Decision making in a dynamically structured holographic memory model: Learning from delayed feedback

Matthew A. Kelly (matthew_kelly2@carleton.ca)
Robert L. West (robert_west@carleton.ca)
Institute of Cognitive Science, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6 Canada

Abstract

We present a first step towards developing a cognitive model of decision making using a Dynamically Structured Holographic Memory (DSHM). DSHM is a class of memory models used to model human performance in game play, memory tasks, and language learning. The ability to detect and predict patterns is an integral part of memory models and an important part of decision making. However, decision making also requires the ability to evaluate states as more or less desirous in order to motivate the decisions. We apply a variant of the DSHM model to the decision-making task of Walsh and Anderson (2011). By initializing memory to a state of optimism and by making the model sensitive to dependencies between non-consecutive events, we find that the model is able to learn the task at a rate similar to humans.

Keywords: Vector symbolic architectures; Holographic reduced representations; cognitive modelling; memory; learning; decision making; BEAGLE; DSHM.

This paper details the first step in developing a decision-making model from a DSHM/BEAGLE model.

BEAGLE (Jones & Mewhort, 2007), a model of semantic memory, has been used to model how humans develop an understanding of the meaning and part-of-speech of words from experience with language. Dynamically Structured Holographic Memory (DSHM) is a variant of the BEAGLE model applied to non-language stimuli. DSHM has been used to model the fan-effect memory task (Rutledge-Taylor & West, 2008), to model human performance in game playing (Rutledge-Taylor & West, 2011), and has also been used for AI, namely as a recommender-system for films and research papers (Rutledge-Taylor, Vellino, & West, 2008). Rutledge-Taylor and West (2008) have noted that DSHM can be understood as a lower-level implementation of the declarative memory in the cognitive architecture ACT-R (Anderson & Lebiere, 1998), with the advantage that DSHM/BEAGLE is more flexible and better suited for long-term learning of large quantities of data, such as an individual's lifetime language experience.

While memory models can be powerful tools for detecting structure within data and making predictions from experience, memory models are generally only used to make very simple decisions, such as deciding whether a stimulus is familiar or novel in a familiarity task, or when to 'give up' trying to recall more items in a free recall task.

Conversely, cognitive architectures such as ACT-R and Soar (Laird, 2012) are designed to make complicated sequences of decisions to solve problems. ACT-R uses two systems: a procedural memory and a declarative memory. Procedural memory consists of condition-action rules weighted by utility scores. Declarative memory consists of chunks of associated data weighted by activation values.

Both the activation of a chunk in declarative memory and the utility of a condition-action rule in procedural memory can be understood as estimates of the usefulness of the chunk or rule. According to Anderson's (1991) rational analysis, the activation of a chunk in memory is an estimate of the likelihood of the information in the chunk being useful in the current situation. Likewise, a production rule's utility is an estimate of the usefulness of performing the rule's action. However, these two kinds of usefulness are quite distinct. For example, it is very useful to *know* that touching a hot object will burn you, but it is not very useful to *touch* a hot object. Thus the knowledge of the consequences of touching a hot object will have a high activation in declarative memory but the utility of touching hot objects as a rule in procedural memory will be low.

More generally, the probability of an event is distinct from the utility of the event. Thus, while computational models of human memory are well-equipped to estimate probabilities, they are not designed for decision making because they do not estimate utility.

Adapting DSHM to decision making is attractive because the complexity and intelligence of human decision making is thanks, at least in part, to the immense quantity of experiential knowledge that humans draw upon. A DSHM decision-making model may be able to scale up to decision-making tasks that require a great deal of knowledge.

In what follows, we model human performance in a sequential decision task. This task has already been modelled as a utility learning task by Walsh and Anderson (2011) using temporal difference learning models that can be incorporated into the ACT-R procedural memory. We instead model the task as a prediction task using DSHM.

