, and perhaps an explanation for why this level of analysis has been avoided: to more highly activate a word, it must be presented, but for it to be presented, someone must successfully retrieve it. Nonetheless, one can assume that there are some people, the originators, for whom the word is more highly active. Then, as these people are relatively few in number, we can still measure the approximate activation for the adopters. This allows us to find the threshold for adoption, and thus guess at the threshold for retrieval.

In this paper, we thus present a simple cognitive model of word adoption. It uses a computational measure of activation and a corpus of the *Reddit* web forum to investigate the role of memory in word adoption. Beyond word adoption, we are interested in using measures of activation to compare to more empirical results, such as frequency. By using such empirical measures across a wide dataset, we can measure accurate values for certain parameters of ACT-R that have only been guessed at based on small-scale experimental results (Anderson, 1983). In particular, we focus on fitting the value of d, the decay parameter, and estimating the value of rt, the retrieval threshold parameter.

There are a few related topics that converge to our research questions. In particular, we are interested in the cognitive mechanisms that cause the adoption of new words (or neologisms) or new ideas in general, as well as the ability to use big data to provide evidence toward parameters in cognitive models. Lastly, most models ultimately provide information about declarative memory elements that already have been presented. This model's novelty, in part, is due to its evaluation of new elements and an evaluation on corpus data.

Related Work

Beyond naming games, which have focused primarily on social factors, there are a few studies on the impact of cognitive factors on word adoption. For instance, Gilhooly (1984) showed that age of acquisition is more important than 'residence times' in naming times. They likewise relied on new words based on their introduction to language. Indeed, age of acquisition has been related to several such experimental paradigms and in many other studies (e.g., Morrison and Ellis 1995). While these studies are interesting, they have not focused on how such factors impact the adoption of new words, just how well they nestle in a single person. Other studies we know of that take memory into account at all also do not take adoption into account (e.g., De Vaan et al. 2007).

Most work that is focused on word adoption at a large scale has focused on *lexical innovation*, which normally has a focus on word forms, rather than memory and time course (e.g., Baayen and Renouf 1996). However, an important component of word adoption is not just whether the word form is easy to learn, but whether it can be retrieved from memory at all.

Previous work has focused on the relationship between memory and traditional measures of activation found in corpora, such as recency and frequency (e.g., Anderson and Schooler 1991). While that work was fundamental, it did not develop estimates for modern ACT-R parameters.

This calls for a cognitive model, as some value of activation should correspond to the retrieval threshold. While this value is used in ACT-R, to our knowledge, there are no papers estimating its empirical value in any field, and we are certainly aware of none estimating it in language.

Cognitive models of language are of course not new. Both comprehension (e.g., Lewis and Vasishth 2005; Ball et al. 2010) and production have been explored (e.g., Guhe 2009; Reitter et al. 2011). Language acquisition has also been explored (e.g., Dörnyei 2009), though it has mostly focused on second language acquisition. This is because it is difficult to acquire realistic human language data at acquisition time. Cognitive models of language acquisition without such strict constraints are much more common (e.g., Pinker and Prince 1988). By focusing on new words, we provide a possible workaround. By using a corpus, we have a lot of data in order to look at certain effects.

Methods

In general, our evaluation relies on comparing the data created by our model of activation with the human data from the corpus. This type of evaluation lets us fit against a large amount of data, not only confirming previous findings about ACT-R but tuning and estimating certain parameters.

Data: Reddit Corpus

Our data set consists of approximately 426GB of Reddit data, ranging from the year 2012 to the year 2014. Reddit.com is a community-driven news aggregation website that mostly contains discussions and ratings on a variety of topics (Bergstrom, 2011). The various communities the topics are organized around are called *subreddits*.

After the submission, people can reply with their thoughts in a *comment*. Users can also comment on these comments. We study these comments. Before applying any of our analysis, we filter out comments in subreddits with a small number of users (defined as 500). As anyone can make a subreddit and invite their friends to join, we wanted to avoid small subreddits that may more closely resemble social networks than communities.

What constitutes a new word?

As discussed, our data spans 2012-2014. In this sense, we came up with a simple way to determine if a word is new: it did not occur in 2012, but it did occur in 2013 or 2014. To ensure we excluded non-linguistic or pseudo-linguistic elements (such as hyperlinks), we excluded every token that did not entirely consist of alphabetic characters. To ensure we excluded typos or words that only had meaning in a single conversation, we used a simple arbitrary cutoff of one hundred occurrences. We claim that these three requirements are sufficient to define a new word, or a neologism. Some example words can be found in Table 1. In total, we found 3545 words matching these criteria.

There are two important limitations to this. Some elements of this set of words only have meaning to members of that subculture; some of them may have even fallen out of use already. Secondly, some of these words originate from a culture external to Reddit. In some of these cases, the usage of the words is still novel: Square Cash, a financial product, was frequently referred to as *squarecash* by Reddit users. Others, however, are strictly adoptions, such as Chromecast. Thus, we will refer to these as *first adoption* events. However, the cutoff for number of occurrences does indicate that these are true adoptions, not simply one-off usages.

While some of the first adoption events are origination events, all of them are a discussion of something new. The first discussion of a new idea has social consequence. In Reddit, people receive both explicit and implicit rewards for social acceptance, through the karma mechanism. We will use *adoption* to refer to any usage of a new word by a subreddit, using origination or first adoption for the first subreddit to adopt it, and *later adoption* for later usages.

Table 1: A table containing examples of new words, along with the subreddit it first appeared in, and a subreddit that it appeared in later. Both of these events are treated the same in our model.

Word	First Adopter	Later Adopter
dogetips	dogecoin	funny
misanderkirby	AdviceAnimals	AskReddit
peshka	gaming	Warthunder
gamecribs	leagueoflegends	counterstrike
squarecash	economy	Bitcoin
watchapps	pebble	Android

Cognitive Model

We see the adoption process as one that is governed by declarative memory (DM). A word is added to the lexicon (in DM), and through repeated presentations, it becomes available. We conjecture that initial use of the word is aided by short-term memory, from direct copying, or aided by cues (which spread activation, if seen from an ACT-R perspective). At some point, activation of the memory trace in the modeled individuals reaches the point where this word is retrievable without the help of cues. This *retrieval threshold*, as well as the function governing the gradual rise in activation, are central to this model, and we will estimate their parameters from the data.

We compute the activation of adopted words using the base-level learning equation defined originally by Anderson (1983).

$$bll(x) = \log\left(\sum_{i\in P_x} t_i^{-d}\right)$$

In this equation, x represents any symbol, a word in our case, and P_x refers to the list of x's presentations. So t_i is the time from that presentation to the present. Naturally, for something with as many presentations as any given word, it is infeasible to computationally manage that sum. However, the full equation can be approximated using only the total number of presentations and the k most recent presentations and $n_x = |P_x|$ (Petrov, 2006).

$$bll(x) \approx \log\left[\sum_{i}^{k} t_i^{-d} + \frac{(n_x - k)\left(t_{n_x}^{1-d} - t_k^{1-d}\right)}{(1-d)\left(t_{n_x} - t_k\right)}\right]$$

Petrov (2006) shows that the equation is close even for $k = 1$. As the amount of events in the Reddit corpus is very large, computing the many previous events is computationally expensive. Thus, we relied on this approximation and only kept track of the single most previous event. Note that causes the left sum to collapse to t_k^{-d}.

Table 2 gives brief descriptions of how each parameter was computed. For the constant d, we initially examine two values: 0.5, the ACT-R default (Bothell, 2004), and 0.16, as found by Vasishth and Lewis (2004).

Activation, in ACT-R, is composed of the base-level learning function (as above), in addition to spreading activation from cues and noise.

Table 2: The parameters of our activation equation and a description of how we computed them

Parameter	Description
n_x	The total number of occurrences of that word across Reddit
t_k	The time in between the current usage and the previous usage
d	The decay parameter, 0.5 or 0.16
t_{n_x}	The amount of time since the first usage of the word
k	1

Retrieval Threshold

The retrieval threshold rt defines the point of total activation for a memory trace to be retrievable. Obviously, many assumptions influence this parameter, to include how many times we assume the item to have been used in the past, outside of the context of the experiment at hand. As a consequence, no canonical value for this parameter is available. However, by looking at new words, which not based on past experience, and influenced less by external influence, we may be able to approximate this threshold.

Filtering the data

In order to get a realistic estimate of rt, we had to look at the pattern of the data. In particular, we wanted to see at what point words were adopted. However, the time course data taken naively is somewhat biased: because each word is adopted at a different point, and our data is only for just over 400 days, the number of words being evaluated is different at each day. Thus, to get the full range of effects while still avoiding bias, we only included words with over 400 days of data, and excluded all data beyond 400 days.

Relating adoption to declarative retrieval

Defining exactly what it means when the word is 'adopted', and thus has activation above rt is non-trivial, which is likely why there is so little information on it throughout the literature. However, our hypothesis was fairly simple: once activation is high enough that retrieval is possible, the frequency of usage will rapidly expand, as it usage no longer relies on referencing external sources. We will estimate that point by observing the pattern of results and finding where the derivative increases.

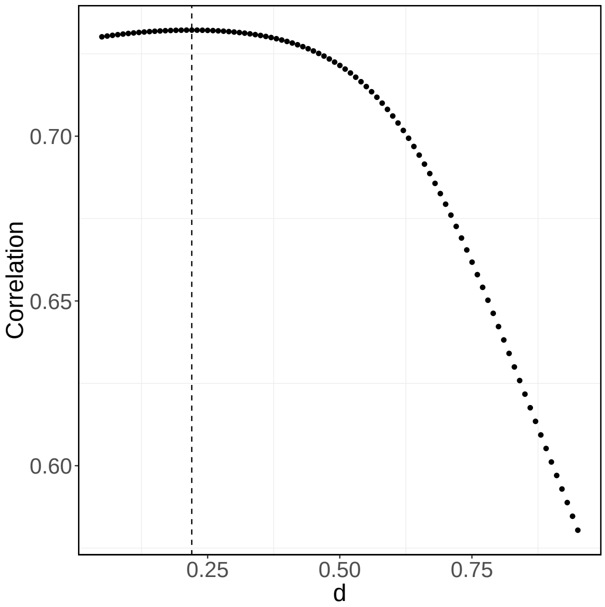


Figure 1: Correlations between word usage and the calculated activation for different d values, ranging from 0.05 to 0.95.

Fitting the decay parameter

In order to fit the decay parameter, we show which measure of activation, computed as described earlier, best predicts usages per day. The usage of each word every day is an empirical metric that should show how active that word actually is. We ask whether the ACT-R default (.5) or the value found by Vasishth and Lewis (2004) (0.16) yields a better fit, or if a different value would be found altogether. A grid search between between .05 and .95 was used, optimizing the activation's correlation with usages per day while using that value.

Results

By closely examining the data, we are able to see a clear inflection point for rt, as well as a pattern in the fit of activation.

Decay parameter

The activation for each word occurrence was calculated for different values for d. Figure Figure 1 shows the correlation between activation and observed word usage as a function of d. The correlation peaks at 0.22 (see Figure 1). Note that this methodology is approximate and assumes, e.g., $k = 1$. So, this disagrees with a value of 0.5, but, largely, agrees with 0.16 reported in the literature.

Retrieval Threshold

After showing the pattern of usages per day over time, there is a point where the function changes from oscillating but linear to a more exponential curve. In other words, we see a change in the derivative as the word is 'adopted', leading to larger gains as the word is able to be used more freely. This is around 250 days in, as shown in Figure 2. Then, looking at the activation over time

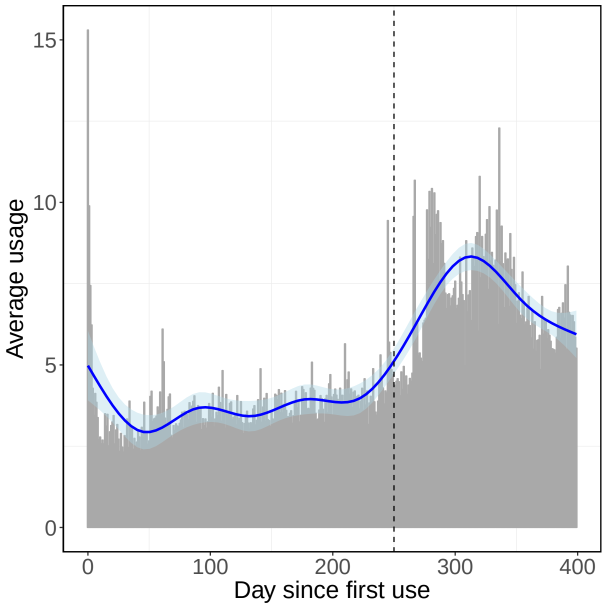


Figure 2: Average usage of new words per day, over time. Day 0 represents the day on which the word was first adopted. The dotted line marks the day where the derivative has clearly changed, around day 250. This inflection point represents the adoption event.

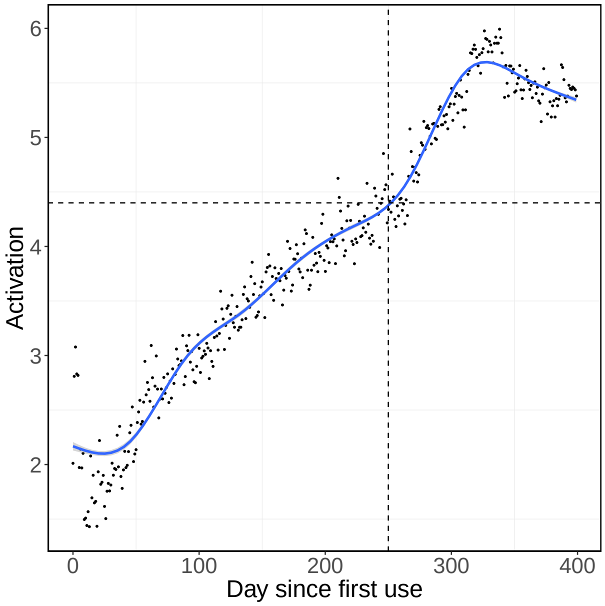


Figure 3: Base-level activation for $d = 0.16$ calculated for each word occurrence over time. Day 0 represents the day on which the word was first adopted. The vertical dotted line is at the same day as the inflection point shown in Figure 2; the horizontal line shows the activation for this value of decay, about 4.4. This value represents a possible value for rt.

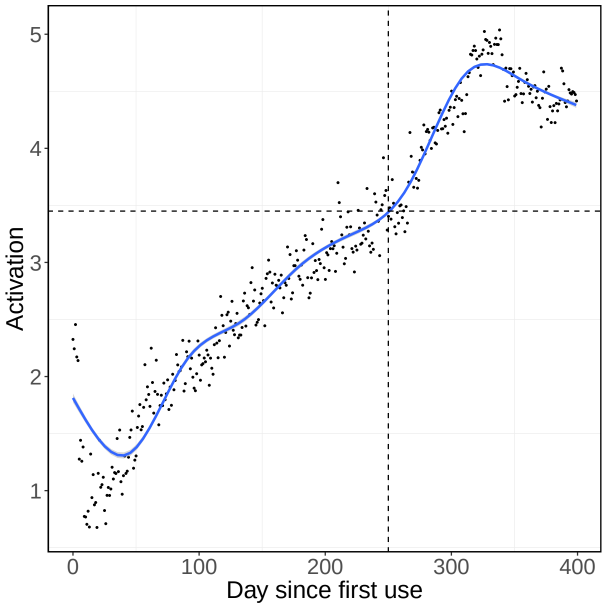


Figure 4: As Figure 2, for $d = 0.22$. The inflection is near activation 3.45. This value represents a possible value for rt.

