Holographic Declarative Memory and the Fan Effect: A Test Case for A New Memory Module for ACT-R

Matthew A. Kelly (matthew.kelly2@carleton.ca)
Kam Kwok (kamhung.kwok@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 Holographic Declarative Memory (HDM), a new memory module for ACT-R and alternative to ACT-R's Declarative Memory (DM). ACT-R is a widely used cognitive architecture that models many different aspects of cognition, but is limited by its use of symbols to represent concepts or stimuli. HDM replaces the symbols with holographic vectors. Holographic vectors retain the expressive power of symbols but have a similarity metric, allowing for shades of meaning, fault tolerance, and lossy compression. The purpose of HDM is to enhance ACT-R's ability to learn associations, learn over the long-term, and store large quantities of data. To demonstrate HDM, we fit performance of an ACT-R model that uses HDM to a benchmark memory task, the fan effect. We analyze how HDM produces the fan effect and how HDM relates to the standard DM model of the fan effect.

Keywords: ACT-R; memory; cognitive modeling; cognitive architectures; artificial general intelligence; integrated cognition; holographic reduced representations; vector-symbolic architectures

Introduction

Computational cognitive architectures provide the formal, unified theories of cognition necessary for cognitive scientists to achieve an understanding of the mind. ACT-R (Anderson & Lebiere, 1998) is a widely used cognitive architecture that can model diverse aspects of cognition. As an integrated architecture, ACT-R is a good choice for modelling complex tasks. However, the ACT-R Declarative Memory system (DM) was designed for modelling the results of psychology experiments and as such presents certain limitations for modelling complex, real world behaviour. In what follows, we present Holographic Declarative Memory (HDM), a new module for the ACT-R cognitive architecture that addresses some of DM's limitations. To help establish that HDM can provide the same functionality as ACT-R's DM, we have modelled the fan effect task (Anderson, 1974), analyzed how HDM generates the fan effect, and used this analysis to compare the HDM and DM models.

Holographic Declarative Memory (HDM) replaces ACT-R's symbols with holographic vectors. Holographic vectors retain the expressive power of symbols but have a similarity metric, allowing for shades of meaning, fault tolerance, and lossy compression of stored information.

HDM is based on BEAGLE (Jones & Mewhort, 2007), a learning algorithm that models how people abstract the meaning of words from their lifetime language experience, and DSHM (Rutledge-Taylor, Kelly, West, & Pyke, 2014), a model that uses a similar approach to BEAGLE but repurposes and extends the algorithm as a general memory model. HDM is implemented for Python ACT-R and the code for both Python ACT-R and HDM are available through GitHub[1]. Our intent with HDM is to replicate the basic functionality of DM and provide new capabilities.

First, we provide an introduction to holographic models of memory and the fan effect. Next, we detail Anderson and Reder's (1999) ACT-R model of the fan effect. We then describe HDM and the ACT-R HDM model of the fan effect. We contribute a novel analysis of how holographic models produce the fan effect and relate to Anderson and Reder's model. Finally, we outline future work.

Holographic Models of Memory

First proposed by Longuet-Higgins (1968) and Gabor (1969), a holographic memory is a type of computational associative memory based on the mathematics of holography. Holographic memory has been of interest to cognitive psychologists because of the following:

(i) Associative memories are content-addressable, allowing for memory retrieval without search.

(ii) Holographic memories can compactly store complicated and recursive relations between ideas.

(iii) Holographic memories have "lossy" storage, which is useful for modelling human forgetting.

Cognitive models based on holographic memory can explain and predict a variety of human memory phenomena, such as the serial position curve in free recall (Franklin & Mewhort, 2015). Holographic memory has also been used to model analogical reasoning (Plate, 2000; Eliasmith & Thagard, 2001) and how humans perform simple problem-solving tasks such as playing rocks, paper, scissors (DSHM; Rutledge-Taylor et al., 2014) or solving Raven's progressive matrices (Eliasmith, 2013). Knowledge in SPAUN, the world's largest functional brain model (Eliasmith, 2013), is represented using holographic memory.

ACT-R DM is not designed for modelling tasks that involve large databases, such as language comprehension. Conversely, BEAGLE (Jones & Mewhort, 2007) and

[1] A Python ACT-R distribution with HDM included can be downloaded from <https://github.com/MatthewAKelly/ccmsuite> and the fan effect model, which requires Python ACT-R and HDM, can be downloaded from <https://github.com/MatthewAKelly/faneffect>. A guide to using Python ACT-R can be found at <https://sites.google.com/site/pythonactr/>.