Our approach to modelling decision making is related to the way DSHM models game playing (Rutledge-Taylor & West, 2011) and is conceptually similar to ACT-R models that use declarative, rather than procedural, memory to make decisions (e.g., Lebiere, Gonzalez, & Martin, 2007). Our model uses a contextualized, instance based memory to understand each situation and decide what to do next.

Experiment

We summarize the pertinent details of the Walsh and Anderson (2011) experimental task here.

Figure 1 illustrates the structure of a single trial of the experiment. Boxes are states in the experiment. Arrows indicate transitions. Thicker arrows indicate more probable transitions. Unlabelled arrows have a probability of 100%.

In each trial of the experiment, participants make two choices that affect their chance of receiving positive feedback at the end of the trial. At the start of each trial, participants were presented with their first choice: a pair of letters, one on the left of the screen and one on the right. By pressing the corresponding arrow key, participants selected either the left or right letter. After their selection, a cue shape flashed onto the screen for over a second. Participants then selected a second letter from a different pair of letters. After their second choice, one of two symbols appeared indicating either positive or negative feedback. This ended the trial after which a new trial would begin.

The position of a letter on either the right or left side of the screen was randomized from one trial to the next, ensuring that participants could not rely on learning a pattern of motor responses to perform the task.

Additionally, the particular letters, shapes, and symbols used in the task was randomized across participants to control for any effect of the symbols themselves.

However, for the purposes of discussing the task, we will use the letters **R** and **J** to represent the first binary choice, ▮ and ◆ to represent the pair of shapes used as cues, **T** and **V** to represent the second binary choice, and we will refer to the pair of symbols that represent positive and negative feedback as **good** and **bad**.

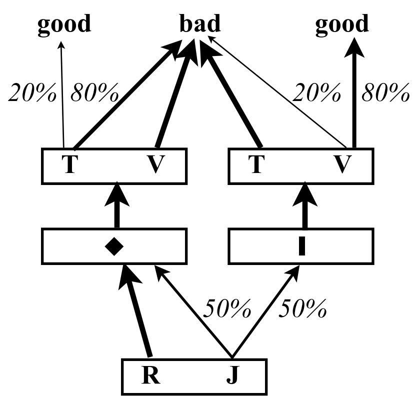

Figure 1: Experimental states and transition probabilities. Adapted from Figure 2 in Walsh & Anderson (2011).

As illustrated in Figure 1, there are three distinct decisions to be made by the participant. For each decision, there is a correct choice, that is, a choice that maximizes the chance of receiving positive feedback. Between **R** and **J**, **J** is correct. After ◆, **T** is correct, whereas after ▮, **V** is correct.

Walsh and Anderson (2011) measured participant response accuracy after 200 and 400 trials. Additionally, Walsh and Anderson used scalp-recorded event-related potential (ERP) to detect feedback-related negativity, a hypothesized neural correlate of prediction error. We attempt to model only response accuracy. We believe that prediction error, or surprise, could be modelled using DSHM, however, that is beyond the scope of this paper.

Dynamically Structured Holographic Memory

DSHM (Rutledge-Taylor & West, 2008) and BEAGLE (Jones & Mewhort, 2007) are models based on Holographic Reduced Representations (HRRs; Plate, 1995). HRRs are a technique for instantiating and manipulating symbolic structures in distributed representations. When using HRRs, vectors represent symbolic information. These vectors, or symbols, can be combined and manipulated using a small number of operations, namely: vector addition, circular convolution, and permutation.

In BEAGLE and DSHM there are two types of vectors: *environmental vectors* and *memory vectors*.

An *environmental* vector is a vector that stands for atomic perceptions from the environment (e.g., a *red circle* needs two environmental vectors, one for *circle* and one for *red*). Environmental vectors are fixed and do not change. Environmental vectors are created by randomly sampling values from a normal distribution with a mean of zero and a variance of $1/n$, where n is the dimensionality of the vector.

A *memory* vector is a complex representation stored by the model and used to produce behaviour. There are a fixed number of memory vectors, and incoming environmental vectors are used to update them.