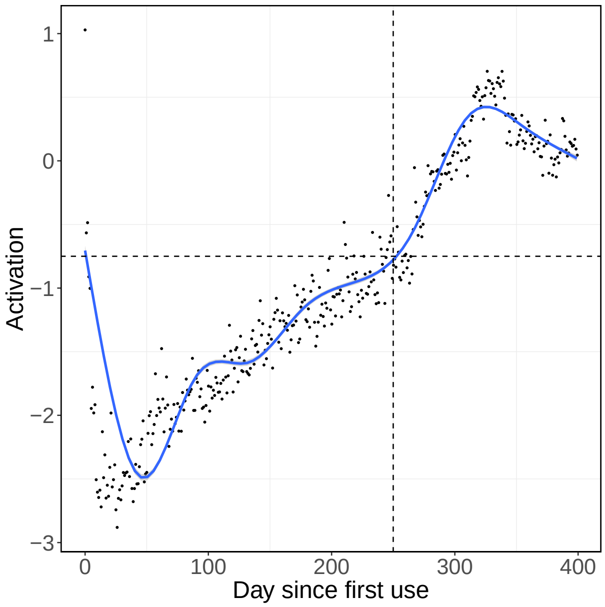


Figure 5: As Figure 2, for $d = 0.50$. The vertical shows the time of the same inflection point shown, here at activation -0.75

we are able to find an approximate value for activation where that changes, approximately 4.4 (see Figure 3). However, this is based on $d = 0.16$, we also estimate it at about 3.45 for $d = 0.22$, our own empirically found value (see Figure 4). Lastly, we compute it for the ACT-R default of $d = 0.5$, and find it to be -0.75 (see Figure 5). In general, the values found for activation for $d = 0.5$ also suggest it is not particularly suitable, as a negative activation should not be retrievable at all. These values correspond to reasonable guesses for rt. In particular, based on our methodology, we do not claim either 250 or the values for rt are the exactly correct values; however, based on the results of this study, they present reasonable constraints for an estimation of rt. In particular, we chose 250 as a point that is clearly starting an ascent and is significantly above the earlier trend.

Discussion

This paper has used ACT-R memory retrieval on data that reflects long-term language use in a social context. With this, we examine two critical parameters in declarative memory retrieval: decay (d) and the retrieval threshold (rt).

With this idea, we follow the idea of rational analysis: can we observe environmental data to draw conclusions about the individual cognitive system, assuming that it has evolved to be optimally adapted to process information from this environment while contributing to the production of such data in the first place. However, what is perhaps more unique to our approach is that we observe language behavior in a large-scale and long-term social context.

As for d, we obtain a best fit at a very different rate of decay than what is observed in controlled behavioral experiments. Of course, many standard experiments on memory retrieval use words, so language is not necessarily unique to the data in the present study. For language in context, as in the Reddit data, the slower decay of language could be due to the heavy semantic relatedness in language, which causes constant spreading activation. Even new words are largely derivative of old ones, borrowing phonetics, ideas, roots, or at least lexicography. Naturally, when dealing with models over time courses that make sense for language, the difference between 0.50, and 0.22 is substantial. We provide additional evidence that the value could be different.

One explanation of the slower decay that we observe may be reinforcement through cognitive function that is not observed: in other words, people do not write a word in a Reddit post every time they think of it. The other consideration is the time-course of word adoption: we have examined language use through about one year (the *social band*, Newell, 1990), while ACT-R's declarative memory framework is currently best suited to seconds and minutes (the *cognitive band*).

Still, an important constraint is that Reddit consists of written language and conversations can span several days. This could be a possible problem, as it is unclear how applicable forum discourse is to laboratory studies. A similar study performed on a corpus of real-time communication could be informative.

Our method for fitting rt opens up many interdisciplinary questions beyond the scope of this exploratory study. What is the range of the inflection point when considering multiple samples from the corpus, and from other corpora? Are there meaningful *bands* (to use Newell's term) identifiable in the behavior of activation before day 50 and after day 300? Does language use in context not follow the patterns of cue-based memory retrievals found in dedicated experiments? How does the socio-informational network contribute to a changing inflection point and a critical mass necessary for contagion? Nonetheless, we acknowledge the limitations in the approach for measuring rt, though leave it to future work to determine a more precise measurement, perhaps based on fitting a model of retrieval to the corpus data.

Conclusion

In this paper, we use a cognitive model of memory that models the process of learning new words. By evaluating the model on corpus of social, contextual language use, we are able to model large amounts of human data, which gives us insight into the process and lets us examine the ACT-R model of memory itself. By comparing against an empirical measure of 'activation', we are able to correlate activation was computed by ACT-R in order to determine a reasonable value for d in language. We are also able to compute a reasonable estimate for rt, a parameter that has yet to be fitted in language or other domains. Specifically, we found that the bounds for a retrieval threshold we found lie somewhere between 3.4-4.5; the decay d at .22 or lower – unlike in many other studies.

Acknowledgements

This project was supported by the National Science Foundation (BCS-1457992 and IIS-1459300).

References

Anderson, J. R. (1983). A Spreading Activation Theory of Memory. *Journal of Verbal Learning and Verbal Behavior*, *22*, 261–295.

Anderson, J. R. & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*(6), 396–408.

Baayen, R. H. & Renouf, A. (1996). Chronicling the times: productive lexical innovations in an english newspaper. *Language*, 69–96.

Ball, J., Freiman, M., Rodgers, S., & Myers, C. (2010). Toward a functional model of human language processing. In *Proceedings of the 32nd annual meeting of the Cognitive Science Society* (pp. 1583–1588). Portland, Oregon.

Baronchelli, A. (2011). Role of feedback and broadcasting in the naming game. *Physical Review E*, *83*(4), 46–103.

Bergstrom, K. (2011). Don't feed the troll: shutting down debate about community expectations on Reddit.com. *First Monday*, *16*(8).

Bothell, D. (2004). *ACT-R 6.1 reference manual*. Tech. Rep., 2004.

De Vaan, L., Schreuder, R., & Baayen, R. H. (2007). Regular morphologically complex neologisms leave detectable traces in the mental lexicon. *The Mental Lexicon*, *2*(1), 1–23.

Dörnyei, Z. (2009). *The psychology of second language acquisition*. Oxford: Oxford University Press.

Gilhooly, K. (1984). Word age-of-acquisition and residence time in lexical memory as factors in word naming. *Current Psychology*, *3*(2), 24–31.

Guhe, M. (2009). Generating referring expressions with a cognitive model. In *Proceedings of the Workshop Production of Referring Expressions: Bridging the Gap between Computational and Empirical Approaches to Reference*. Amsterdam, Netherlands.

Lehrer, A. (2006). Neologisms. In K. Brown (Ed.), *Encyclopedia of language & linguistics (second edition)* (Second Edition, pp. 590–593). Oxford: Elsevier.

Lewis, R. L. & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*(3), 375–419.

Morrison, C. M. & Ellis, A. W. (1995). Roles of word frequency and age of acquisition in word naming and lexical decision. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*(1), 116.

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

Petrov, A. A. (2006). Computationally efficient approximation of the base-level learning equation in ACT-R. In *Proceedings of the seventh international conference on cognitive modeling* (pp. 391–392).

Pinker, S. & Prince, A. (1988). On language and connectionism: analysis of a parallel distributed processing model of language acquisition. *Cognition*, *28*(1), 73–193.

Reitter, D., Keller, F., & Moore, J. D. (2011). A Computational Cognitive Model of Syntactic Priming. *Cognitive Science*, *35*(4), 587–637.

Romero, D. M., Meeder, B., & Kleinberg, J. (2011). Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In *Proceedings of the 20th international conference on World Wide Web* (pp. 695–704). ACM. Hyderabad, India.

Vasishth, S. & Lewis, R. L. (2004, July). Modeling sentence processing in ACT-R. In *Proceedings of the ACL workshop incremental parsing: bringing engineering and cognition together* (pp. 82–87). Barcelona, Spain.

Examining Working Memory during Sentence Construction with an ACT-R Model of Grammatical Encoding

Jeremy R. Cole
The Pennsylvania State University
University Park, PA
jrcole@psu.edu

David Reitter
The Pennsylvania State University
University Park, PA
reitter@psu.edu

Abstract

We examine working memory use and incrementality using a cognitive model of grammatical encoding. Our model combines an empirically validated framework, ACT-R, with a linguistic theory, Combinatory Categorial Grammar, to target that phase of language production. By building the model with the Switchboard corpus, it can attempt to realize a larger set of sentences. With this methodology, different strategies may be compared according to the similarity of the model's sentences to the test sentences. In this way, the model can still be evaluated by its fit to human data, without overfitting to individual experiments. The results show that while having more working memory available improves performance, using less working memory during realization is correlated with a closer fit, even after controlling for sentence complexity. Further, sentences realized with a more incremental strategy are also more similar to the corpus sentences as measured by edit distance. As high incrementality is correlated with low working memory usage, this study offers a possible mechanism by which incrementality can be explained.

Introduction

Working memory has long been thought to play an important role in language processing (e.g., Gibson, 1998). In language production, one important question concerns *grammatical encoding*, the process by which words are combined into sentences. In this paper, we make progress in understanding the interaction of working memory with strategies and representations that are needed for grammatical encoding.

One strategic decision that is crucial to grammatical encoding is how incremental the process is: are words planned in the exact order they are output, or are other mechanisms at play? There are, obviously, opportunities to leverage representational insights from computational linguistics and algorithmic choices known in natural-language generation. Models of grammatical encoding thus far have been either too general or too specific to both make clear predictions that are testable with behavioral methods or against known effects.

Computational implementations of linguistic models (e.g., Steedman, 2000) are geared towards performance, not explanatory value, which makes it difficult to evaluate their demand for limited cognitive resources and to determine their interactions with general cognition. On the other hand, connectionist models often aim to be more agnostic to the linguistic task (e.g., Dell et al., 1999), making them challenging to interpret once they are trained on data. In short, the general motivations to engage in a combination of cognitive modeling and computational linguistics apply to grammatical encoding.

With the availability of large-scale data, language models should strive to explain as much data as possible. Reusing syntactic alternatives, such as the difference between double object and prepositional object, can only take the field so far in explaining the richness of human discourse. Conversely, if our models are high coverage but not cognitively plausible, the performance of the model may be good from an engineering perspective, but the model cannot be said to have explained any of the data from a scientific perspective.

Big data, used appropriately, also allows us to contrast different model variants in terms of their explanatory power. This can lead to incremental improvements. In short, we call for a new type of cognitive modeling where it is possible: instead of modeling a relatively small number of experiments surrounding a phenomenon, we model a large amount of the raw data produced by the phenomenon itself. For our task, we evaluate against a corpus.

In this paper, we advance towards such a computational cognitive model of grammatical encoding. The implemented model that we will discuss has clear, interpretable representations in the form of Combinatory Categorial Grammar (CCG, Steedman and Baldridge, 2011). It is cognitively plausible and implemented in an empirically validated framework, ACT-R (Anderson et al., 2004). It is empirically testable, as the model can produce output for any target sentence, allowing competition among alternative models. Our model in particular examines how incrementality in syntax, working memory availability, and working memory usage can improve or worsen the model's fit to linguistic data.

Related Work

This paper builds on a rich body of work from both psychology and linguistics attempting to characterize the language production process.

Grammatical Encoding

There are many ways to discretize the steps of the full process of language production. For instance, we could say after an idea is formulated, it is grammatically encoded and then phonologically encoded (Bock and Levelt, 2002). In turn, grammatical encoding could consist of lexical selection, function assignment, and constituent assembly. The first stage maps ideas to words; the second stage maps words to parts of speech, and the last stage combines these lexical-syntactic units (hereafter lexsyns) into constituents. As syntactic trees are formed by recursively combining constituents, this process eventually leads to a sentence. Thus, the full process of grammatical encoding transforms ideas, or semantics, into a realized sentence.

Working Memory and Language Production

The precise effect of memory on syntactic processing has not been the focus of previous studies. Nonetheless, some constraints have been proposed on sentence formulation, including showing that a higher span can decrease certain types of grammatical errors (Hartsuiker and Barkhuysen, 2006; Badecker and Kuminiak, 2007). Further, Slevc (2011) suggests working memory load can affect the incrementality of a sentence. However, the discussion of representations in working memory for the function assignment and constituent assembly process is a matter of linguistic theory. We turn to Combinatory Categorial Grammar (Steedman and Baldridge, 2011), which provides a possible representation, and we show how it could map to the psychological architecture of attention, processing and memory.

Theories of Incrementality

V. S. Ferreira (1996) makes an argument for incrementality, based on the observation that competitive syntactic alternatives facilitate production rather than making it more difficult. An incremental account of sentence realization would predict such an effect, as syntactic “flexibility” introduced by the alternatives makes it easier to find a workable syntactic decision. By contrast, without incremental commitment to each structure, competing material slows down the process, because it would lead to a combinatory explosion. Further results, however, relativize this account when it comes to the syntax-phonology interface (F. Ferreira and Swets, 2002). Incremental production is possible, but it is “under strategic control”; it depends on semantic information, and it could be modulated by external factors, such as stress.

Based on the literature, we assume that incrementality in grammatical encoding may be graded: the degree to which a sentence is realized incrementally may vary based on certain cognitive factors. However, the literature has yet to address how speakers (or comprehenders) might use the limited memory resources available to guide the attendant strategic choices surrounding incremental realization. Our corpus-driven model has the potential to explain this by contrasting models with different available working memory, and by examining actual working memory use, as well as by measuring the activation and availability of linguistic structures.

Background

Our model relies on the unification between a linguistic theory (CCG) and a cognitive framework (ACT-R), which will be explained in turn in the following section.

Linguistic Theory

Combinatory Categorial Grammar (CCG) is a grammar formalism (Steedman and Baldridge, 2011). While it was not conceived as a purely psycholinguistic theory, interpreting it as such has a few important consequences. Most importantly, after a syntactic operation, the representations are simplified. This is as opposed to other grammar formalisms, where combination always results in a more complicated representation, e.g. Tree-Adjoining Grammar (Joshi and Schabes, 1997). In general, grammar formalisms operate based on *types*, such as noun phrase, and *rules*, which are methods for combining the types into a sentence. See Table 1 for a demonstration of CCG’s combinatory rules. See Figure 1 for an example of how these rules can create sentences.

Table 1: The four basic rules of CCG, which specify how syntactic types can be combined. They are the rules by which types can be systematically combined into a sentence. The left-hand side specifies two types, each of which are recursively composed of one or two types (e.g. X/Y is one type). The right-hand side specifies the resulting type of the operation on the left.

Forward Application (>):	$X/Y > Y = X$
Backward Application (<):	$Y < X \setminus Y = X$
Forward Composition (>>):	$X/Y >> Y/Z = X/Z$
Backward Composition (<<):	$Y \setminus Z << X \setminus Y = X \setminus Z$

ACT-R

ACT-R is a general theory of cognition (Anderson et al., 2004). ACT-R, combined with a linguistic theory like CCG, can provide a unification of computational modeling, cognitive science, and linguistics. ACT-R’s basic system for writing models involves chunks and production rules, where chunks represent declarative memory and production rules represent procedural memory. In the following sections, we discuss how we infer the chunks from a corpus.