DSHM (Rutledge-Taylor, Vellino, & West, 2008) are holographic models that have been used, respectively, to infer word meanings from a corpus and to infer patterns of movie preferences from a database of user movie scores.

Holographic memory models have also been previously used to model the fan effect. Specifically, Dynamically Structured Holographic Memory (DSHM; Rutledge-Taylor et al., 2014; Rutledge-Taylor, Pyke, West, & Lang, 2010) has been used to model two versions of the fan effect task.

Though HDM is based on DSHM, the HDM module for ACT-R differs sufficiently from DSHM that it is worth demonstrating that HDM can, in fact, model the fan effect task. The differences between HDM and DSHM stem from HDM's integration into ACT-R. As a module for ACT-R, HDM makes commitments as to the cognitive structure that the memory system is situated in. To interface with ACT-R, HDM commits to a particular way of encoding information and to a particular way of calculating reaction times that are distinct from the DSHM model.

Fan Effect

The *fan effect task* (Anderson, 1974) is a recognition memory task. During the study phase of the task, participants memorize a set of sentences that vary on some number of dimensions. In the original fan effect task (Anderson, 1974), each sentence is of the form "the person is in the location" where the person and location vary from sentence to sentence (e.g., "the hippy is in the park").

Once the participants have the sentences memorized, they are given a recognition task. In the recognition task, some sentences are from the study set (*targets*), and some sentences are novel combinations of the people and locations from the study set (*foils*). Participants are instructed to identify as quickly as possible which combinations of person and location were in the study set and which were not.

The *fan* of a concept is the number of different sentences in the study set that contained that concept. For example, if "the hippy is in the park" is the only sentence in the study set that mentions the hippy, then *hippy* has a fan of one. If participants learn that there are four people in the park during the study phase, then *park* has a fan of four.

The *fan effect* refers to the finding that participants are slower to recognize or reject sentences that contain concepts that have a higher *fan.* The more people in the park, the slower participants are to decide if the phrase "hippy is in the park" was in the study set. Likewise, if participants learn that the hippy is in several different locations, they are slower to decide if the hippy was in a particular location.

The fan effect illustrates a fundamental principle of human memory: the availability of a piece of information in memory with respect to a cue is a function of the probability of that piece of information conditional on the cue. If the participants learn four facts about the park, then given the cue *park*, each of those facts have only one chance in four of being the relevant fact to retrieve. The retrieval time from memory will reflect that one in four chance. Conversely, if the participants know only one fact about the park, given the cue *park*, retrieval time will be rapid, reflecting the 100% chance that the fact will be relevant.

ACT-R's Declarative Memory (DM)

In ACT-R, knowledge is represented in Declarative Memory (DM) as lists of *slot:value* pairs called *chunks*. Each slot is a task-relevant dimension of the stimulus, such as "colour" or "location". For example, a red square could be described by the chunk "colour:red shape:square". In the fan effect task, each sentence is represented by a chunk, e.g., "person:hippy place:park". In Python ACT-R, chunks can also be ordered lists of values without slots, "red square" or "hippy park". When the slots are omitted from a chunk, the order of the values in the chunk is used as the organizing principle.

Each chunk in DM has an activation. 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. Given a cue that describes the current situation, ACT-R retrieves the chunk in DM with the highest activation. Activation is a sum of a base level activation and a measure of the similarity between the chunk and the cue. Base level activation is a measure of how frequently and how recently the chunk has been used. For a chunk i, the activation of that chunk, A_i, is

$$A_i = B_i + \sum_{j=1}^{n} W_j \, S_{ji} \tag{1}$$

where B_i is the baseline activation of the chunk, n is the number of slot-value pairs in the cue, W_j is the attention paid to slot-value pair j of the cue, and each S_{ji} is an association strength: a measure of the probability that chunk i is relevant given that the cue contains slot-value pair j.

DM can be understood by analogy to a hydraulic system. Activation is like water and connections between cues and chunks are like pipes. Activation spreads from the cue to the chunks in DM. Chunks with stronger associations to the cue receive more activation. The chunk that receives the most activation is selected and retrieved from memory. The time, T, to retrieve a chunk, i, is a function of the chunk's activation, A_i, and two fitting parameters I and F,

$$T = I + Fe^{-A_i} \tag{2}$$

The higher the activation, the shorter the retrieval time.