In BEAGLE the items of interest are words: an environmental vector stands for a word's orthography or phonology and a memory vector stands for the word's meaning. In DSHM, the items of interest are objects relevant to the experimental task: an environmental vector stands for the perception of a thing and a memory vector stands for the concept of that thing.

For example, to create an association between *keyboards* and *computers*, each time a *computer* and *keyboard* co-occur a copy of the environmental vector $\mathbf{e}_{keyboard}$ can be added to the memory vector $\mathbf{m}_{computer}$ and likewise a copy of $\mathbf{e}_{computer}$ can be added to $\mathbf{m}_{keyboard}$. The effect of this would be to move $\mathbf{m}_{computer}$ closer to $\mathbf{e}_{keyboard}$ and $\mathbf{m}_{keyboard}$ closer to the $\mathbf{e}_{computer}$. Over time, the result of this is to organize the space so that memory vectors are clustered around environmental vectors that they co-occur with such that the distance between the vectors equals strength of association.

Another, more complicated example uses circular convolution (denoted by *). Consider the phrase or stimulus *blue triangle*. We could update memory as follows:

$$\mathbf{m}_{blue} \mathrel{+}= \mathbf{e}_{blue} * \mathbf{e}_{triangle}$$
$$\mathbf{m}_{triangle} \mathrel{+}= \mathbf{e}_{blue} * \mathbf{e}_{triangle}$$

By convolving together the environmental vectors (**e**) for *blue* and *triangle* and adding the result to the memory vectors (**m**) for *blue* and *triangle* (an operation denoted by +=), we move the two memory vectors towards the point in space described by the vector $\mathbf{e}_{blue} * \mathbf{e}_{triangle}$, and thereby move $\mathbf{m}_{blue}$ and $\mathbf{m}_{triangle}$ closer together, indicating similarity between the two concepts. But this is a bad model because people almost *never* get the concepts *blue* and *triangle* confused with each other. This is because *blue* is a colour (or an adjective), and *triangle* is a shape (or a noun), i.e. they are different sorts of thing.

Conversely, consider updating memory using the *placeholder vector* (denoted by **Φ**). The placeholder is a random vector used to encode all associations and thus acts as a universal retrieval cue. We update memory as follows:

$\mathbf{m}_{blue} \mathrel{+}= \boldsymbol{\Phi} * \mathbf{e}_{triangle}$
$\mathbf{m}_{triangle} \mathrel{+}= \mathbf{e}_{blue} * \boldsymbol{\Phi}$

$\mathbf{m}_{blue}$ moves towards $\boldsymbol{\Phi} * \mathbf{e}_{triangle}$, i.e. towards all properties of triangles, and $\mathbf{m}_{triangle}$ moves towards $\mathbf{e}_{blue} * \boldsymbol{\Phi}$, i.e. towards all things that are blue. Thus, by using a placeholder, the memory vectors for nouns will cluster together in one region of space, and the memory vectors for adjectives will cluster together in another region of space, and things that are *colours* will cluster separately from things that are *coloured*. This is a subtle distinction but Jones and Mewhort (2007) have shown it to be very important and very powerful.

The Model

Presentation of the experimental task is simplified for the model. In this simplified format for the experiment, there are nine distinct symbols. There is a symbol that indicates the start of a trial (which we will refer to as **start**), two letters that represent the first binary choice (**R** and **J**), a pair of shapes used as cues (▮ and ◆), a second pair of letters used to represent the second binary choice (**T** and **V**) and a pair of symbols to represent positive and negative feedback (which we will refer to as **good** and **bad**).

At the beginning of the experiment, the model generates a an environmental vector and a memory vector for each of the nine symbols used in the experiment. Vectors are high dimensional. In the model runs discussed in this paper, 1024 dimensions were used. Noise can be increased by using fewer dimensions, especially fewer than 512 dimensions.

Each environmental vector represents the perception of a symbol, i.e., what the model sees when the symbol is presented. The environmental vectors are generated randomly and are not modified over the course of an experimental run, but are different for each artificial participant simulated by the model.