Methods

We create a model that is automatically derived from the syntactic and lexical information present in 1,200 sentences sampled from the *Switchboard* corpus (Godfrey et al., 1992). Switchboard is a spoken language corpus of two strangers having a phone conservation about a provided topic. We then run this model with no interruptions or constraints, using the unordered bag-of-words from the corpus sentences as input as an approximation of the meanings (one sentence at a time), and expecting sentences or sentence fragments as

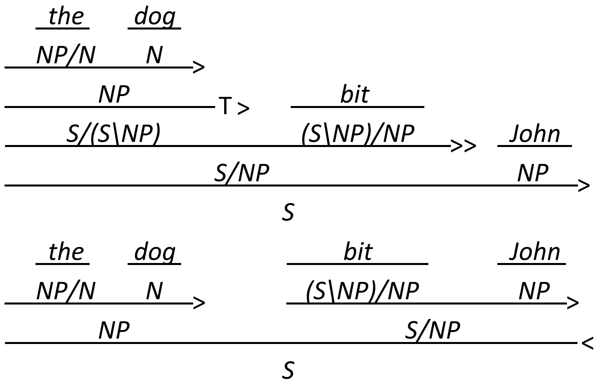


Figure 1: Two contrasting CCG derivations: The top is more incremental (right-branching) than the bottom. Note that the $T>$ is normally used to mark non-standard derivations, which are usually more incremental.

output. The model's process is recorded in the form of syntax trees (”derivations”) for further analysis, as these derivations reflect the strategy applied by the model to produce the sentences. The model's performance under different working memory conditions will be evaluated by comparing to each original sentence. Thus, the core task of the model is to recover the original ordering of the words in each sentence.[1]

Model

Our model is implemented in *jACT-R* (Harrison, 2005), a full Java implementation of the ACT-R theory (Anderson et al., 2004). This was primarily due to convenience, portability, and scalability, rather than any difference in theoretical predictions between jACT-R and the core Lisp ACT-R.

The model's combinatory mechanism is based on CCG. As CCG specifies clear symbolic and procedural components, it maps naturally to chunks and production rules. The exact mapping will be discussed in the following sections. As discussed earlier, using CCG as the combinatory mechanism of the cognitive model means that combinations will reduce the current memory use. We acknowledge that such predictions should match data on working memory, though we don't see such a prediction as out of line with current ideas about chunking (Conway et al., 2005).

The model is generated from a corpus. The model's goal is to encode all of the sentences found in the corpus into declarative memory and production rules. A wide range of models can be created in this way; however, our empirical evaluation is based on a model learned from a subset of the Switchboard corpus. The chosen sentences use more frequent syntactic types and are of shorter length. Then, the model learns the words and potential syntactic types from the raw text and CCG annotations. These learned syntactic types serve as possible function assignments for the words. Importantly, this is all the model learns: the production rules are encoded with no knowledge of the sentences.

In short, the current model forms a sentence by combining lexical-syntactic chunks together. Out of simplicity, it chooses what to combine greedily. Importantly, it treats no words, types, or rules as special, and it has no knowledge of what words or types should go together beyond the constraints of CCG. Nonetheless, simply following CCG rules can lead to unidiomatic sentences and potentially even ungrammatical sentences by violating certain thematic constraints. This is true of the presented model. However, the full expressive range of syntactic and lexical constructions found in a corpus requires substantial learning, which is out of scope for the present paper. Thus, while many constructions of our model are unidiomatic, we provide a baseline for future work to be evaluated against. Randomly selected example constructions by the model can be found in Table 2.

Declarative Memory

Declarative Memory (DM) is composed of a few simple chunk types, described below. The basic organizational scheme has Sentences composed of Lexsyns, and Lexsyns composed of a single Word and several Types that the word can be in a given context.**Words** simply have a name, which corresponds to its lexical information (e.g. family).

Types are an arbitrarily complicated CCG type. The types that exist in DM are the types that are used in Switchboard CCG derivations of our chosen sentences.

Lexsyns associate a Word with some number of Types. These associated types are taken from the function assignments of each word in the Switchboard CCG derivations of our chosen sentences. The types are ordered from most common to least common, which would mean more common types would be selected if all else is equal.

Sentences are normally in the goal buffer. Thus, the sentence contains the current state of grammatical encoding. If we think of the goal buffer as working memory, then differing the slots available to realize a sentence corresponds to different predictions about working memory availability. Additionally, the sentence chunk also contains the input for the task. However, this is more of a limitation than a theoretical commitment: due to our focus on grammatical encoding, we had to assume the previous tasks of idea generation and lexical selection were complete. In reality, it is likely that all three tasks overlap to some extent.

Table 2: Example sentences produced by the model. The ‘target’ is the actual sentence from the corpus, while the realization is what the model produced. The quality of the model's output varies.

Realization	Target
downhill going like everybody	but then they started going downhill like everybody else
you fire never something unless anybody ’re caught	they never fire anybody unless you ’re caught doing something illegally
still taxes raise probably and	i think he can probably raise taxes and still get elected
i then and decided i like author this	and then i decided i like this author
are school working you	are you working anywhere while you are going to school

Production Rules

We define a small set of about ten production rules (which are compiled into several thousand production rules through an automatic process, which we will not describe in detail here). Depending on the production, the architecture will choose an appropriate rule; there is no predefined algorithmic flow. The model's production rules fall into three basic categories.

1. **Syntax Rule Application** This production rule may fire if Working Memory contains at least two Lexsyns whose types would follow the constraints of at least one CCG rule.

[1] Due to our lack of test set, the careful reader may note that it would be possible to overfit. However, our model does not learn word orderings directly from the corpus, instead only learning syntactic types: indeed, we are much more interested in the effects of working memory and grammatical encoding strategy.

If so, it initiates Rule (3) to determine the result of the rule application.

2. **Goal buffer modification** (A) Move word from Input to Working Memory: This production rule can only fire if there is space in the goal buffer for it to be added. It simply deletes the word from the input, and initiates Rule (3) to retrieve its function assignment. (B) Flush: If no other rules apply, the model will flush, clearing its retrieval buffer or working memory to try again. (C) Resolve Syntax Rule: This deletes the unnecessary entries in working memory after the resulting type is known, as the two entries are combined into the single CCG entry representing the result of applying the rule.

3. **Lexical Retrieval from DM** Retrieve the possible function assignments of a word (its lexsyn), or retrieve the type resulting from the application of a syntax rule.

Input

Due to our focus on grammatical encoding, the input to the model is a bag of words generated from the target corpus sentence. To be clear, the model does not order the words its given as input, instead it only combines two words if its possible under CCG using their current function assignments. Thus, not every output sentence uses every word in the input.

Experiment

In our experiment, we use the model to ask how the size of verbal working memory relates to the fidelity of the produced sentences, and how this interacts with the strategy for grammatical encoding.

Conditions

Working Memory (WM) We contrast two basic versions of the model with 3 and 5 working memory slots, respectively. We consider these values as realistic lower and upper bounds of working memory capacity as found in language tasks (Daneman and Carpenter, 1980). This is implemented simply by limiting the number of slots in the Sentence chunk, so the model has less available working memory to use to combine Lexsyns. We distinguish working memory span (controlled) from actual working memory usage (observed).

Dependent Variables

Branching Factor We see grammatical encoding as a process that is quite flexible: the set of production rules, and the absence of a fixed algorithm (and order in which they are applied) is commensurate with that (as well as with ACT-R as a cognitive architecture framing the model). Strategies emerge as a result of the available cognitive resources, such as WM, and, ultimately (not modeled) the success of rule sets. We measure an important aspect of the strategy: incrementality, as determined by branching factor: The more right-branching a syntax tree is, the more incrementally it was realized.

We define two basic metrics for measuring branching factor. The unweighted branching factor (UBF) is the number of right-branching decisions compared to the number of total decisions. The weighted branching factor (WBF) takes into account how far up the syntax tree the decision was made; it short, it sums all of the subtrees rather than simply comparing the decisions. An example tree and computation can be found in Figure 2, which is an syntax tree created from the model's syntactic decisions. Alternatively, to reference the CCG derivations from earlier in Figure 1, the top derivation has a WBF of 3 and a UBF of 7, while the bottom derivation has a WBF of 1.0 and a UBF of 1.0. These values are not on the same scale: 1.0 is the mean for WBF, but 0.2 is the mean of UBF. Both metrics correlate with each other and higher values represent more incremental constructions.

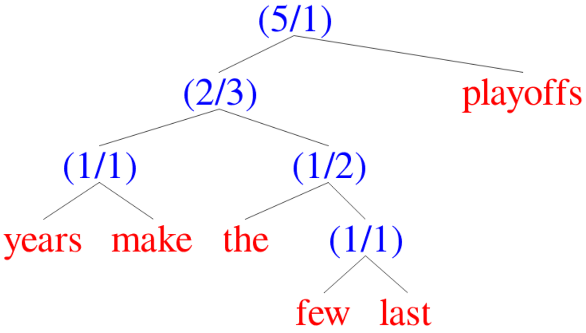


Figure 2: An example of the syntax tree of actual output from the model. To compute the weighted branching factor (WBF), sum the numbers in parentheses for the left and the right, then divide the sum of the left numbers by the sum of the right numbers. The numbers indicate the number of leaves in the right and left subtrees. This computes to 10/8, or 1.25. The unweighted branching factor (UBF) divides the total number of left leaf nodes (in this tree, 3) by right leaf nodes (in this tree, 3), getting a branching factor of 1.0.

Working Memory Usage (WMU) This is based on the maximum amount of slots the model used while realizing a sentence. This includes slots for retrieval and all lexsyns stored in the working memory portion of the goal buffer. We additionally compute the adjusted working memory usage (AWMU), which takes into account the length of the sentence, as longer sentences could possibly require additional working memory, especially if constructions tend to be less incremental.

Edit Distance This measure evaluates fidelity of the model output, i.e., match between the result and the input sentence is computed using Levenshtein distance (Levenshtein, 1966). An edit distance in general is the number of changes (additions, swaps, and deletions) to transform one list into another one: in this case, a sentence is treated as a list of words. Thus, it is a measurement for how dissimilar two sentences are from each other. If the model produces multiple fragments rather than a single utterance, the distances are averaged. We chose edit distance as a metric to ensure the model's trace of syntax was being measured, rather than simply the meaning. It correlates well with metrics used to evaluate natural language processing tasks (e.g., Lin, 2004). We define the edit distance between the two sentences as the model fit.

Results

We examined the correlations between the branching factor, working memory usage, and fit, as measured by edit distance between the realization and the target sentences. We analyze the influence of observed branching factor, available WM and observed WM usage separately.

Both branching factor metrics (UBF and WBF) were found to be significant with a negative effect on edit distance, implying more incremental constructions produce realizations more similar to the initial sentences ($p < 0.001$). Conversely, working memory use (WMU) was found to have a positive effect, implying increased working memory usage decreased fit ($p < 0.001$). This had an even larger effect for adjusted working memory use (AWMU), implying minimizing working memory usage was especially important for longer sentences. Branching Factor and Working Memory usage (all metrics) were also significantly correlated ($p < 0.001$).

Table 3: Individual linear models correlating predictors with edit distance in the WM=3 condition.

	WBF	UBF	WMU	AWMU
p-value	< 0.0001	< 0.0001	0.007	< 0.0001
effect	−0.221	−0.168	0.057	0.074
Intercept	0.873	0.763	0.507	0.610
r^2	0.056	0.025	0.162	0.065
df	1038	1038	1038	1038

Because increased working memory usage is correlated with decreased fit, this begs the question of whether that is because sentences with lower working memory usage requirements are easier, or whether using more working memory directly decreases fit. As sentences have different working memory requirements, the ones with lower requirements could just be easier to realize incrementally, possibly reducing production errors. The five-slot model helps elucidate this.

In the five-slot condition, the model performs slightly better, even though it uses more working memory on average by both metrics. However, it is also more right-branching than the other model by both metrics.

Table 4: Individual linear models correlating predictors with edit distance in the WM=5 condition.

	WBF	UBF	WMU	AWMU
p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
effect	−0.162	−0.228	0.120	0.041
Intercept	0.915	0.773	0.427	0.685
r^2	0.079	0.536	0.092	0.015
df	1038	1038	1038	1038

Discussion

The most interesting take-away from this is that higher working memory usage, which was previously associated with fewer speech errors, is associated with worse fit to the corpus data in our model. This could be because increased working memory usage, rather than alleviating stress caused by low-resources, causes the realizer to garden-path itself. By allowing itself to work breadth-first, it can potentially make syntactic choices that won't eventually lead to a good utterance. The branching factor could partially be a result of this: having a higher right-branching factor should lead to lower working memory use, as new elements are added to the current state, rather than built up in another way. However, it could also be a simple consequence of the fact that since language is outputted in order, it's easier to combine it in order, thus allowing earlier outputs. Importantly, this result indicates the effect of *using* less working memory when all else is equal. It does not indicate the effect of *having* less working memory available. An important caveat then, is that this effect could simply be explained with the observation that easier sentences use less working memory.

Having working memory available when needed clearly improves fit, even though in general, using more working memory worsens fit. Varying WM capacity does not change the general strategy of grammatical encoding, which prefers to use less working memory and more right-branching constructions. Still, the model with less working memory was less right-branching. This could perhaps be because without additional working memory available, it sometimes had to settle for an inferior strategy, perhaps explaining its fit decline, in line with work such as Slevc (2011). We consider the lower fit of the lower working memory model to be in line with previous research, which leaves open as a possible avenue for future experimentation the correlation of lower working memory usage to higher fit.

We consider both of these results to be compatible with the hypothesis of strategic incrementality. More incremental processes require less working memory. This is because lexsyns can be combined and outputted, freeing space. Moreover, reducing working memory usage is normally used as a possible argument for why incremental strategies might be preferred. That still leaves two basic possibilities: (1) Speakers prefer to use constructions that are possible to realize more incrementally, or (2) speakers attempt to realize all constructions as incrementally as possible. We have reason to believe, from F. Ferreira and Swets (2002), that (2) is not the case, unless the speakers are under some stress to speak as quickly as possible. Possibly, (1) can be fairly easily examined from frequency rates, though we are unaware of work doing so.

Table 5: Paired t-tests between WM=3 and WM=5 conditions.

	WBF	UBF	WMU	AWMU	dist
Cond1-Mean	1.050	0.132	3.156	1.721	0.743
Cond2-Mean	1.023	0.105	2.703	1.575	0.748
p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001

By limiting working memory directly, we are able to demonstrate through the model that working memory is critical to grammatical encoding: limiting it or having less of it increases errors experimentally and reduces the fit of our model. While our task was naturally not perfectly analogous to experimental work, it does provide converging evidence in this discussion (Hartsuiker and Barkhuysen, 2006; Badecker and Kuminiak, 2007).

However, actually using less working memory on any sentence is correlated with increased model fit, even after controlling for effects of sentence length or complexity. There are several possible explanations for this. For instance, pushing working memory to capacity could be more likely to cause errors, as speakers retrieve too many lexsyns that can't be combined, forcing themselves to flush, thereby losing track of part of the sentence. Conversely, each sentence may dictate an minimum amount of working memory needed to realize a sentence even in an incremental fashion. In that case, the model predicts, testably, sentences with a lower minimum work memory requirement will have fewer production errors than others.

Based on our modeling simulations, we argue that there is a specific amount of modality-specific working memory available to speakers for grammatical encoding, and that speakers generally do not maximize working memory use. Importantly, our conclusions require researchers to take our model for granted, though we do provide metrics by which future models can be compared.

Conclusion

In this paper, we created a model of grammatical encoding (specifically function assignment and constituent assembly) by combining linguistic theory and computational cognitive modeling. We examined working memory's role during this stage of language production, along with additional data on incrementality, finding the model's fit increases with higher incrementality and lower working memory usage, but that having additional working memory available improves overall fit. Lastly, we present the first cognitive model of language production that is evaluated on a corpus, with a paradigm of inquiry that makes progress in modeling by comparing generative fits across different model versions.

Acknowledgements

This project was supported by the National Science Foundation (BCS-1457992). We thank Frank E. Ritter and Matthew Kelly for their comments on an earlier version.

References

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

Badecker, W. & Kuminiak, F. (2007). Morphology, agreement and working memory retrieval in sentence production: evidence from gender and case in Slovak. *Journal of Memory and Language*, *56*(1), 65–85.