Although ACT-R has a mechanism for learning the association strengths, this has not been tested with the fan effect. Instead, each S_{ji} for chunk i and slot-value j is

$$S_{ji} = \mathrm{S} + \ln(\, \mathrm{P}(i|j) \,)$$

where S is a fitting parameter and $\mathrm{P}(i|j)$ is the probability that chunk i will be useful given the presence of the concept j in the cue. In the fan effect, the chunk i might be "hippy park" and j might be *park*. If there are four people in the park then *park* has a fan of four. The probability that "hippy park" is the correct chunk given *park* is then 1/4 or, more generally, $1/f$ where f is the fan.

In the fan effect task, the experimental design is supposed to control for frequency and recency effects, and so the ACT-R model of the fan effect assumes all chunks have the same baseline activation, B_i, and thus baseline activation can be removed from the equation.

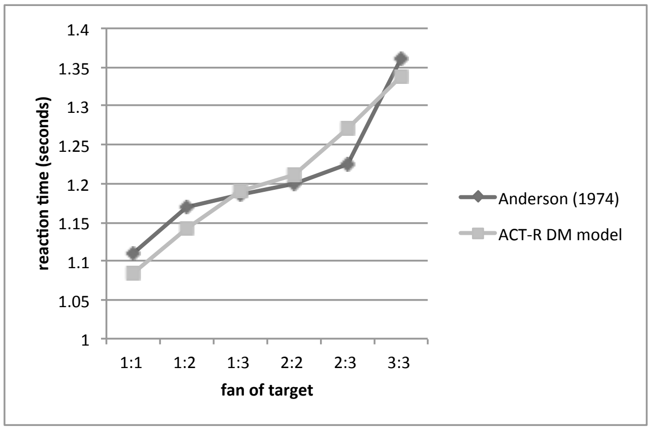

Figure 1: Real versus simulated reaction times for the fan effect from Anderson's (1974) data and Anderson and Reder's (1999) ACT-R DM model.

In Anderson and Reder's (1999) ACT-R model of the fan effect, reaction time for correctly identifying a target as belonging to the study set is calculated in milliseconds with the parameters $S = 1.45$, $W_j = 1/3$, $I = 845$, and $F = 613$. The target retrieval time for the fan effect model works out to be a function of the product of the person and place's fans:

$$T = 233\ (f_{\text{person}} f_{\text{place}})^{1/3} + 845$$

where f_{person} is the person's fan and f_{place} is the place's fan. This model provides a good fit to participant reaction times to targets in the fan effect task, $r = 0.95$ (see Figure 1).

Holographic Declarative Memory (HDM)

In Anderson and Reder's (1999) ACT-R DM model of the fan effect, it is necessary to set the correct association strength S_{ji} for each concept j and chunk i. However, Holographic Declarative Memory (HDM) produces the fan effect by learning the study set. The association strengths are not explicitly programmed. The studied items are presented to HDM as ACT-R chunks. HDM uses holographic reduced representations (Plate, 1995), a technique for instantiating and manipulating symbolic structure in high-dimensional vectors. To interface with ACT-R, HDM translates chunks into vectors, and vectors into chunks.

In HDM, a *value* is represented by a vector of n numbers randomly sampled from a normal distribution. These randomly generated vectors are referred to as *environment vectors*. Any two vectors chosen at random in a high dimensional space will tend to be approximately orthogonal. In HDM, angles indicate degrees of similarity. Orthogonality indicates complete dissimilarity. If we wanted to represent values with intrinsic similarity (e.g., *brother* and *sister*) we could choose non-orthogonal vectors, but for the purposes of modelling the fan experiment, we assume that the persons and locations are dissimilar.

In HDM, a *slot* is represented by a random permutation: a randomly selected reordering of a vector's elements. A *slot-value* pair is represented by reordering the elements of the *value* vector by the *slot* permutation.

Information storage in HDM is based on BEAGLE (Jones & Mewhort, 2007) and DSHM (Rutledge-Taylor et al., 2014). HDM is a *concept-based* memory system. Rather than storing chunks per se, HDM stores relationships between concepts, i.e., the *values* from an ACT-R chunk. Each concept is represented by two vectors: an environment vector $\mathbf{e}_{\text{concept}}$ that represents the percept of that concept, and a memory vector $\mathbf{m}_{\text{concept}}$ that stores the relationship between that concept and other concepts.

As information storage in HDM differs from DM, so too does the process of retrieval. To recall from DM, DM is given a retrieval cue that is a description of a chunk and DM retrieves a chunk that matches that description. Conversely, in HDM, a cue is a question, represented by a vector, and HDM retrieves the concept that best answers that question.