Each memory vector represents the concept of a symbol, i.e., everything the model knows about that symbol. The memory vectors are initialized to a state of optimism, such that the model initially believes that any symbol could be followed by **good**. Noise is added to the memory vectors to create random variation in the initial choices made by the simulated participants. The initialization of the memory vectors is explained in detail under subheading *Initializing memory to a state of optimism.* The memory vectors are used to make decisions and are updated at every iteration over the course the experiment.

The experiment is run for 400 trials. Each trial consists of five iterations as illustrated in Table 1. On the first iteration in a trial, the **start** symbol appears and the model must select **R** or **J**. On the second iteration, the selected letter appears. On the third iteration, a cue appears (either ◆ or ▮) and the model must select either **T** or **V**. On the fourth iteration, the selected letter appears. On the fifth iteration, either **good** or **bad** appears and then a new trial begins.

The model keeps track of what symbols have occurred so far in the current trial and the order in which those symbols occurred. This information can be understood as being held in a working memory buffer, but how this information is held is not a theoretical commitment of the model.

Table 1: The symbols presented to the model and choices available at each iteration in a trial.

Iteration	1	2	3	4	5
Presented	**start**	**R** or **J**	◆ or ▮	**T** or **V**	**good** or **bad**
Choices	**R** or **J**		**T** or **V**		

On each iteration of the experiment, the model updates the memory vectors corresponding to each symbol that has appeared so far in the current trial. If, on the third iteration of a trial, the symbols presented so far are "**start J ◆**", then, on that iteration, the model will update the memory vectors for **start**, **J**, and ◆. If the model selects **T**, then on the fourth iteration it will be presented the letter **T**, and the memory vectors for **start**, **J**, ◆, and **T** will be updated. Memory vectors are updated with observed relationships between symbols, detailed in *Updating memory using open n-grams.*

Decisions are made by selecting the choice which has the greatest similarity to the query vector. The query vector is constructed to ask the question, "Of the choices available, which choice is more likely to be followed by **good**?". This is detailed in *Querying memory*.

Gradually, the model learns that for each choice, there is a better alternative. Learning happens for each of the three choices at roughly the same rate as human participants performing the task, as illustrated in the next section, *Comparing the model to human performance*.

Developing this model required addressing two questions that have not been previously addressed in DSHM or BEAGLE models, namely, how to model motivation and how to learn dependencies between non-consecutive events. The question of modelling motivation is addressed in *Representing positive and negative feedback* and in *Initializing memory to a state of optimism.* The question of learning dependencies between non-consecutive events is addressed in *Updating memory using open n-grams.*

Comparing the model to human performance

Walsh and Anderson (2011) had 13 participants in their experiment. Each participant performed two blocks of 400 trials. Walsh and Anderson present data aggregated across the 26 experimental blocks. Figure 2 depicts mean response accuracy with standard error indicated by error bars.

For the purposes of comparison, we average across 26 runs of the model. On each run, the model performs 400 trials. Comparing Figures 2 and 3, we see that performance of the model roughly mirrors human performance. Performance for both humans and the model is better in the latter half of the trials. Both humans and the model learn that **V** is the correct choice after ▮ with relative ease, learn that **J** is a better choice than **R** more slowly, and have the most difficulty learning that **T** is the correct choice after ◆.

The reliability with which the model learns to choose **T** after ◆ varies with a parameter, the *optimism coefficient.* This parameter is explained in the section *Initializing memory to a state of optimism*. The parameter encourages exploration, but if the value is too high, the model tends to explore too much and insufficiently exploit knowledge of the task gained by this exploration. Figure 3 shows results for an *optimism coefficient* of 30.