Bock, J. K. & Levelt, W. J. M. (2002). Language production. *Psycholinguistics: Critical concepts in psychology*, *5*, 405–452.

Conway, A. R., Kane, M. J., Bunting, M. F., Hambrick, D. Z., Wilhelm, O., & Engle, R. W. (2005). Working Memory Span Tasks: A Methodological Review and User's Guide. *Psychonomic Bulletin & Review*, *12*(5), 769–786.

Daneman, M. & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, *19*(4), 450–466.

Dell, G. S., Chang, F., & Griffin, Z. M. (1999). Connectionist models of language production: lexical access and grammatical encoding. *Cognitive Science*, *23*(4), 517–542.

Ferreira, F. & Swets, B. (2002). How incremental is language production? Evidence from the production of utterances requiring the computation of arithmetic sums. *Journal of Memory and Language*, *46*(1), 57–84.

Gibson, E. (1998). Linguistic complexity: locality of syntactic dependencies. *Cognition*, *68*(1), 1–76.

Godfrey, J. J., Holliman, E. C., & McDaniel, J. (1992). Switchboard: telephone speech corpus for research and development. In *Ieee international conference on acoustics, speech, and signal processing (ICASSP-92)* (Vol. 1, pp. 517–520). IEEE.

Harrison, A. (2005). jACT-R. http://jact-r.org/.

Hartsuiker, R. J. & Barkhuysen, P. N. (2006). Language production and working memory: the case of subject-verb agreement. *Language and Cognitive Processes*, *21*(1-3), 181–204.

Joshi, A. K. & Schabes, Y. (1997). Tree-adjoining grammars. In *Handbook of formal languages* (pp. 69–123). Springer.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady* (Vol. 10, p. 707).

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop* (pp. 74–81).

Slevc, L. R. (2011). Saying what's on your mind: working memory effects on sentence production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *37*(6), 1503.

Steedman, M. (2000). Information structure and the syntax-phonology interface. *Linguistic inquiry*, *31*(4), 649–689.

Steedman, M. & Baldridge, J. (2011). Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell*.

Ferreira, V. S. (1996). Is it better to give than to donate? Syntactic flexibility in language production. *Journal of Memory and Language*, *35*, 724–755.

A database of ACT-R models of decision making

Cvetomir M. Dimov (cvetomir.dimov@unil.ch) and Julian N. Marewski (julian.marewski@unil.ch)
Department of Organizational Behavior, Université de Lausanne,
Quartier UNIL-Dorigny, Bâtiment Internef, 1015 Lausanne, Switzerland

Lael J. Schooler (lschoole@syr.edu)
509 Huntington Hall, Department of Psychology,
Syracuse University, NY 13244, USA

Keywords: model database; ACT-R; process models; decision making; inference from memory

Introduction

In a typical *two-alternative forced-choice* task of inference from memory, two objects are presented on a computer screen, which act as the *alternatives* among which a subject has to choose. Models of inference describe how *attributes* of those objects are used as *cues* to infer which of the two objects scores higher on a *criterion* of interest. Many models of inference have focused on describing not just what the outcome of this inference would be, but also which processing steps a decision maker would transverse to reach a decision. These models have increased substantially our understanding of the inferential process we follow (e.g., Bröder, 2012) and why this process is successful (Gigerenzer & Brighton, 2009).

However, some scientific question on inference from memory remain unanswered, because many models are frequently underspecified compared to the data that they are tested against. Cognitive mechanisms that remain underspecified include perception, motor action or a detailed memory theory. We argue that specifying all cognitive processes will help those models make precise predictions and address currently unaddressable questions.

The aim of this paper is to implement existing models of inference in the *cognitive architecture ACT-R* (Anderson, 2007), thus creating a database of publicly available *architectural process models of decision making*. We proceed with a brief description the classes of models that we include.

Models included in the database

Inferential models can be dichotomized, based on the type of information they rely upon, into *availability-based* and *cue-based models*. Following Newell and Bröder (2008), we further divide cue-based models into rule-based cue abstraction models, evidence accumulation cue abstraction models and configural models.

Availability-based decision models

We have included two availability-based models in our database: the *recognition heuristic* (Goldstein & Gigerenzer, 2002) and the *fluency heuristic* (Schooler & Herwtig, 2005). ACT-R implementations of availability-based models already exist (e.g., Marewski & Mehlhorn, 2011). However, we have included those for completeness. To make inferences, both of these models require only *declarative chunks* that represent the decision alternatives.

A Knowledge-based decision model

As a starting point, we include a general cue-based mechanism, which checks whether there is any knowledge present for the alternatives beyond availability and, if there is such knowledge for one alternative only, it selects that alternative (see Fechner et al., 2016).

Rule-based cue abstraction models

Cue-abstraction models operate on individual cues. These models retrieve cues one by one and make a decision when a decision rule is met. Among these models, we include fast-and-frugal heuristics (Gigerenzer, Todd, & the ABC Research Group, 1999), like *take-the-best* (Gigerenzer & Goldstein, 1996), Δ-*inference* (Luan, Schooler, & Gigerenzer, 2014) and *take-the-last* (Gigerenzer & Goldstein, 1999). We have also included more complex models, like the *weighted-linear model* (Gigerenzer & Goldstein, 1996). Some cue-abstraction models have already been implemented in ACT-R (e.g., Dimov, Marewski, & Schooler, 2013). All of these models require declarative chunks that store cue values of alternatives.

Evidence accumulation cue-abstraction models

Just like rule-based models, *evidence accumulation models* (Lee & Cummins, 2004) retrieve cues sequentially and require declarative chunks that store cue values. Unlike rule-based models, evidence accumulation models make a decision when enough evidence is accumulated in favor of one alternative or the other. We have implemented several such models, which differ in how they weigh cue values.

Configural models

Unlike cue-abstraction models, which require a separate chunk for each cue, configural models work on a set of cues. For example, *exemplar models* (e.g., Nosofsky, 1984) compare the *cue profiles* of alternatives (i.e., the set of cues associated with an alternative) to similar cue profile in memory and make inferences based on those profiles. We implement three different exemplar models. The first model evaluates each alternative based on a single similar exemplar, the second based on a weighted average of all exemplars in memory, while the third model considers fluency information. In addition, we include two *prototype models*, which differ in whether they evaluate the alternative based on a set of rules working on the entire cue profile (see Johansen & Kruschke, 2005) or based on fluency information.

In addition, we consider configural models which work with *cue-profile pairs*. These models are *instance-based learning theory* (Gonzalez, Lerch, & Lebiere, 2003) and *parallel constraint satisfaction* (Glöckner & Betsch 2008). In analogy to the exemplar implementations, we have created two instance-based learning models: the first retrieves the most similar cue-profile pair, while the second retrieves a weighted average of cue-profile pairs from memory.

Discussion and conclusion

We have provided a database of ACT-R implementations of models of inference from memory. These implementations provide comparable predictions, which can serve as a basis for model tests. Specifically, this database can be used, first, in model comparison simulations and, second, it can be utilized in future studies to identify decision processes using both behavioral and neural data. We expect that this will speed up addressing the currently present theoretical issues.

Acknowledgments

We thank the Swiss National Science Foundation for generously funding this research with Project 146702 “Architectural Process Models of Decision Making”.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Bröder, A. (2012). The quest for take the best - Insights and outlooks from experimental research. In P. Todd, G. Gigerenzer, & the ABC Research Group, *Ecological rationality: Intelligence in the world* (pp. 216-240), New York: Oxford University Press.

Dimov, C. M., Marewski, J. N., & Schooler, L. J. (2013). Constraining ACT-R models of decision strategies: An experimental paradigm. In M. Knauff, M. Pauen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th Annual Conference of the Cognitive Science Society* (pp. 2201–2206). Austin, TX: Cognitive Science Society.

Fechner, H. B., Pachur, T., Schooler, L. J., Mehlhorn, K., Battal, C., Volz, K. G., & Borst, J. P. (2016). Strategies for memory-based decision making: Modeling behavioral and neural signatures within a cognitive architecture. *Cognition, 157*, 77-99.

Gigerenzer, G., & Brighton, H. (2009). Homo heuristicus: Why biased minds make better inferences. *Topics in Cognitive Science, 1*, 107-143.

Gigerenzer, G., & Goldstein, D. G. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review*, *104*, 650–669.

Gigerenzer, G., & Goldstein, D. G. (1999). Betting on one good reason: The take the best heuristic. In G. Gigerenzer, P. M. Todd, & the ABC Research Group, *Simple heuristics that make us smart* (pp. 75-95). New York: Oxford University Press.

Gigerenzer, G., Todd, P. M., & the ABC Research Group. (1999). *Simple heuristics that make us smart.* New York: Oxford University Press.

Glöckner A., & Betsch T. (2008). Modeling option and strategy choices with connectionist networks: Towards an integrative model of automatic and deliberate decision making. *Judgment and Decision Making*, *3*, 215-228.

Goldstein, D. G., & Gigerenzer, G. (2002). Models of ecological rationality: The recognition heuristic. *Psychological Review*, *109*, 75–90.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*, 591-635.

Hertwig, R., Herzog, S. M., Schooler, L. J., & Reimer, T. (2008). Fluency heuristic: a model of how the mind exploits a by-product of information retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 34*, 1191-1206.

Johansen, M. K., & Kruschke, J. K. (2005). Category representation for classification and feature inference. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 31*, 1433-1458.

Lee, M. D., & Cummins, T. D. R. (2004). Evidence accumulation in decision making: Unifying the ‘take the best’ and the ‘rational’ models. *Psychonomic Bulletin & Review, 11*, 343–352.

Luan, S., Schooler, L. J., & Gigerenzer, G. (2014). From perception to preference and on to inference: An approach–avoidance analysis of thresholds. *Psychological Review, 121*, 501-525.

Marewski J. N. & Mehlhorn K. (2011). Using the ACT-R architecture to specify 39 quantitative process models of decision making. *Judgment and Decision Making*, *6*, 439–519.

Newell, B. R., & Bröder, A. (2008). Cognitive processes, models and metaphors in decision research. *Judgment and Decision Making, 3*, 195–204.

Nosofsky, R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10*, 104-114.

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

Modelling Simple Ship Conning Tasks

Bruno Emond
Norman G. Vinson
(firstname.lastname@nrc-cnrc.gc.ca)
National Research Council, Canada
1200 Montreal Rd, Ottawa, ON K1A 0R6

Keywords: Basic conning skills; Training simulations; Cognitive modelling; ACT-R.

Introduction

Virtual reality simulators are commonly used to train Officers of the Watch (OOW) on conning skills, which are involved in directing the steering of a ship (Reber, Bernard, & Sullivan, 2012). While valuable, these high-fidelity simulators are large, expensive, require instructor oversight, and, as a result, only allow trainees limited access. There is therefore a training gap to be filled by accessible low-cost simulations, particularly for novice and intermediate learners (Reber et al., 2012; RCN, 2015).

In the course of our work on such systems for the Royal Canadian Navy (RCN) (Emond et al., 2016), we created a gamified tutor to help novice learners apply and improve basic ship conning skills (ULearn Basic Conning). In this conning game, learners played the role of an OOW on a ship's bridge by issuing conning orders to execute a virtual captain's navigational commands.

This paper presents an analysis and a cognitive model of learners' performance on one of the training scenarios implemented in our game. We developed the cognitive model to explain the learners' performance data rather than provide information to an intelligent tutor (Wong, Kirschenbaum, & Peters, 2010). The learner response times between the three conning order conditions were not significantly different. These overall response times were modelled via the execution of several cognitive sub-tasks using the ACT-R cognitive architecture (Anderson & Lebiere, 1998; ACT-R Research Group, 2017), in particular the procedural, audio, speech, imaginal, and declarative modules. The closest cognitive model of the observed data seems to suggest that learners process the captain's command utterance to the end, even though a retrieval could be initiated before the end of the sentence, and that retrieval of the correct conning order is not always successful and that a mnemonic strategy is required.

Method

Participants 13 employees of the National Research Council Canada (NRC) volunteered for our study. No participant had prior experience with conning orders. The study was approved by the NRC Research Ethics Board.

Apparatus The apparatus consisted of a laptop computer and separate display, with a headset and microphone. Virtual Battlespace 3 simulated sea states and the ship's bridge. We enabled speech recognition of the learners' conning orders with the Microsoft Speech Platform SDK 11.

Game Scenario The game had five levels of increasing difficulty, each with trials following a particular scenario. Once a learner had successfully completed a certain number of trials in succession, the learner advanced to the next, more difficult, level. If the learner failed to complete a trial, the virtual captain provided corrective feedback. Feedback was also provided upon successful trial completion. A learning session was limited to 30 minutes. Only some learners had enough time to progress through all levels. A scenario was comprised of a series of trials of equivalent difficulty. Each trial began with a text-to-speech command from the virtual captain. The learner then responded by speaking one (scenario 1) or several conning orders (other scenarios) to have the captain's command executed by the virtual crew. This then generated the appropriate visual and auditory feedback. The first scenario required learners to issue one of three conning orders in response to the captain's command. Learners were first instructed on the three order types: *port fifteen* turned the ship left, *starboard fifteen* turned it right, and *midships* straightened the ship's head. The sequence of turning trials was randomized. However, after every order to turn the ship, the next trial required the learner to bring the ship out of the turn with a *midships* order. Consequently, learners gave twice as many *midships* orders than either turn order.

Experiment Design The dependent variable was Response Time (RT), the duration between the start of the captain's request and the response by the learner. Order type was an independent fixed effect variable with 3 levels: left/port, right/starboard, steady/midships. Learners and trials were random variables. These variables were included in a statistical mixed-effects model to identify outliers (see below).

Data Trimming The original data set consisted of 158 observations. The removal of speech recognition errors reduced the data set to 151 observations. Outliers were then identified by inspecting the response time residuals of the mixed-effects model (see *Experiment Design*) (Baayen & Milin, 2010). The procedure removed 8 additional observations when the response times were lower than 2 seconds, or greater than 8 seconds. Therefore, the data set for the analysis of response correctness had 143 observations. However, the analysis of the response time only included correct responses, which reduced the data set to 138 observations.

Results

Response Times Figure 1 shows the predicted response time for each order type. The error bars represent a 97.5% confidence interval. There was no significant effect of order type, though the predicted values increased about 250 ms from midships to port, and from port to starboard. In addition, the participants clustered into 2 groups of fast (n=10), and slow respondents (n=2). The response time per trails did not indicate any noticeable reduction of response time across the trials.

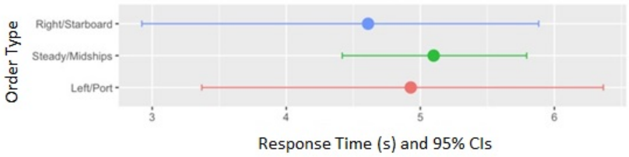


Figure 1: RTs by Order Type.

Cognitive Modelling

In spite of the lack of significant statistical differences, we explored a set of ACT-R cognitive models with the objective of reproducing the RT pattern from Figure 1. The models were partitioned according to three main processing steps required to execute a conning order: 1) the interpretation of the captain's verbal command leading to a representation of the manoeuvre to execute, 2) the retrieval of the relevant conning order, and 3) the utterance of the conning order. The first two steps had modelling variations. The modelling of the interpretation step varied in terms of when during the processing of the captain's command a representation of the task was available to initiate a retrieval. The modelling of the retrieval step varied by providing either sufficient initial declarative chunk references so that the retrieval time would be only affected by the level of chunk activation, or by allowing retrieval failure (insufficient initial declarative chunk references), which requires a prior use of a mnemonic strategy followed by the retrieval. All models used the default values for the ACT-R parameters for the procedural, audio, speech, imaginal, and declarative modules. Figure presents the modelling results.