A memory vector for a concept, $\mathbf{m}_{\text{concept}}$, stores a list of questions to which HDM knows, from experience, that the concept is a candidate answer. When cued, that is, posed a question, HDM selects the memory vector with the greatest similarity to the cue and gives as answer the concept represented by that memory vector.

In Python ACT-R, a cue may contain the value question mark, '?', to indicate a 'wildcard', that is, an unknown value. DM can retrieve more than one unknown value at a time because it is retrieving a complete chunk. Whereas in HDM, each unknown value requires a separate retrieval because HDM retrieves a value rather than a chunk (though we are open to the possibility that these retrievals could be performed in parallel). A chunk used as a cue for recall in HDM must contain exactly one '?' to indicate the concept (i.e., value) that HDM should retrieve.

Memory Encoding and Recall with Slots

In HDM, there are two ways to structure knowledge corresponding to the two kinds of chunk in Python ACT-R: lists of *values* or unordered lists of *slot-value* pairs. We first discuss storing unordered slots-value pairs in HDM.

To store in HDM the chunk "colour:red shape:square size:large", we update the memory vector for each concept in the chunk: $\mathbf{m}_{\text{red}}$, $\mathbf{m}_{\text{square}}$, and $\mathbf{m}_{\text{large}}$. To update the memory vector for red, $\mathbf{m}_{\text{red}}$, we need to construct a vector representing the relationship between the concept red and all other concepts in the chunk and then add that vector to $\mathbf{m}_{\text{red}}$. In other words, we need to describe the set of questions for which *red* is an appropriate answer given "colour:red shape:square size:large" and add those questions to $\mathbf{m}_{\text{red}}$. Those questions are "What colour is it?", "What colour is the large thing?", "What colour is the square?" and "What colour is the large square?".

The question "What colour is it?" can be represented by the chunk "colour:?", "What colour is the large thing?" by the chunk "colour:? size:large", and "What colour is the large square?" by "colour:? size:large shape:square".

When the cue is translated into a vector, the '?' becomes the *placeholder* (Jones & Mewhort, 2007). The *placeholder*, denoted by $\mathbf{\Phi}$, is a vector used to encode all associations and thus serves as a universal retrieval cue. The placeholder is randomly generated like an environment vector. Using the placeholder, the cue "colour:?" is translated into the vector

$\mathbf{q}_{colour:?} = (\mathbf{P}_{colour}\ \mathbf{\Phi})$, where $\mathbf{P}_{colour}$ is the permutation representing the slot *colour*.

In holographic reduced representations (Plate, 1995), there are two ways of combining a pair of vectors to create a new vector: + vector addition and * circular convolution. An association between concepts is represented by convolving together the environment vectors representing those concepts. Addition is used to superimpose vectors representing separate information into a single vector.

The vector that represents the question "colour:? size:large" is $(\mathbf{P}_{colour}\ \mathbf{\Phi})*(\mathbf{P}_{size}\ \mathbf{e}_{large})$, i.e., the placeholder permuted by *colour* and convolved with *large* permuted by *size*. This vector will only match the memory vectors of concepts that are colours associated with large objects.

By default, HDM allows for partial matching of cues to concepts in memory. To do so, HDM translates each cue into a set of questions: the question explicitly specified by the cue and all less specific variants of that question. The cue "colour:? size:large shape:square" is translated into a sum of vectors representing "colour:? size:large shape:square" and also "colour:? size:large", "colour:? shape:square", and "colour:?", calculated as follows:

$$
\begin{aligned}
\mathbf{q}_{colour:?\ size:large\ shape:square} = \\
&(\mathbf{P}_{colour}\ \mathbf{\Phi}) \\
&+ (\mathbf{P}_{colour}\ \mathbf{\Phi})*(\mathbf{P}_{size}\ \mathbf{e}_{large}) \\
&+ (\mathbf{P}_{colour}\ \mathbf{\Phi})*(\mathbf{P}_{shape}\ \mathbf{e}_{square}) \\
&+ (\mathbf{P}_{colour}\ \mathbf{\Phi})*(\mathbf{P}_{shape}\ \mathbf{e}_{square})*(\mathbf{P}_{size}\ \mathbf{e}_{large})
\end{aligned}
$$

Cues are used both to retrieve from memory and to add new knowledge to memory. When the chunk "colour:red size:large shape:square" is added to memory, HDM updates $\mathbf{m}_{red}$, $\mathbf{m}_{square}$, and $\mathbf{m}_{large}$ as follows:

$$
\begin{aligned}
\Delta\mathbf{m}_{red} &= \mathbf{q}_{colour:?\ size:large\ shape:square} \\
\Delta\mathbf{m}_{square} &= \mathbf{q}_{colour:red\ size:large\ shape:?} \\
\Delta\mathbf{m}_{large} &= \mathbf{q}_{colour:red\ size:?\ shape:square}
\end{aligned}
$$

Given a retrieval cue, HDM selects the memory vector with the greatest similarity to the cue's vector and the cue's chunk is returned to ACT-R with the '?' substituted for the concept that the memory vector represents.