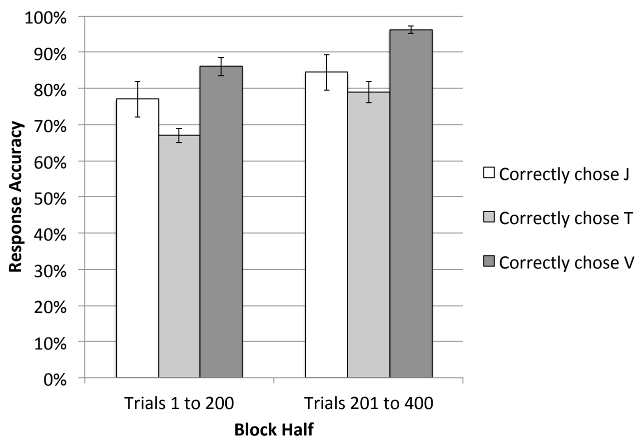

Figure 2: Response accuracy for human participants. Adapted from Figure 3 in Walsh & Anderson (2011).

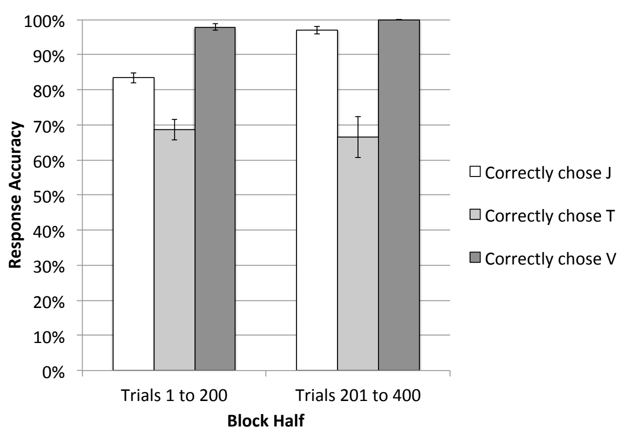

Figure 3: Response accuracy for the model.

While the model captures the overall pattern of human performance, there are two noteworthy differences.

First, the model often learns to do the task nearly perfectly, whereas humans participants do not tend to do so. The model's rate of correctly choosing **V** after ▮ reaches ceiling in the first block half and so does not improve significantly in the second block half. However, the model's accuracy can be impaired by using vectors with far fewer dimensions. In Figure 3, we use 1024 dimensions; 256 may produce a slightly closer fit to human performance.

Second, the model's mean rate of correctly choosing **T** after ◆ does not increase in the second block half. Improving the model's performance after ◆ may require implementing an attentional mechanism to allow the model to learn that the relationship between ◆ and **T** is critical to the feedback received and thus needs to be weighted more heavily. Implementing an attentional mechanism is discussed in *Querying memory.*

Representing positive and negative feedback

In the experimental task, participants receive positive or negative feedback at the end of each trial. Positive feedback and negative feedback take the form of arbitrary symbols that the participant has been told indicate "1 point gained" or "no points gained" respectively. Participants in Walsh and Anderson's (2011) experiment were promised $1.00 for every 50 points they gained.

Our best model makes no attempt to represent positive and negative feedback as being associated with positive or negative feelings. The **good** symbol and **bad** symbol are represented, like any other symbol in the experiment, as randomly generated vectors. The model's goal is to act to maximize the likelihood of getting **good** to appear at the end of the trial. The model avoids choices that are likely to cause **bad** to appear simply because those choices are less likely to cause **good** to appear.

We have explored modelling the **bad** symbol as an aversive stimulus. This is implemented by making **bad** the negation of **good**. We find that when bad is represented this way, the model becomes, as one might expect, risk averse and reluctant to explore possibilities. As a result, when **bad** is the negation of good, the model never learns that **T** yields a 20% chance of **good** after the cue ◆. Instead, the model chooses **V** regardless of which cue has been presented because it quickly learns that **V** often yields **good** and is reluctant to try its chances with **T**. We speculate that if humans were subjected to an aversive stimulus (e.g., an unpleasant image) every time they received negative feedback in this task, we might observe an impairment in problem solving performance similar to when **bad** is modelled as the negation of **good**. Testing this prediction of the model is a matter for future work.