The closer model to the observation is the Model 3, which process the captain's command utterance to the end, and used a mnemonic strategy to augment the activation of the declarative knowledge linking the captain's command to the appropriate conning order. The models 2 and 4 use a strategy for which a retrieval could be initiated before the end of the sentence (only applies to midships), while the models 1 and 3 initiated the retrieval process only at the end of the captain's utterance. The models 1 and 2 assumed an initially strong activation of the relationship between the captain's commands and the conning orders, while the models 3 and 4 used a mnemonic strategy.

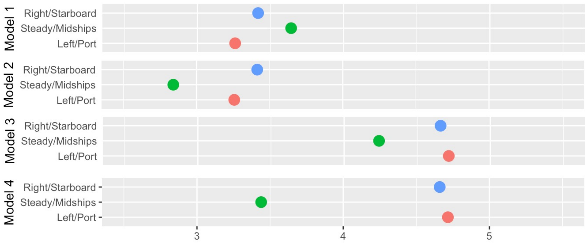


Figure 2: Predictions from 4 cognitive models.

Conclusion

Cognitive modelling offers an interesting methodology to account for reaction times data. Our modelling efforts indicate that learners process the captain's command utterance to the end, even though a retrieval could be initiated before the end of the sentence, and that retrieval of the correct conning order is not always successful and that a mnemonic strategy is required. Limitations of the current study include the non-inclusion of the visual environment for the cognitive model. Future work will include the data analysis and cognitive modelling of more complex scenarios of the ULearn Basic Conning game.

Acknowledgements

We thank Cedric Martin and Cajetan Bouchard for implementing the game. This work was performed under contract with the Royal Canadian Navy.

References

ACT-R Research Group, C. M. U., Department of Psychology. (2017). *Act-r.* Retrieved 2017-03-30, from `http://http://act-r.psy.cmu.edu`

Anderson, J. R., & Lebiere, C. J. (1998). *The atomic components of thought*. Psychology Press.

Baayen, R. H., & Milin, P. (2010). Analyzing Reaction Times. *International Journal of Psychological Research*, *3*(2), 12 – 28.

Emond, B., Maugeais, M., Vinson, N. G., Saikh, K., Fournier, H., Lapointe, J.-F., & Martin, C. (2016). Adaptive training simulation using speech interaction for training navy officers. In *Interservice/industry training, simulation, and education conference (i/itsec)* (pp. 2924–2934). Orlando, FL: National Training and Simulation Association.

RCN. (2015). *Royal Canadian Navy Future Naval Training Strategy.* Royal Canadian Navy.

Reber, E. A., Bernard, B. J., & Sullivan, J. (2012). *The Sea of Simulation: Improving Naval Shiphandling Training and Readiness Through Game-Based Learning* (Tech. Rep.). Monterey, CA: U.S. Naval Postgraduate School.

Wong, J. H., Kirschenbaum, S. S., & Peters, S. (2010). Developing a Cognitive Model of Expert Performance for Ship Navigation Maneuvers in an Intelligent Tutoring System. In *Proceedings of the 19th conference on behavior representation in modeling and simulation* (pp. 21–24).

Model Predictions of Reward Optimization in Discrete Dual-Task Scenarios

Christian P. Janssen (c.p.janssen@uu.nl)
Emma Everaert, Heleen M.A. Hendriksen, Ghislaine L. Mensing, Laura J. Tigchelaar,
Veere Weermeijer, Hendrik Nunner
Utrecht University, Experimental Psychology and Helmholtz Institute
Heidelberglaan 1, 3584 CS Utrecht, The Netherlands

Keywords: dual-task; task interleaving; cognitive modeling.

Efficiency & Rewards in Multitasking

Human multitasking occurs in many settings. For example, office workers switch tasks roughly every two to three minutes, with about half of the switches due to self-interruption (González & Mark, 2004). An open question is how efficient it is to interleave between tasks so frequently.

To assess efficiency, one first needs to assess what the most efficient ways of interleaving are. This can be done by predicting the (financial) outcome of multiple interleaving strategies using a model (cf. Janssen et al., 2011). In a subsequent step, one can measure in an experiment whether people applied the predicted most efficient strategies.

Previous work has applied this approach in various multitasking settings (e.g., Duggan et al., 2013; Farmer et al., accepted; Howes et al., 2009; Janssen et al., 2011; Janssen & Brumby, 2015; Payne & Howes, 2013; Payne et al., 2007; Zhang & Hornof, 2014). In general, people change their strategies based on the task rewards. However, they do not always apply the global maximum strategies, but also satisfice or end up at local optima (Janssen & Brumby, 2015).

A limitation of previous work is that efficiency has mostly been studied in specific task settings, without exploring the effect of different reward or payoff functions. However, in everyday life, many different payoff functions are possible. We address this limitation by studying efficiency in a dual-task setting that allows the use of various payoff functions.

Method: Whack-a-Mole Task

In our experiment, participants perform two instances of the same task: to "whack moles". A picture of a mole appears and remains at a random location in a 3x3 grid until a numeric key is hit (e.g., '9' for top-right, '1' for bottom-left). Participants can hit moles in two available screens, of which only 1 is visible (and active) at a time (see Figure 1). They can switch between screens by pressing the space bar. The objective is to hit 50 moles total across both screens.

Between blocks of trials, we manipulated how rewarding it is to hit moles in each screen. Three payoff schemes were used. For each scheme, a total of 100 points was earned if all 50 moles were hit. However, the distribution of scores per mole differed. In the linear scheme, each mole gave 2 points. In the diminishing return scheme, the first moles gave most points and this decreased over time (scores 16.00, 13.45, 11.29, … 0.003 points). In the exponential scheme high points were only given towards the end (0.03, …, 10.05, 11.33). Participants received full feedback on how much points they earned with each mole.

Within this paradigm, one can investigate how many moles are hit per screen. The total of 50 moles can be reached in different divisions between the two screens (e.g., 50-0, 49-1, 48-2, 25-25, 0-50). We refer to these as different strategies. We studied three different dual-task scenarios that differ in the distribution of possible scores across strategies.

We label the combination of two linear tasks as an 'easy' scenario, as each strategy results in the same score (50 x 2 = 100 points). We predicted that the applied strategies vary within and between participants, as there is no environmental pressure to converge on one single strategy.

We label the combination of diminishing and exponential payoff as "hard", as the optimal strategy requires some investment in both tasks: get high points early on the diminishing task, and then switch to the exponential task. We predict variation of strategies between and within subjects, as the optimal strategy might be hard to find.

Finally, we label the combination of diminishing returns and linear as "constrained". Here, the optimal strategy is to gather points on the diminishing returns task until scores diminish to less than 2 points per mole (the score on the linear task). Given this constrained prediction of optimality and its clear pattern, we expect that participants will find the optimal strategy relatively quickly and then apply it consistently.

Experiment 1 (48 participants) had 15 trials per condition, and no time limit per trial. In experiment 2 (23 participants) we introduced a time limit of 30 seconds, to test performance under time pressure. In experiment 3 (21 participants) we decreased the time limit to 25 seconds. Only the constrained and hard condition were used. We also increased the number of trials per block to 25 to rule out that some sub-optimality could be due to insufficient experience.

Model

The model for experiment 1 analytically explores how scores change depending on how many moles are hit on each task. For the experiments with a time limit, we also had to

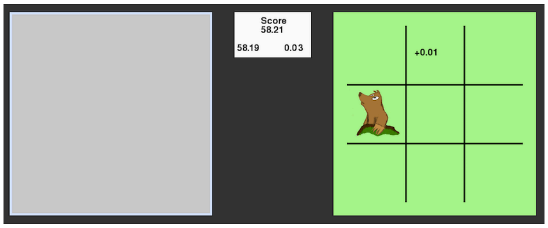


Figure 1: Layout of the whack-a-mole task.

take typing speed into account. In this model we assumed that participants switched at most once between screens, and set the switch cost to the average measured switch cost. Then it was investigated how strategy (i.e., hitting 1, 2, 3, etc. moles on task A and the remaining moles on task B) affected scores.

For each participant we measured the mean and standard deviation of their inter-keypress interval (i.e., time between hitting two moles in dual-task). These formed the parameters of a normal distribution from which, for each mole, a sample was drawn to determine typing time. This allowed modeling of individual differences in typing speed. This was crucial as the predictions for optimal strategies varied based on typing speed. For each participant, performance of each strategy was simulated 100 times. We report mean performance per strategy and participant.

Results

In all three experiments the patterns fitted our predictions (see Method). For example, the optimal strategy was consistently found earlier and applied more consistently in the constrained condition compared to the hard condition. When a time limit was introduced, more individual differences emerged in what the optimal strategy was and whether people applied those. As an illustration, Figure 2 shows data from experiment 3 for the constrained (left plot) and hard condition (right). In each plot, the vertical axis shows how many moles were hit on the diminishing task, with the remainder being hits on the linear (left plot) or exponential task (right). On the horizontal axis, all unique participants are plotted (sorted based on total achieved score, the best participants are to the right). The heatmap color of each bar shows the predicted score, with warmer colors for better scores. The black crosses show the applied strategies per participant during the last five trials.

The figure shows that the distribution of predicted scores differs between participants, due to individual differences in typing (mole hitting) speed. Moreover, in the constrained condition, participants are more consistent (overlapping crosses in a narrow band) in applying the best strategies. In the hard condition, there is variation in where the optimum strategy is: the left most participants should focus solely on the diminishing task (warmest colors at the top), as they type too slowly to earn points on the exponential task (note that the model predicts that not all 50 moles are hit). In contrast, the other participants need to balance both tasks.

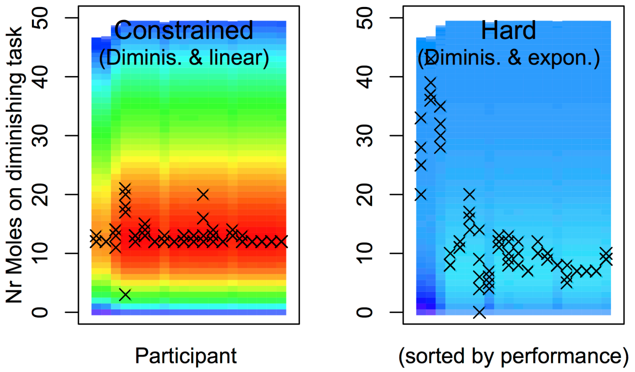


Figure 2: Predictions & performance in experiment 3

Taken together, the results demonstrate that rewards influence a priori how easy it is for people to divide their time efficiently between two discrete tasks. They also affect whether people can achieve efficient performance. An implication for theory is that some scenarios (constrained) are better than others (easy, hard) to test people's general multitasking efficiency. The work also demonstrates that hard scenarios might require substantial training, or different feedback for people to perform efficiently.

Acknowledgments

CPJ is supported by EU H2020-MSCA-IF-2015 grant 705010. EE, HH, GM, LT contributed equally to study 1; VW conducted study 2; HN conducted study 3. CPJ coordinated task design, modeling, writing, and supervised all studies.

References

Duggan, G. B., Johnson, H., & Sørli, P. (2013). Interleaving tasks to improve performance: Users maximise the marginal rate of return. *International Journal of Human-Computer Studies*, *71*, 533–550.

Farmer, G. D., Janssen, C. P., Nguyen, A. & Brumby, D. P. (accepted). Dividing attention between tasks: Testing whether explicit payoff functions elicit optimal dual-task performance. *Cognitive Science*.

González, V. M., & Mark, G. J. (2004). 'Constant, constant, multi-tasking craziness': managing multiple working spheres. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 113–120). New York, NY: ACM Press

Howes, A., Lewis, R. L., & Vera, A. (2009). Rational adaptation under task and processing constraints: Implications for testing theories of cognition and action. *Psychological Review*, *116,* 717–751.

Janssen, C. P., & Brumby, D. P. (2015). Strategic Adaptation to Task Characteristics, Incentives, and Individual Differences in Dual-Tasking. *PLoS ONE*, *10*(7), e0130009.

Janssen, C. P., Brumby, D. P., Dowell, J., Chater, N., & Howes, A. (2011). Identifying Optimum Performance Trade-Offs using a Cognitively Bounded Rational Analysis Model of Discretionary Task Interleaving. *Topics in Cognitive Science*, *3*, 123–139.

Payne, S. J., & Howes, A. (2013). *Adaptive Interaction: A utility maximisation approach to understanding human interaction with technology*. Morgan & Claypool.

Payne, S. J., Duggan, G. B., & Neth, H. (2007). Discretionary task interleaving: heuristics for time allocation in cognitive foraging. *Journal of Experimental Psychology: General*, *136*, 370–388.

Zhang, Y., & Hornof, A. (2014). Understanding Multitasking Through Parallelized Strategy Exploration and Individualized Cognitive Modeling. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 3885-3894). New York, NY: ACM Press

Modelling the role of grammatical functions in language processing

Stephen Jones
Faculty of Linguistics, Phonology & Phonetics, University of Oxford,
Clarendon Institute, Walton Street, Oxford OX1 2HG, UK

Keywords: ACT-R; Lexical Functional Grammar; language processing.

This poster presents a novel computational model of language processing that uses functional relationships as defined in Lexical Functional Grammar (LFG, Dalrymple, 2001), as well as phrase-structure, to generate syntactic structure online. The model will be used to derive time-courses of processing from competing syntactic accounts of unbounded multiple-gap dependencies and compare these with empirical data.

The empirical challenge It has been demonstrated for a number of years that reading speed slows at points in a sentence where an open unbounded dependency might be attached, sometimes called gap sites. For example, Stowe (1986) compared the reading time-course of sentences (1)-(3).

(1) My brother wanted to know if Ruth will bring [A] us home to [B] Mom at Christmas.

(2) My brother wanted to know who$_i$ Ruth will bring [A] $\_\_\_{}_i$ home to [B] Mom at Christmas.

(3) My brother wanted to know who$_i$ Ruth will bring [A] us home to [B] $\_\_\_{}_i$ at Christmas.

Both sentences contain an embedded clause, but in (2) and (3) an unbounded dependency is created at *who*. In (2) longer reading times are seen at [A] compared to (1), but not at [B] where the unbounded dependency has been closed. However, in (3) longer reading times are seen at both [A] and [B]. The inference is that the parser, once aware of an open unbounded dependency, predicts an attachment at each possible gap site. The longer reading time reflects the processing needed to revise this prediction if a word is encountered at the site, such that it is not possible to attach the unbounded dependency. These phenomena are sensitive to island constraints, with no reading slowdown seen at illicit attachment sites. Thus Phillips (2013) argues that at least some elements of syntactic structure are available to the parser early in processing (cf. models where syntactic well-formedness is checked at a later stage, or optionally).

The picture with parasitic gaps is more complicated. For example, the attachment site [A] in (4) is illicit, compared to the licit attachment site [B] in (5). However, as seen in (6), attachment site [A] is allowed if it is co-referent with a subsequent licit attachment at site [B].

(4) * What$_i$ did the attempt to repair [A]$\_\_\_{}_i$ ultimately damage the car?

(5) What$_i$ did the attempt to repair the car ultimately damage [B]$\_\_\_{}_i$?

(6) What$_i$ did the attempt to repair [A] $\_\_\_{}_i$ ultimately damage [B] $\_\_\_{}_i$?

Phillips (2006), investigating the processing time-course of (6), found a processing slowdown at [A], although the site is only licit when a further attachment site is available. In structures where a parasitic gap was never possible, island constraints held and no slowdown was seen at illicit attachment sites. Phillips argues that the parser's access to syntactic constraints is more sophisticated simply recognising islands and suspending the prediction of attachment within them.