Similarity is measured by the cosine of the angle between vectors, which can be calculated as:

$$\text{cosine}(\mathbf{q}, \mathbf{m}) = (\mathbf{q} \bullet \mathbf{m}) / (\ (\mathbf{q} \bullet \mathbf{m})^{0.5}\ (\mathbf{q} \bullet \mathbf{m})^{0.5}\ )$$

where $\mathbf{q}$ is a cue vector, $\mathbf{m}$ is a memory vector, and • is the dot product. The cosine is the dot product normalized by the magnitudes of the vectors. A cosine of 1 means the vectors are identical and 0 means they are completely dissimilar. HDM uses DM's retrieval time equation (Equation 2), but calculates activation as similarity measured by the cosine.

Vectors without Slots

Without slots, relationships are indicated by the order of the values in the chunk. Convolution is commutative, $\mathbf{a} * \mathbf{b} = \mathbf{b} * \mathbf{a}$, so the order is not preserved. To preserve the order we use $\mathbf{P}_{before}$, a random permutation indicating that a vector occurred before another a vector. To add the chunk "large red square" to memory, we would update $\mathbf{m}_{red}$, $\mathbf{m}_{square}$, and $\mathbf{m}_{large}$. We would update $\mathbf{m}_{red}$ as follows:

$$
\begin{aligned}
\Delta\mathbf{m}_{red} = &(\mathbf{P}_{before}\ \mathbf{e}_{large})*\mathbf{\Phi} \\
&+ (\mathbf{P}_{before}\ \mathbf{\Phi})*\mathbf{e}_{square} \\
&+ (\mathbf{P}_{before}\ ((\mathbf{P}_{before}\ \mathbf{e}_{large})*\mathbf{\Phi})*\mathbf{e}_{square}
\end{aligned}
$$

which adds the questions "large ?", "? square" and "large ? square" to the memory of the concept of red.

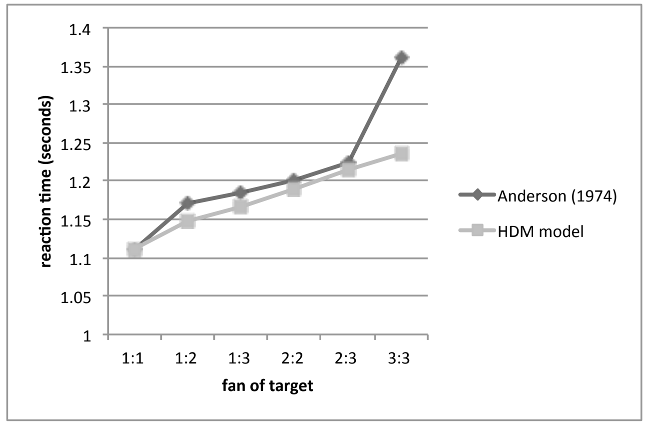

Figure 2: Real versus simulated reaction times for the fan effect from Anderson's (1974) data and the HDM model.

Recognition with Holographic Declarative Memory

In the DM model of the fan effect, the activation of a chunk is calculated as a weighted sum of the association strengths of the chunk's constituent concepts. In HDM, association strengths are measured by vector cosine, so we can calculate that activation in HDM as a weighted sum of cosines.

When determining whether HDM recognizes a cue, the cue chunk must contain no unspecified values '?'. For each value in the cue, HDM creates a new cue with that value substituted for '?', performing one retrieval for each value in the original cue. Activation is calculated as the mean of the cosines between each of these cues and the memory vector of the concept that was substituted out to create the cue. This method for calculating activations in the fan effect has been used before by DSHM (Rutledge-Taylor et al., 2014; Rutledge-Taylor, Pyke, West, & Lang, 2010).