Conversely, representing **bad** as a neutral stimulus encourages the model to try selecting options that have frequently yielded **bad**, which, when combined with optimism, leads to a more thorough exploration of the choices available within the task, allowing the model to better understand the consequences of its choices and discover the optimal pattern of decisions (see Figure 3).

The contrast between how the model behaves when **bad** is modelled as aversive versus neutral is consistent with *broaden-and-build theory* of positive emotions, which holds that positive emotions encourage exploration by broadening the repertoire of actions considered, whereas negative emotions narrow that repertoire (Fredrickson, 2001).

Initializing memory to a state of optimism

At the start of the experiment, memory vectors are initialized to a state of optimism. For all symbols in the task (except for **good** and **bad**), the model initially believes that symbol may later be followed by the symbol **good**.

The purpose of optimism is to motivate the model to select untried (or infrequently tried) choices. Without optimism, the model would simply repeat the same choices over and over again, as the model has no reason to believe that the other options would be any better.

Lebiere et al. (2007) also encountered this problem in modeling decision-making and likewise decided to initialize their ACT-R model to a state of optimism.

Making **bad** an aversive stimulus is an alternative to optimism that motivates the model to avoid choices that yield **bad**. However, there are two disadvantages to this approach: (1) Human participants are not punished for incorrect choices: they are rewarded for correct choices, and so such a model is performing a different task than the task that humans are asked to perform, and (2) either with or without optimism, variants of the model that are punished for incorrect choices fail to learn that **T** is the correct choice after the ◆ cue, as discussed in the previous section.

Without optimism, the memory vector for a symbol is initialized to be equal to the environmental vector for that symbol. This serves the purpose of adding random noise such that the initial decisions made by the model vary from one simulated participant (i.e., run of the model) to the next.

With optimism, an expression meaning "this is followed by **good**" is added to the memory vector when it is initialized. For any symbol *s,*

$$\mathbf{m}_s = \mathbf{e}_s + a(\mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{good})$$

where $\mathbf{e}_s$ and $\mathbf{m}_s$ are respectively the environmental vector and memory vector for *s*, $\mathbf{e}_{good}$ is the environmental vector for the symbol **good**, **Φ** is the placeholder, a special vector that represents "this" as in "this is followed by **good**", *before* is a random permutation selected at the beginning of a run used to indicate that the placeholder comes before **good**, * is circular convolution, and *a* is the *optimism coefficient*, a scalar value and parameter of the model. Figure 3 shows performance of the model for $a = 30$.

Updating memory using open *n*-grams

In the experiment, the choice **J** is more likely to lead to **good** than the choice **R**. However, there are two intervening events between feedback and the initial choice: the appearance of a cue and a second decision. In human memory, is a direct association formed between the first choice and the feedback at the end of the trial? Or do people rely on a chain of associations, learning that **J** leads to a particular cue, which in turn leads to a choice, which in turn leads to an increased probability of positive feedback?

With our model we have assumed that people form a direct association between their initial choice and the feedback received at the end of the trial. The memory update scheme used is a modification of the scheme used by BEAGLE (Jones & Mewhort, 2007) and inspired by Hannagan, Dupoux, and Christophe (2011).

Hannagan et al. (2011) considered a variety of models for how letter position in words is encoded in memory and found that the encoding scheme, *unconstrained open bigrams,* provided a good model of the kinds of errors people typically make in spelling. In this scheme, all pairs of letters within a word are associated with each other, even pairs of letters that are not next to each other in the word.

In our scheme, at each iteration of a trial, all symbols presented so far in a trial are associated with each other and all groups of symbols are associated with each other, even if the symbols do not occur consecutively.