The grammar-theoretical challenge Many transformational accounts of unbounded dependencies and island constraints make use of constituent structure relationships (e.g. constraints related to c-command). Conversely accounts within LFG such as Bresnan et al. (2016) and Dalrymple (2001) have focused on the role of functional structure. This has included theories of functional uncertainty, path constraints, and explicit empty categories. For multiple-gap dependencies, Falk (2011) proposes additional constraints on the feature WHPATH, whereas Alsina (2008) uses functional prominence constraints on structure-sharing.

One way to assess the competing claims of these accounts is to compare the impact of including them as grammatical assumptions in computational models of processing. Each account presents different requirements for the content of memory storage chunks, available cues for memory retrieval, and maintenance vs. reactivation of information in working memory. The model presented is an attempt to do this.

Existing models of processing The model being presented is built in ACT-R (Anderson, 2007), a symbolic, rather than connectionist, cognitive architecture. Early work in ACT-R to model the time-course of language processing was carried out by Lewis and Vasishth (2005), henceforth L&V, who assume that only the word being attended to is maintained in working memory, and that attachment into structure is necessary before the next word is attended. Attachment follows via cue-based retrieval of previously-processed memory chunks, with abstract structural chunks being created if necessary. The developing syntactic structure is not available on-line, although it can be computed from the contents of working memory, producing a binary-branching constituent-structure tree . However, the time required to calculate the impact of long-distance c-structure constraints is likely to be much greater than values reported from experimental data, which raises problems for transformational accounts of island constraints. Engelmann (2016) has expanded L&V's work to model eye-movements during reading. The capacity of his model is enhanced such that the current IP node can be accessed without a time penalty, and provides more retrieval buffers for grammatical information. This enables attachments to be generated between multiple chunks simultaneously, which is used in the new model.

The new model This model's core processing cycle adapts that of Engelmann's model. Syntactic structure comprises memory chunks representing f-structures, rather than c-structure nodes. As in L&V's and Engelmann's models, the full syntactic structure can only be calculated by retrieving memory chunks, but in addition to the word being attended, a 'live' chunk may be maintained in working memory, reflecting the observed contextual prediction of syntactic structure (e.g. Wicha et al., 2004). There is enhanced access to the memory chunk representing the current verbal predicate, analogous to Engelmann's assumption of enhanced access to an IP node and is supported by Chow et al. (2015), who find an advantage for local arguments over matrix arguments, during the processing of an embedded clause. Crucially, it is not necessary for a syntactic chunk to be attached before the next word is attended, provided that its grammatical function (GF) is clear. The model also assumes that some information about the target GF for an open dependency is maintained, following Wagers (2013). The number of memory chunks needed to represent a sentence, and therefore the number of attachments, is smaller than earlier models because the information provided by functional categories C, D and I sits in a memory chunk with a semantic PRED value, rather than in a distinct chunk. Thus parsing the sentence *The writer surprised the editors* generates 6 grammatical memory chunks in L&V's model: one for each word and one abstract structural chunk. The presented model generates 3 for the same sentence, representing *editor*, *surprise*, and *writer* respectively.

Regarding attachment, this is governed in L&V's and Engelmann's models by the availability of attachment sites found using retrieval cues appropriate to structural expectation and lexical information, and by specific attachment productions that represent phrase-structure rules applying a left-corner parsing algorithm. Memory chunks are generated for terminal nodes and for any intermediate nodes necessary to attach the current word to existing structure. In the new model, productions are selected taking into account the current processing state; lexical information, including but not limited to syntactic category; and the level of activation of memory chunks outside working memory. The successful production determines whether a new chunk is attached to the current 'live' chunk or whether a chunk must be retrieved; which attachment sites are available within a given memory chunk as attribute-value pairs are added; and also which chunk is 'live' once attachment has been made. This might be the current 'live' chunk, the newly-attached chunk, or a previously-processed chunk (e.g. returning to a matrix clause after processing an argument of an embedded predicate).

Open questions for discussion Open questions presented include assumptions about the details of attachment productions, which must be congruent with phrase structure rules as well as functional relationships; options for the detailed representation of the presence of an open unbounded dependency; and methods of calculating or representing the constraints that determine potential attachment sites.

References

Alsina, A. (2008). A theory of structure sharing: Focusing on long-distance dependencies and parasitic gaps. In Butt, M. and King, T. H., editors, *Proceedings of the LFG08 Conference*, pages 5–24, Stanford, CA. CSLI Publications.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press, New York.

Bresnan, J., Asudeh, A., Toivonen, I., and Wechsler, S. (2016). *Lexical-functional syntax*. Wiley Blackwell, Chichester, West Sussex, 2nd edition.

Chow, W.-Y., Smith, C., Lau, E., and Phillips, C. (2015). A "bag-of-arguments" mechanism for initial verb predictions. *Language, Cognition and Neuroscience*, page DOI: 10.1080/23273798.2015.1066832.

Dalrymple, M. (2001). *Lexical Functional Grammar*. Academic Press, San Diego.

Engelmann, F. (2016). An integrated model of eye movement behaviour in sentence comprehension. https://github.com/felixengelmann/act-r-sentence-parser-em.

Falk, Y. N. (2011). Multiple-gap constructions. In Butt, M. and King, T. H., editors, *Proceedings of the LFG11 Conference*, pages 194–214, Stanford, CA. CSLI Publications.

Lewis, R. L. and Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29:375–419.

Phillips, C. (2006). The real-time status of island phenomena. *Language*, 82(4):795–823.

Phillips, C. (2013). On the nature of island constraints I: Language processing and reductionist accounts. In Sprouse, J. and Hornstein, N., editors, *Experimental syntax and island effects*, chapter 4, pages 64–108. Cambridge University Press.

Stowe, L. A. (1986). Parsing WH-constructions: Evidence for on-line gap location. *Language and Cognitive Processes*, 1(3):227–245.

Wagers, M. W. (2013). Memory mechanisms for *wh*-dependency formation and their implications for islandhood. In Sprouse, J. and Hornstein, N., editors, *Experimental syntax and island effects*, chapter 7, pages 161–185. Cambridge University Press.

Wicha, N. Y. Y., Moreno, E. M., and Kutas, M. (2004). Anticipating words and their gender: An Event-Related Brain Potential study of semantic integration, gender expectancy, and gender agreement in Spanish sentence reading. *Journal of Cognitive Neuroscience*, 16(7):1272–1288.

Conceptual Approach to Model the Effects of Feedback on Mental Model Activation

Oliver W. Klaproth (oliver.klaproth@airbus.com)
Department of Cognitive Modeling in Dynamic Human-Machine Systems
Technische Universität Berlin

Nele Russwinkel (nele.russwinkel@tu-berlin.de)
Department of Cognitive Modeling in Dynamic Human-Machine Systems
Technische Universität Berlin

Abstract

A concept of a cognitive model is presented that allows for examining effects of negative feedback on changing mental models of a user interface. We hypothesize that users who are given more evidence against their mental model are less likely to change it, depending on the type of feedback they are provided with. Assumptions about the effects of feedback are explained from a theoretical account of mental model activation based on mental set and fixation. We propose an experimental keypad task that requires users to switch from a mental model of a calculator keypad to a telephone mental model based on two types of feedback. A scheme for an ACT-R cognitive model is provided, showing how predictions on user behavior can be made using spreading activation and utility learning mechanisms.

Keywords: Mental models; keypad; mental fixation; redistribution theory; mental set; ACT-R; partial matching; utility learning.

A Method to Systematically Find Better Cognitive Models: Conditional Reasoning as an Example

Daniel Lux (luxd@tf.uni-freiburg.de)
Department of Computer Science, Georges-Köhler-Allee 52, 79110 Freiburg, Germany

Marco Ragni (ragni@cs.uni-freiburg.de)
Department of Computer Science, Georges-Köhler-Allee 52, 79110 Freiburg, Germany

Keywords: Wason Selection Task; Optimization; Inference-guessing model; Evaluating Computational Cognitive Models

Introduction

Cognitive models are often built on general cognitive principles, principles derived from empirical data, and sometimes a modelers' intuition and introspection. If the model is able to explain the data better than other models it is considered as a good model. So far most if not all evaluation depends on the models that have been proposed by researchers. Close to never any cognitive model is being systematically optimized after it has been created. In the following we propose a method to optimize a cognitive model by systematically changing the originally proposed one by maintaining its cognitive structure. As an example let us consider one of the most successful models to explain human reasoning with conditionals, the inference-guessing model proposed by Klauer et al. (2007). The model is a multinomial processing tree (MPT) and is used to explain human reasoning in the Wason Selection Task (Wason, 1968), see Figure 1.

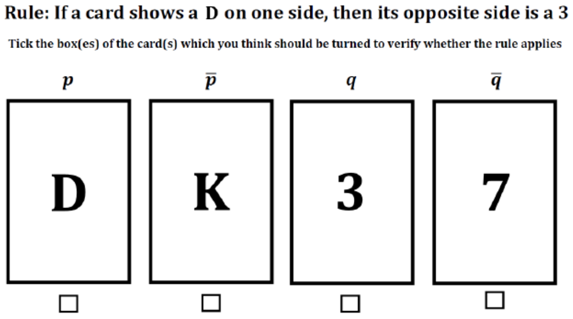

Rule: If a card shows a D on one side, then its opposite side is a 3
Tick the box(es) of the card(s) which you think should be turned to verify whether the rule applies
p
p̄
q
q̄
D
K
3
7
□
□
□
□

Figure 1: A pictorial representation of the Wason Selection Task. Participants have to decide which card(s) they necessarily need to turn to test whether "If a card shows a *D* on one side, then its opposite side is a 3" holds. The small letters p, $\overline{p}$, q, and $\overline{q}$ represent the cards corresponding to a general conditional of the form *if p then q*.

Answers from the participants about the number of cards to be turned range from 0 to 4. A meta-study by Oaksford & Chater (1994) analyzed how many of the cards are turned, and every combination of cards actually occurs although only two of the cards are logically sound (cards "D" and "7).

Multinomial Processing Tree Models

Multinomial Processing Trees (MPTs) are acyclic directed graphs. Each leaf node represents one possible observed outcome denoted by O_i for $1 \leq i \leq K$ for K observed outcomes. Non-leaf nodes represent one single latent part of the overall process, splitting the still possible answers into two distinct groups, assuming the MPT is binary and all arbitrary MPTs can be transformed to a binary MPT (Klauer et al., 2015). These are governed by a transition probability $\theta_i \in [0,1]$ for one group and $(1-\theta_i)$ for the other. The probability of one leafe node is the product of all probabilities along the path to the node. The probability of one outcome O_i is the sum of the probabilities of all leafe nodes with outcome O_i.

The inference-guessing model builds on the core cognitive assumption that the reasoners' understanding and interpretation of the rule leads the inferences about how the rule is actually applied to the cards. The inference part uses five parameters, four of them describe how the rule is interpreted and one referring to the reasoning itself. The guessing part is modeled by 4 parameters which decide for each card independently if it is turned. The two parts are divided by one parameter which separates guessing and reasoning.

A Method to Find Better Models

Our proposed method consists of three subsequent steps. The first step needs to check whether a cognitive model structure can be changed at all, i.e., if it consists of parameters or assumptions. The second step is to identify where the model represents the latent cognitive processes and which core assumptions cannot be changed. The third step needs to formulate a model changing strategy according to the findings from the first step without changing the cognitive core concepts and assumptions from the second step.

Applied to MPTs this works as follows: The first step is quite obvious as it is how acyclic directed graphs can be modified. This includes adding new nodes, deleting nodes, changing the graphs' structure in any way imaginable or changing parameter-node connections. For the second step, we have to look at the models and where the cognitive assumptions are interwoven with the model implementation. For MPT this resembles the parameters. In MPTs one single cognitive process is described by a single parameter and splits the currently possible answers in two distinct defined subsets. This is what cannot be changed by the process because this would change the cognitive assumptions and would lead to a loss of explanation. In formal terms, this means that every parameter has to coincide with one of the cognitive model proposed processes.

The two changes that were done are a substitution of a parameter weight and deleting of redundant subtrees. Substitution of a parameter weight means that one cognitive process which has the same probability weight in the tree at every place it occurs is allowed to have different weights under different circumstances. Formally this means that if there is a parameter A with value p in the original model which occurs in a set of k nodes $N_A = \{n_1, \dots, n_k\}$, which means $\theta_i = p$ in all these nodes, can be replaced by a set of parameters $A_j, j = \{1, \dots, l\}, l \leq k$ where all A_j split the possible answers into the same sets thus they all model the same predicate and $\bigcup_j N_{A_j} = N_A$. This allows the weights (θ) of the nodes governing the same process defined by A to be different at different circumstances.

A delete operation preserves the cognitive structure if the deleted partial tree is redundant. This means that every outcome O_i which has a leaf node in the deleted subtree has at least one leaf node outside of the deleted subtree and therefore is still part of the prediction after the deletion. This is the same as assuming that the parameter at this point is either 1 or 0.

To evaluate the procedure we implemented each model as an MPT and evaluated them with `MPTinR` (Singmann & Kellen, 2013) a package for the software environment R for statistical computation (R Core Team, 2016). The G^2, AIC, BIC, and FIA values were then compared. These values measure how well a model fits given data. AIC and BIC also include the number of parameters into their measure. FIA includes complexity measures of the model (Ragni et al., 2014).

The dataset used is from the studies from Klauer et al. (2007) and Stahl et al. (2008), the first eight are from Klauer et al. (2007) (since the Experiments 5 and 6 consider specific manipulations like directionality or nonstandard arrays) and the other twelve are from Stahl et al. (2008). We chose it because the Klauer et al. (2007) data is the data the inference-guessing model was originally fitted and it was also used on the data from Stahl et al. (2008). Together we considered all 6321 participants from the studies.

Results

For the inference-guessing model (Klauer et al., 2007) about 14,000 unique models were generated by substitution from this procedure. About 6,000 of them fulfill the constraints that in each partial tree the number of different parameters has to be smaller than the number of possible answers. Collapsing nodes enabled the generation of 126 new models. 3,240 of these trees performed better than the inference-guessing model on the aggregated data. When we evaluate the aggregated data the values of the best optimizing model is outperforming the original model by a factor of 20 for the G^2. Collapsing nodes didn't yield equally good results, it performed slightly worse than the original model on the aggregated data.

Table 1: Results on the aggregated data.

MPT-model	G^2	AIC	BIC	FIA
optimized model	9.39	37.39	132.23	49.40
Klauer et al. (2007)	215.50	235.50	303.25	144.89

General Discussion

While there is a vast literature on modeling with multinomial process trees, many models are often introspective models. What we propose is to use this as the starting point and go further by letting optimization algorithms improve on these models while preserving the modeler's intuition and cognitive assumption. Although the inference-guessing model provides quite a good fit already our method achieved an improvement by a factor of 20 for the G^2. But what is shown here is only the first scratch on the surface. There are still many ways to explore this further. Combining and implementing more strategies, finding better quality functions and expanding this procedure on more different types of models are next steps.

Acknowledgement

The paper was partially supported by DFG-project RA 1934-2/1 and Heisenbergproject RA 1934-3/1.

References

Klauer, K. C., Singmann, H., & Kellen, D. (2015). Parametric order constraints in multinomial processing tree models: An extension of Knapp and Batchelder (2004). *Journal of Mathematical Psychology*, *64*, 1–7.

Klauer, K. C., Stahl, C., & Erdfelder, E. (2007). The abstract selection task: new data and an almost comprehensive model. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *33*(4), 680.

Oaksford, M., & Chater, N. (1994). A rational analysis of the selection task as optimal data selection. *Psychological Review*, *101*(4), 608.

R Core Team. (2016). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org`

Ragni, M., Singmann, H., & Steinlein, E.-M. (2014). Theory Comparison for Generalized Quantifiers. In P. Bello, M. Guarini, M. McShane, & B. Scassellati (Eds.), *Proceedings of the 36th cognitive science conference* (pp. 1228 – 1234). Austin, TX.

Singmann, H., & Kellen, D. (2013). MPTinR: Analysis of multinomial processing tree models in R. *Behavior Research Methods*, *45*(2), 560–575.

Stahl, C., Klauer, K. C., & Erdfelder, E. (2008). Matching bias in the selection task is not eliminated by explicit negations. *Thinking & Reasoning*, *14*(3), 281–303.

Wason, P. C. (1968). Reasoning about a rule. *The Quarterly journal of experimental psychology*, *20*(3), 273–281.

Cognitive Modeling of Cardiopulmonary Resuscitation Knowledge and Skill Spanning Months to Years

Sarah Maass (s.c.maass@rug.nl)
Research School of Behavioral and Cognitive Neuroscience,
University of Groningen, The Netherlands

Florian Sense (f.sense@rug.nl)
Department of Experimental Psychology and Department of Psychometrics and Statistics,
University of Groningen, The Netherlands

Matthew Walsh (mmw188@gmail.com)
TiER 1 Performance Solutions
Pittsburgh, Pennsylvania, USA

Kevin Gluck (kevin.gluck@us.af.mil)
Cognitive Models and Agents Branch, Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio, USA

Hedderik van Rijn (d.h.van.rijn@rug.nl)
Department of Experimental Psychology and Department of Psychometrics and Statistics,
University of Groningen, The Netherlands

Keywords: procedural knowledge; declarative knowledge; predictions; adaptive learning; CPR

The acquisition and retention of factual and procedural knowledge is impacted by multiple factors. Most basically, as material is studied, memory performance improves. The change in performance approximates a power law – additional practice produces further improvement, but at a diminishing rate (Newell & Rosenbloom, 1981). Conversely, memory performance degrades during periods of non-use. This also approximates a power law, with rapid loss occurring initially followed by more-gradual, sustained loss (Anderson & Schooler, 1991; Rubin & Wenzel, 1996). In addition to these factors, knowledge acquisition and retention is impacted by the temporal distribution of practice. Separating practice repetitions by a delay slows learning but enhances retention (Cepeda et al., 2006).

The Predictive Performance Equation (PPE; Jastrzembski & Gluck, 2009; Walsh, Gluck, Gunzelmann, Jastrzembski, & Krusmark, in revision) is an extension of the General Performance Equation (GPE; Anderson & Schunn, 2000). Both PPE and GPE represent how amount of practice and elapsed time since test impact learning and retention. However, PPE also accounts for how the temporal distribution of practice affects retention. In PPE, including more space between practice repetitions reduces decay rate, thereby increasing retention. In the research reported here, we test the ability of PPE to generate subject-level performance predictions for cardiopulmonary resuscitation (CPR). Such predictions, if valid, could be used to prescribe personalized refresher training intervals.

In addition to testing a model of knowledge acquisition and retention, we sought to study learning and memory in conditions more reflective of the real world. Although there is an extensive scientific literature associated with learning, forgetting, and spacing effects, much of it suffers from two deficits in ecological validity. First, most published studies of these effects involve abstract, simplified laboratory tasks. Second, in most of these studies, all of the learning and retention taking place in short time periods, often within a single experiment session.

In this effort, we tackle both of these traditional deficiencies. The task domain we studied is cardiopulmonary resuscitation (CPR), which is a real-world basic lifesaving skill relevant worldwide. Using a modern, instrumented QCPR manikin, it is a task that taps into a combination of procedural and script-based learning. The learning and retention periods we tested span months to years, allowing us to assess the predictive accuracy of PPE over an ecologically relevant timeframe.

Participants were German citizens enrolled as students at the University of Groningen. A requirement for initial issuance of a driver's license in Germany is the successful completion of first aid training that includes CPR. Thus, each participant in this study completed initial CPR training when they received their driver's license (an average of 43.1 months before data was collected; SD = 21.9). The initial study session started with one bout of CPR to assess current performance levels, after which participants were retrained and completed three further CPR training bouts (with a 10 min delay between the initial test and the next two bouts, and then an hour between the 3rd and 4th sessions). In addition to CPR data, we also collected Raven-based IQ scores, fact learning proficiency using the SlimStampen algorithm (Sense, Behrens, Meijer, & van Rijn, 2016), and data from a Serial Reaction Time Task (Robertson, 2007).

Here, we focus on the CPR results. We present the data of a first session in which learning and retention of declarative and procedural knowledge is assessed. Based on this data, we used PPE to predict retention performance after an 18-week delay. At the writing of this abstract, the second session has not been collected, but the predictions have been made and will be posted at the Open Science Foundation. (https://osf.io/9er6g/).

During the experiment session, CPR performance scores ranging from 0 to 100 are automatically generated and recorded by the manikin for each training bout. We calibrated PPE separately to each of 50 participants based on their training history and performance scores. We do not know how participants scored during the initial training they completed before receiving their driver's license but we know the approximate delay since the training. To include this observation in a participant's training history, we make the simplified assumption that everyone scored 75 during their training, which is the threshold for adequate performance. We then used the calibrated model to predict retention performance after an 18-week delay.

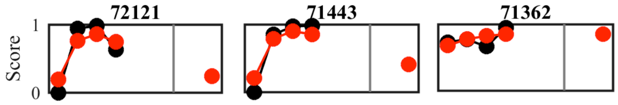


Figure 1: Data of three participants and model performance predictions.

Data in Figure 1 show results from three participants, strictly as examples of the individual differences we see in performance. The total sample, available at OSF, is 50 participants. The black points show observed performance scores from the first study session. The red points show the model's fits to performance from the first session, and the model's predictions for 18-week retention. At the conference, we will present the full data of the initial and second session, and discuss in detail how the model predictions match up with the behavioral data.

Acknowledgments

This work was supported by EOARD grant 11926121. EOARD was not involved in planning or reporting this study.

References

Anderson, J. R., & Schunn, C. (2000). Implications of the ACT-R learning theory: No magic bullets. *Advances in instructional psychology, Educational design and cognitive science*, 1-33.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological science*, *2*(6), 396-408.

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), 354–380. http://doi.org/10.1037/0033-2909.132.3.354

Jastrzembski, T. S., & Gluck, K. A. (2009). A formal comparison of model variants for performance prediction. *Proceedings of the International Conference on Cognitive Modeling*, Manchester, UK.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition*, *1*, 1-55.

Robertson, E. M. (2007). The serial reaction time task: implicit motor skill learning? *Journal of Neuroscience*, *27*(38), 10073-10075.

Rubin, D. C., & Wenzel, A. E. (1996). One hundred years of forgetting: A quantitative description of retention. *Psychological Review, 103*, 734–760.

Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. (2016). An individual's rate of forgetting is stable over time but differs across materials. *Topics in Cognitive Science*. http://doi.org/10.1111/tops.12183

Walsh, M. W., Gluck, K. A., Gunzelmann, G., Jastrzembski, T., & Krusmark, M. (in revision). Evaluating the theoretical adequacy and applied potential of computational models of the spacing effect.

Bayesian network model for human performance assessment using virtual environments

Allison Moyle (p13dabm@mun.ca)
Faculty of Engineering and Applied Science, Memorial University
St. John's, NL, Canada, A1B 3X5

Mashrura Musharraf
Faculty of Engineering and Applied Science, Memorial University
St. John's, NL, Canada, A1B 3X5

Jennifer Smith
Faculty of Engineering and Applied Science, Memorial University
St. John's, NL, Canada, A1B 3X5

Brian Veitch
Faculty of Engineering and Applied Science, Memorial University
St. John's, NL, Canada, A1B 3X5

Faisal Khan
Faculty of Engineering and Applied Science, Memorial University
St. John's, NL, Canada, A1B 3X5

Keywords: Bayesian Network; Human Reliability Assessment; Safety; Training.

Introduction

Human reliability assessment (HRA) techniques are used to predict people's response to emergencies. Human performance data required to perform HRA in emergency scenarios are not readily available. Due to lack of available data on human performance in emergencies, HRA analysis is usually done using expert opinion (Groth, Smith, & Swiler, 2014). Expert opinion, while valid in certain circumstances, can lead to uncertainty in results (Musharraf et al., 2013). This uncertainty can be reduced by use of a virtual environment (VE) to gather data on human's performance during emergency situations. Another limitation of most HRA methods is that they do not consider dependency amongst performance shaping factors and associated actions (Musharraf et al., 2013). Bayesian networks (BN) enable modeling of such dependency (Groth & Mosleh, 2012b). By using a Bayesian network (BN) in combination with a virtual environment, the uncertainty in results can be reduced, and a person's response in emergency situations can be assessed.

Musharraf et al. (2014, 2013) presented a VE based technique to determine an individual's failure probability in emergency situations. They used data from a VE to quantify a BN to assess human reliability in offshore emergency conditions. Performance shaping factors (PSFs) were selected based on a task analysis and the capability of the VE. The factors were varied at different levels and data to quantify the BN were collected. Results from the study indicated that this method is a viable way to overcome uncertainty associated with expert opinion, to create a realistic way to demonstrate dependency amongst PSFs, and to estimate human error more accurately. However, choice of the PSFs in the study was constrained by the VE's capability and did not provide an ideal representation of offshore emergency scenarios.

This paper addresses these limitations by using the same experimental technique, but with more realistic PSFs. The PSFs selected for this research are stress, uncertainty, and complexity. The lack of knowledge about how these factors affect individuals' performance in emergencies undermines organizations' ability to manage safety. This is especially important in the offshore oil and gas industry where facilities tend to be remote and external emergency response is not immediate (Flin, Slaven, & Stewart, 1996).

Method

The methodology followed for this experiment is outlined in Figure 1. As with any HRA, a task analysis was completed prior to the assessment. The tasks of interest here are the required safety compliance procedures associated with emergency situations. After the task analysis, a list of factors that can influence task performance was selected. The next step was to model the dependency between the PSFs and task performance using BN. Once the BN was developed, scenarios were created in the VE to get the data required to quantify the BN. Data was then collected by observing participants' performance in the virtual scenarios. Finally, the collected data were integrated into the BN and the reliability of participants was assessed.

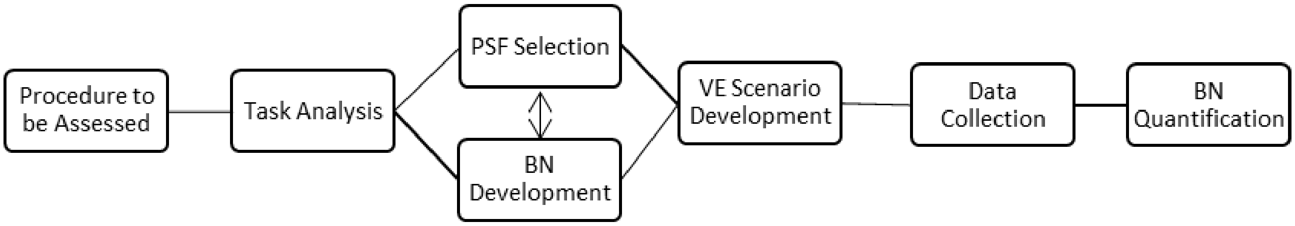


Figure 1: Methodology

Results & Discussion

Results show that the BN method is effective at determining the probability of success of an individual in offshore emergency situations using a VE. It allows for the completion of a HRA analysis using data that is not based on expert opinion. Participants' retention of the information provided in the training, and application of the knowledge and skills they acquired was observed. By combining a BN with data from a VE, predictions can be made about how successful a participant will be in real-life emergency situations.

The results also demonstrate that manipulation of a BN model is useful in investigating the cause-effect relationship between the performance shaping factors and the success probability (Groth et al., 2013; Musharraf et al., 2014). For example, by providing evidence of a change in PSF, the effect on the participant's success probability is shown in the updated BN model. This information is valuable as it can lead to adaptive training that is specific to the individual. Results that demonstrate similar errors across all participants are also useful. These results can provide information that can lead to an improvement in safety critical procedures and physical workplaces.

Conclusion

This poster presents a methodology to investigate the effect of PSFs on offshore emergency evacuation using a virtual environment. The results show that a person's response during an emergency can be probabilistically quantified using the BN approach. This study focuses on individual human factors. Future work would benefit from inclusion of organizational and teamwork factors.

Acknowledgments

The authors acknowledge with gratitude the support of the NSERC-Husky Energy Industrial Research Chair in Safety at Sea.

References

Flin, R., Slaven, G., & Stewart, K. (1996). Emergency decision making in the offshore oil and gas industry. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *38*(2), 262-277.

Groth, K. M., & Mosleh, A. (2012b). Deriving causal Bayesian networks from human reliability analysis data: A methodology and example model. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, *226*(4), 361-379.

Groth, K. M., Smith, C. L., & Swiler, L. P. (2014). A Bayesian method for using simulator data to enhance human error probabilities assigned by existing HRA methods. *Reliability Engineering and System Safety, 128,* 32-40.

Groth, K. M., & Swiler, L. P. (2013). Bridging the gap between HRA research and HRA practice: A Bayesian network version of SPAR-H. *Reliability Engineering and System Safety: 115*, 33-42.

Musharraf, M., Hassan, J., Khan, F., Veitch, B., MacKinnon, S., & Imtiaz, S. (2013). Human reliability assessment during offshore emergency conditions. *Safety Science, 59*, 19-27.

Musharraf, M., Bradbury-Squires, D., Khan, F., Veitch, B., Mackinnon, S., & Imtiaz, S. (2014). A virtual experimental technique for data collection for a Bayesian network approach to human reliability analysis. *Reliability Engineering and System Safety, 132*, 1-8.

Understanding category specific semantic deficits using a network mathematical tool

Kaoutar Skiker (skiker.kaoutar85@gmail.com)
Department of Mathematics and Computer Science, ENSAT,
Abdelmalek Essaadi's University, Tangier, Morocco

Mounir Maouene (mounir.maouene@gmail.com)
Department of Mathematics and Computer Science, ENSAT,
Abdelmalek Essaadi's University, Tangier, Morocco

Abstract

In this paper, we propose a novel quantitative approach to understand why some neurodegenerative diseases such as Alzheimer's disease affect some semantic categories more than others, known in cognitive neuropsychology as category specific semantic deficits. First, we represent semantic features of animal and tool semantic categories using a network mathematical tool. Second, we develop an algorithm building on the idea that, in Alzheimer's disease, semantic memory is gradually degraded with features specific to some objects are lost first then more general ones. We apply this algorithm to the networks of animal and tool features and examine how the gradual degradation of features affects their structures using graph measures. Our results show that, at the early stage, when distinctive features were lost, both networks are robust against the gradual loss of semantic features. However, when shared features were removed, the network of tool features appears to be more sensitive to attacks because the network disconnects into 20 components, compared to the network of animal features that disconnects into only two components. This approach provides a promising way to understand category specific semantic deficits.

Keywords: semantic memory, network model, category specific semantic deficits, degradation of semantic knowledge, small world property, semantic memory impairments, dementia

Modeling Relational Reasoning in the Neural Engineering Framework

Markus Lohmeyer (markus.lohmeyer@gmx.de)
Cognitive Computation Lab, University of Freiburg, Germany

Julia Wertheim (julia.wertheim@tf.uni-freiburg.de)
Cognitive Computation Lab, University of Freiburg, Germany

Abstract

The computational modeling of cognitive processes can be an insightful endeavor for understanding the fundamental mechanisms of cognition. In order to simulate relational reasoning, we modeled three-term tasks using mental model theory as a theoretical framework. We implemented the model in the Neural Engineering Framework (NEF) which provides insights into how spiking neural networks facilitate or aggravate this process. We were able to address issues and peculiarities arising from symbol based relational reasoning implemented in connectionist networks.