For example, on the final iteration of a trial "**start R ◆ T good**" the memory vector for **R** is updated with,

$$\begin{aligned}
&\mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{\blacklozenge}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{\mathbf{T}}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\ \mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{\blacklozenge}) * \mathbf{e}_{\mathbf{T}}) * \mathbf{e}_{good}
\end{aligned}$$

i.e.,
"this came before **good**"
"this came before **◆ good**"
"this came before **T good**"
"this came before **◆ T good**"

where the placeholder vector **Φ** means "this" and acts as a stand-in for **R** for the purposes of memory storage. **R** is also updated with "this came after **start** and came before **good**", "this came after **start** and came before **◆ good**", etc.

Querying memory

The query or probe to memory is constructed as the sum of:

$$\begin{aligned}
&\mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\mathbf{e}_a) * \mathbf{\Phi}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\ \mathit{before}(\mathbf{e}_b) * \mathbf{e}_a) * \mathbf{\Phi}) * \mathbf{e}_{good} \\
&\ldots \text{ etc. } \ldots
\end{aligned}$$

where *a* is the symbol presented on this iteration, *b* is the symbol presented on the previous iteration, and so on, back to the beginning of the current trial. For the first decision in a trial, the probe vector is simply:

$$\begin{aligned}
&\mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\mathbf{e}_{start}) * \mathbf{\Phi}) * \mathbf{e}_{good}
\end{aligned}$$

For the second decision, the probe contains four terms, i.e.,

$$\begin{aligned}
&\mathit{before}(\mathbf{\Phi}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\mathbf{e}_{\blacklozenge \text{ or } ▮}) * \mathbf{\Phi}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\ \mathit{before}(\mathbf{e}_{\mathbf{R} \text{ or } \mathbf{J}}) * \mathbf{e}_{\blacklozenge \text{ or } ▮}) * \mathbf{\Phi}) * \mathbf{e}_{good} \\
&+ \mathit{before}(\ \mathit{before}(\ \mathit{before}(\ \mathit{before}\ (\mathbf{e}_{start}) * \mathbf{e}_{\mathbf{R} \text{ or } \mathbf{J}}) * \mathbf{e}_{\blacklozenge \text{ or } ▮}) \\
&\qquad\qquad * \mathbf{\Phi}) * \mathbf{e}_{good}
\end{aligned}$$

Decisions are made by measuring the similarity, as calculated by vector cosine, between the probe vector and the memory vectors of the two available choices. The symbol corresponding to the memory vector with the higher similarity to the probe is selected by the model as its choice.

Note that for the purposes of this particular model, the only symbols that need to have memory vectors are the choice symbols, represented here as **R**, **J**, **T**, and **V**. However, it is a theoretical or architectural commitment of DSHM/BEAGLE that all perceptual stimuli have an associated memory vector.

The probe vector is constructed to ask a distinct question for each term that comprises it. For the first decision, those questions are "Which of **R** or **J** is most likely to lead to **good**?" and "Given that the **start** symbol was just seen, which of **R** or **J** is most likely to lead to **good**?". For the second decision, four questions are posed by the probe: "Which of **T** or **V** is most likely to lead to **good**?", "Given the cue shape that was just seen, which of **T** or **V** is most likely to lead to **good**?", "Given the cue shape that was just seen and the previous decision that was made, which of **T** or **V** is most likely to lead to **good**?", and "Given the cue shape that was just seen, the previous decision that was made, and the fact that the start symbol was seen before that, which of **T** or **V** is most likely to lead to **good**?". The similarity of the probe vector to a particular choice's memory vector can be thought of as a mean of the likelihoods of that choice for each of those questions.

For example, in this experiment, **V** is, overall, very likely to be followed by **good**, but if **V** is preceded by the ◆ cue or the choice **R**, **V** will not be followed by **good**. The one question in the probe that is answered with "very likely" for **V** is answered with "not likely at all" by the other three questions, and as a result, when preceded with the ◆ cue, the model rightly estimates that the chances of **V** being followed **good** is very low and as a result the model may correctly select **T** in those circumstances instead.

One could introduce into the model an attentional mechanism that weights the terms of the probe. The appropriate value of these weights could be gradually learned by the model as it discovers which relationships are reliable and informative predictors and which relationships are not reliable or not useful predictors. Such a mechanism has not been implemented in this model.