Keywords: Relational reasoning; Neural Engineering Framework; Nengo; Cognitive modeling; Transitive reasoning; Theory of mental models

Introduction

Relational reasoning is a cognitive capability on which many aspects of human actions are based on, for instance intelligence (Waltz et al., 1999). An example for this is reasoning about locations: If it is known that the supermarket is to the north of the gas station and the gas station is to the north of the church, we can easily determine that the church is to the south of the supermarket. Although relational reasoning is crucial, its cognitive and neural mechanisms remain largely unclear. Yet, psychological hypotheses have been formulated. One well-known hypothesis invoking the application of mental models has been proposed by Johnson-Laird (1983). According to mental model theory, all relevant information is integrated in one model representing the configuration of all objects. To derive the conclusion whether a mental model is valid or invalid, three distinct phases are passed. Phase one, construction, involves the initial creation of the mental model while phase two, inspection, comprises the generation of new information. In the third phase, variation, the creation of alternative mental models is elicited to disprove the information gathered in phase two (Fangmeier, Knauff, Ruff & Sloutsky, 2006).

Since relational reasoning has been researched for several decades, cognitive models of it are quite common. For example, spatial relational reasoning was successfully modeled in the symbolic framework ACT-R (Ragni, Fangmeier & Brüssow, 2010). However, since the theory of mental models is based on the usage of symbols, an implementation in neural networks has not been attempted so far. One approach to tackle this issue are vector symbolic architectures like the NEF. These use neuronal clusters to represent, modify and combine vector symbols. Due to these, it is possible to model symbolic reasoning in connectionist networks. Nevertheless, modeling within a vector symbolic approach differs greatly from classic production systems like ACT-R. The first step to expand relational reasoning towards vector symbolic architectures is the implementation of transitive closure in its simplest form. By that, we are able to show how relational reasoning can be implemented in the NEF. Subsequently, we address several issues resulting from this architecture.

The Model

The Task

As proposed by Fangmeier et al. (2006), we aim for a presentation of the three-term tasks which abstracts from reading processes which might influence reasoning. Accordingly, the presentation is geared to the fMRI study by Fangmeier et al. (2006). Here, premises like ‘A is left of B’ are presented without the usage of syntax. Their position in the premise is coded in their screen position. In the abovementioned example, ‘A’ is presented on the left side of the screen, while ‘B’ is presented on the right side. After the presentation of the second premise, a putative conclusion is presented that is confirmed or refused by button press.

The Modules

The model consists of five parts accomplishing diverging tasks. The visual system module processes information about an object and its location and combines these. The mental model module consists of several buffers containing the elements of the mental model. The compare module is connected to the part of the visual system processing the objects and to the different buffers that are designated for the mental model formation. The buffer module comprises two buffers to store information which cannot be processed immediately and one additional buffer providing information about the model’s state. The motor module simulates the button press. This ensemble of modules is complemented by the NEF’s basal ganglia system containing and triggering the right action depending on the situation.

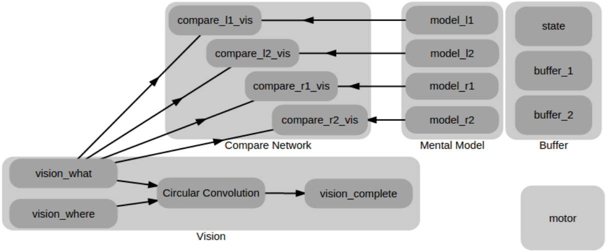


Fig 2.: Schematic illustration of the model's main components and connections.

The Reasoning Process

After the presentation of the first premise, a preliminary mental model is created in the buffers L1 and R1 whereat the left term is stored in L1 and the right one in R1. With the presentation of the second premise, two cases can be differentiated. In the simple case, the first object of the second premise is already known from the first premise. With knowledge of the common term and its position in the second premise, the position of the last term is already determined, leading to an easy integration. In the hard case, the first term of the second premise is not known and its position in the mental model remains unclear until the second term is presented. Here, the first term and its position are stored in the buffer module. After the presentation of the second term, all necessary information is present and the integration of the third term can proceed. One can easily see that comparing objects in the mental model with objects in the visual module is paramount for transitive closure.

With the presentation of the putative conclusion, the model needs to verify it with regard to the mental model at hand. For this purpose, it must determine if the configuration in the putative conclusion is identical to that in the mental model. To achieve this goal, objects from the putative conclusion are compared to the mental model and the matching positions of the model are written in the buffer module. Here, one must consider that buffer_1 stores the position within the mental model that the left term of the putative conclusion corresponds to, while buffer_2 does the same for the mental model position corresponding to the right term of the putative conclusion. Having gathered and stored this information, rules from the basal ganglia can apply on different configurations and determine whether a putative conclusion is consistent with the mental model.

Results

Generally, it was possible to model relational reasoning in the NEF. However, since the NEF does not impose any restrictions and presumptions considering the modeling of higher cognition, it is impossible to compare the output data with human behavioral data. Hence, all findings only refer to concepts and structures when modeling relational reasoning in the NEF. Considering the model, the compare module is well connected since it receives information from the mental model and the visual input. Additionally, it shows a distinct behavior regarding the objects in the mental model and the visual module. Hypotheses declaring a compare network could be reviewed under these criteria. Furthermore, the comparison process depends on the availability of objects without the location bound to it. That is why we implemented a ventral ('what') and a dorsal ('where') pathway for the visual perception (Goodale & Milner, 1992). Consequently, connections between the compare and the visual module should only exist within the ventral stream.

Before the compare module came to its final form with every object of the mental model compared in parallel, serial approaches have been tested. With these, it is hard to maintain the information flow in a reliable manner if every object is processed serially. Consequently, there are problems that our model cannot discriminate. Parallel comparison can be refuted, if empirical data suggest that there is a difference between these problems.

Discussion

Our model provides valuable insights into the successful modeling of higher cognition in the NEF. Nonetheless, an issue is the difficulty of comparison with behavioral data. This could only be resolved by modeling all parts of the brain that are involved in the reasoning process. Herewith, quantitative comparisons and evaluations of the model would generally be possible but not with the nowadays limited knowledge about the specifications constituting the human brain. Nonetheless, the model offers insight into the peculiarities of modeling higher cognitive processes within the NEF. Further, it suggests how to design experiments that might shed a light on the issues.

References

Ragni, M., Fangmeier, T., & Brüssow, S. (2010, August). Deductive spatial reasoning: From neurological evidence to a cognitive model. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 193-198). Philadelphia, PA: Drexel University.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford University Press.

Fangmeier, T., Knauff, M., Ruff, C. C., & Sloutsky, V. (2006). fMRI evidence for a three-stage model of deductive reasoning. *Journal of Cognitive Neuroscience*, *18*(3), 320-334.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in neurosciences*, *15*(1), 20-25.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness* (No. 6). Harvard University Press.

Waltz, J. A., Knowlton, B. J., Holyoak, K. J., Boone, K. B., Mishkin, F. S., de Menezes Santos, M., ... & Miller, B. L. (1999). A system for relational reasoning in human prefrontal cortex. *Psychological science, 10(2)*, 119-125.

Detecting Macro Cognitive Influences in Micro Cognition: Using Micro Strategies to Evaluate the SGOMS Macro Architecture as implemented in ACT-R

Robert L. West (robert_west@carleton.ca), Nathan Nagy (Nathan.nagy@carleton.ca), Fraydon Karimi (Fraydon.karimi@Carleton.ca), & Kate Dudzik (Kate.dudzik@carleton.ca)

Institute of Cognitive Science, Carleton University, 1125 Colonel By Drive,
Ottawa, Ontario, Canada, K1S 5B6

This study uses a methodology similar to Gray & Boehm-Davis (2000) to look for evidence of the SGOMS architecture (West & Macdougal, 2014) in a simple response task. SGOMS is a theory of how people manage complex real world tasks involving expertise. The SGOMS macro architecture is a specific way of structuring an ACT-R to model expertise. However, it is not the most efficient way to model a simple, uninterrupted task in ACT-R. This is because SGOMS uses extra productions to handle unexpected interruptions and to save information for re-planning. SGOMS also uses extra productions to manage the subcomponents of SGOMS (planning units, unit tasks, methods, and operators) so that the productions representing these units can be flexibly re-deployed during the task. The SGOMS macro architecture deploys these extra productions as efficiently as possible, but they still result in extra processing time due to the single production bottleneck in the ACT-R procedural memory system. Therefore, an SGOMS model will systematically vary from the optimal (fastest) model in ACT-R.

Procedure

Subjects

Two of the authors, NN and FK volunteered as subjects.

Method

On each trial subjects were presented with a four letter cue code and had to respond with the appropriate two letter response code. Therefore, every trial was exactly the same in terms of response actions. The only difference was that sometimes participants knew what code followed another, and sometimes they needed to perceive the cue code to know which code to respond with.

Subjects first learned the three *unit tasks* depicted in Figure 1. Subjects practiced until they were satisfied they had attained their best speed and accuracy. To do this, subjects trained at home on their laptop computers. Next, subjects memorized the three *planning units*, described in Table 1. Each planning unit was cued by a four-letter code and each was composed of the three unit tasks. Two of the planning units were *ordered* planning units, with the unit tasks in different orders. The third planning unit was a *situated* planning unit that cued each unit task once, in random order. Again, subjects practiced on their laptops at home until they were satisfied they were doing the task as quickly and accurately as possible. Once this stage was reached we collected the data for analysis.

The optimal model worked in the following way: the code for the current planning unit was stored in the imaginal buffer (i.e., working memory) and the production representing the correct response was selected by matching with this information and the current code, which was in the visual buffer. The perceptual/motor methods were the same as the SGOMS model.

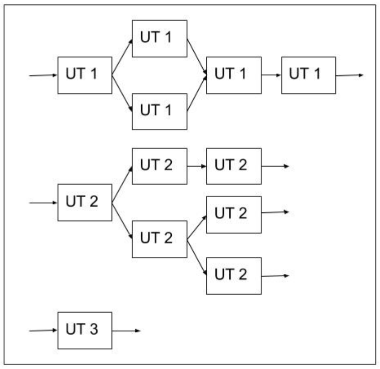


Figure 1. Unit task structure: Each box represents a response within the unit task structure.

Planning Unit	Unit Task order
Ordered Planning Unit 1	Unit Task 1 Unit Task 2 Unit Task 3
Ordered Planning Unit 2	Unit Task 2 Unit Task 3 Unit Task 1
Situated Planning Unit 3	Unit Task ? Unit Task ? Unit Task ?

Table 1. Planning Unit structure.

Results

To evaluate the results we divided the trials into different categories corresponding to different predictions of the SGOMS architecture. In SGOMS, an action occurring inside a unit task occurs as it would in the optimal ACT-R model. That is, there is no overhead. These response categories were labeled Unknown Unit Task Middle (Unknown Mid UT) and Known Unit Task Middle (Known

Mid UT), where Known refers to conditions in which the subject knew the next response and Unknown refers to conditions in which the subject had to read the new code to know the right response (see Figure 2).

These two conditions formed the baseline for fitting the results. After we removed errors and outliers more than two standard deviations from the mean, we saw that FK's response times were considerably faster than NN. We assumed that this was a perceptual motor difference and equalized the response times for both baseline conditions by subtracting the difference between FK's average RT and NN's average RT from NN's average score. We applied this same correction to all of the other conditions according to whether the response was known or unknown.

We also fit the SGOMS model and the optimal model to FK's results for these two conditions. Since both models made the same predictions for the baseline conditions, and both used the same perceptual motor model, the adjustment was the same. Therefore, for the two baseline conditions, the adjusted results of the two subjects and two models were identical. Assuming that the perceptual/motor responses remained constant across the other conditions, all differences on the other conditions between the subjects and between the models were due to differences in cognitive processes. No adjustments or modifications were made for the other conditions

Figure 2 shows the results with 0.05 confidence intervals for our subjects' data. For all conditions NN was similar to the SGOMS model and dissimilar to the optimal model. For the bottom two conditions in Figure 2, NN falls outside the confidence intervals for the SGOMS model, but is still closer to the SGOMS model than the optimal model. FK was similar to NN and the SGOMS model for all except the bottom three conditions, where he was more similar to the optimal model. FK reported experimenting with specific strategies to speed up the task, which we believe accounts for the differences he showed.

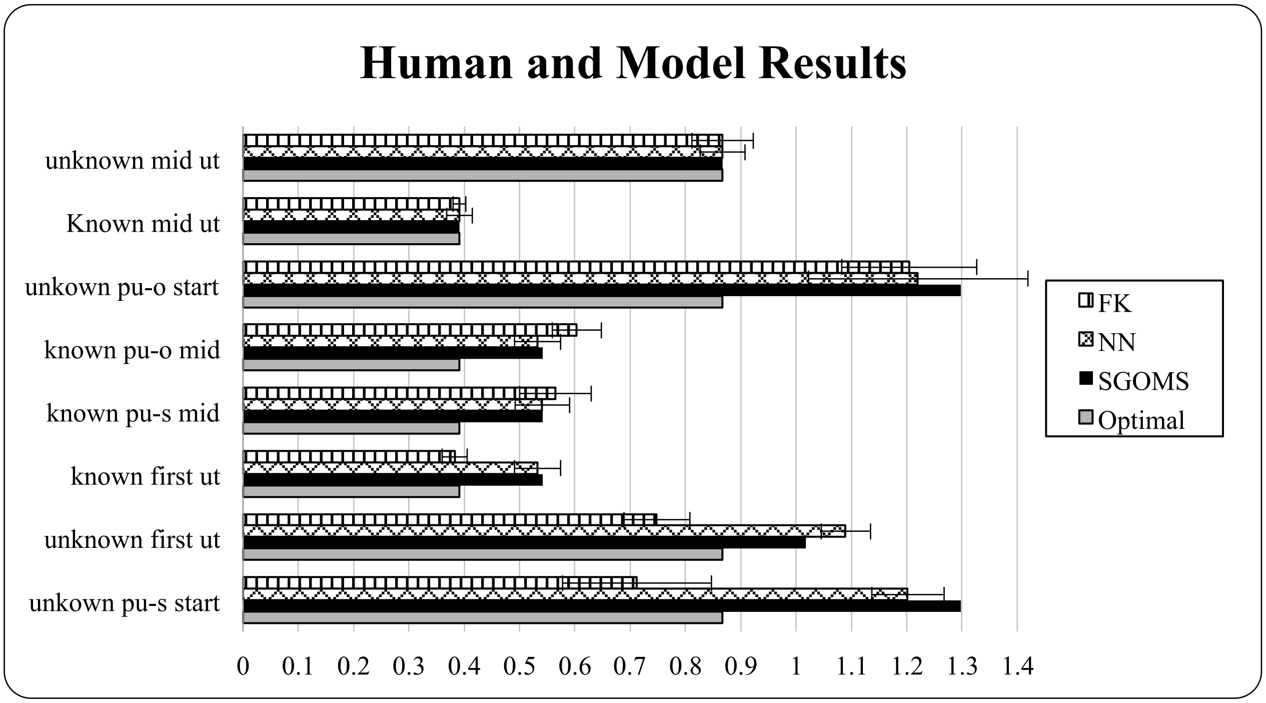


Figure 2. Human and model results.

Conclusion

Our results clearly show support for the SGOMS macro architecture. The SGOMS model provided a better fit to the data for six out of six response categories for NN and for three out of six response categories FK. For FK, in the three cases where the SGOMS model did not match the data, the optimal model provided a reasonable match. This pattern of results fits well with our claim that people use the SGOMS architecture as their default system, and only later convert to an optimal form if the task can be performed without interruptions and they have the motivation for thinking about it and practicing it.

References

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.

West, R. L., & Macdougall, K. (2014). The macro-architecture hypothesis: Modifying Newell's system levels to include macro-cognition. *Biologically Inspired Cognitive Architectures,* 8, 140-149.

Author Index