Walsh and Anderson (2011) informed participants before the task "that the initial choice affected which cue appeared, and that the final choice depended only on the cue that appeared". Given this information, one might assume that participants only attended to two relationships:

(1) the relationship between the initial choice, the cue shape that appears, and the feedback at the end of a trial.
(2) the relationship between the cue, the final choice, and the feedback received.

In our model, we instead make the following assumptions:

(1) participants may not have fully absorbed the clues provided by the instructions
(2) attention and recall is, to a degree, automatic such that participants will pay attention to and be affected by things that they may know are irrelevant to the task
(3) and that the rapid pace of the experiment prevents participants from reasoning particularly carefully about the task and the decisions they are making.

Given these assumptions, we believe our model is a reasonable approximation to participant behaviour.

Conclusion

The work presented here suggests that DSHM/BEAGLE can be adapted to the task of modelling human decision making.

While Walsh and Anderson (2011) have already modelled the task using temporal difference learning models that can be incorporated into the ACT-R production system, our model adopts an approach that is importantly different in two ways. First, as in Lebiere et al. (2007), decisions are made on the basis of generalizing from learned instances. Two, DSHM is vector symbolic rather than symbolic.

Vector-symbolic memory systems can account for noise and generalization easily as part of the model's mechanisms and can efficiently and effectively model memory systems that contain very large quantities of data (Jones & Mewhort, 2007). These advantages motivate us to extend the reach of DSHM to decision-making.

In modelling human performance in this task, we addressed two novel questions for DSHM or BEAGLE models: (1) how to make the model sensitive to dependencies between non-consecutive events and (2) how to motivate the model to explore untried or infrequently tried choices and exploit choices associated with positive feedback. We accomplish the former by storing associations between non-consecutive events in memory. The latter we accomplish by initializing memory to a state of optimism, which encourages the model to explore.

There are some differences between the model's behaviour and the observed human behaviour, but our decision-making model is very simple, much simpler than what human participants are doing in the task. The model does not form, test, and reject hypotheses, nor is it curious, nor does it grow bored. Walsh and Anderson (2011) argue that the neural data from the experiment suggest a central role for surprise (or prediction error) in the task. Modelling surprise, attention, curiosity, or hypothesis testing, all provide routes forward for improving on this model.

References

Anderson, J. R. (1991). The place of cognitive architectures in a rational analysis. In K. Van Len (Ed.), *Architectures for Intelligence*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Lawrence Erlbaum Associates.

Fredrickson, B. L. (2001). The role of positive emotions in positive psychology: The broaden-and-build theory of positive psychology. *American Psychologist, 56*, 218-226.

Hannagan, T., Dupoux, E., & Christophe, A. (2011). Holographic string encoding. *Cognitive Science, 35*, 79-118.

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review, 114*, 1-37.

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

Lebiere, C., Gonzalez, C., & Martin M. (2007). Instance-based decision making model of repeated binary choice. *Proc. of the 12th International Conference on Cognitive Modelling,* 67-72.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks, 6*, 623– 641.

Rutledge-Taylor, M. F., Vellino, A., & West, R. L. (2008). A holographic associative memory recommender system, *Proc. of the Third International Conference on Digital Information Management*, 87-92.

Rutledge-Taylor, M. F., & West R. L. (2008). Modeling the fan-effect using dynamically structured holographic memory. *Proc. of the 30th Annual Conference of the Cognitive Science Society*, 385-390.

Rutledge-Taylor M. F., & West, R. L. (2011). Using DSHM to model paper, rock, scissors. *Proc. of the 33rd Annual Conference of the Cognitive Science Society*, 2341-2346.

Walsh, M. M., & Anderson, J. R. (2011). Learning from delayed feedback: Neural responses in temporal credit assignment. *Cognitive, Affective, and Behavioral Neuroscience, 11*, 131-143.