In the fan effect task, for the cue "hippy park", HDM does two retrievals, "hippy ?" and "? park" with the vectors $\mathbf{q}_{hippy?} = (\mathbf{P}_{before}\ \mathbf{e}_{hippy})*\mathbf{\Phi}$ and $\mathbf{q}_{?park} = (\mathbf{P}_{before}\ \mathbf{\Phi})*\mathbf{e}_{park}$. Activation A is calculated as:

$$A = 0.5\ \text{cosine}(\mathbf{q}_{hippy?}\ ,\ \mathbf{m}_{park}) + 0.5\ \text{cosine}(\mathbf{q}_{?park}\ ,\ \mathbf{m}_{hippy})$$

The HDM Model of the Fan Effect

We ran the HDM model of the fan effect task 20 times, simulating 20 virtual participants, and averaged across runs. Because each run uses a different set of random vectors, the cosines and reaction times vary randomly with each run. Anderson's (1974) experiment had 18 participants. The model fits the human participant data reported by Anderson (1974) with a correlation of r = 0.91 (see Figure 2). The fit was obtained using the exact same values for the fitting parameters as Anderson and Reder's (1999) ACT-R fan effect model. The only change was to compute activation as a mean of cosines, as described in the previous section.

Anderson and Reder's (1999) model and the HDM model are strongly correlated, $r = 0.99$. While there are slight differences in the predictions made by the two models, both the DM and HDM models are within the range of human variability for performance on this task. These results show that HDM replicates DM's ability to model the fan effect, but HDM does so in a radically different way: by measuring the cosine between vectors in a high-dimensional space.

Why does the cosine model the fan effect so well? The cosine acts as an estimate of the conditional probabilities that the Anderson and Reder's (1999) fan effect model uses to compute association strengths. The memory vector for a concept keeps a fuzzy count of the number of times that concept has co-occurred with each other concept. Taking the dot product of the cue with a memory vector gives you an estimate of the frequency with which that cue has been added to that memory vector, that is, the number of times the relationships described in that cue have occurred with that concept. The cosine is a dot product normalized by the magnitudes of the vector, which in this case, is a frequency normalized by the total number of instances, that is to say, the cosine is roughly the probability.

We can imagine all vectors in HDM as points on a n-dimensional hypersphere. For the HDM fan effect model, we used 256 dimensions, but for the sake of visualization, imagine a 3-dimensional sphere.

Let us first consider a fan of one. Suppose the model has learned only one fact about the hippy, namely, the "hippy is in the park". After learning this fact, the memory vector for hippy will be $\mathbf{m}_{\text{hippy}} = (\mathbf{P}_{\text{before}}\ \mathbf{\Phi})*\mathbf{e}_{\text{park}}$. The model is later given the cue "the hippy is in the park" during the recognition phase. To test for recognition, we take the cosine of $\mathbf{m}_{\text{hippy}}$ with the cue $\mathbf{q}_{\text{?park}} = (\mathbf{P}_{\text{before}}\ \mathbf{\Phi})*\mathbf{e}_{\text{park}}$. As $\mathbf{m}_{\text{hippy}} = \mathbf{q}_{\text{?park}}$ the angle between the cue and the memory vector is zero, the distance between them on the surface of the hypersphere is zero, and the cosine is 1.00.

Let us consider a fan of two. If the model knows "hippy is in the park" and "hippy is in the bank", then $\mathbf{m}_{\text{hippy}}$ is the sum of the park cue $\mathbf{q}_{\text{?park}}$ and the bank cue $\mathbf{q}_{\text{?bank}}$,

$$\mathbf{m}_{\text{hippy}} = (\mathbf{P}_{\text{before}}\ \mathbf{\Phi})*\mathbf{e}_{\text{park}} + (\mathbf{P}_{\text{before}}\ \mathbf{\Phi})*\mathbf{e}_{\text{bank}}$$

In high dimensional spaces, randomly chosen vectors are approximately orthogonal to each other. Let us assume that the cues $\mathbf{q}_{\text{?bank}}$ and $\mathbf{q}_{\text{?park}}$ are perfectly orthogonal. As illustrated in the left half of Figure 3, on the surface of the hypersphere, $\mathbf{m}_{\text{hippy}}$ will be halfway between the two cues at a 45˚ angle. The cosine is 0.71.

Let us consider a fan of three. If the model knows that the hippy is in the bank, park, and store, $\mathbf{m}_{\text{hippy}}$ will be at an equidistant point on the hypersphere between the cues for bank, park, and store. In the fan of three, $\mathbf{m}_{\text{hippy}}$ is further away from all the cues than in a fan of two. The angle between $\mathbf{m}_{\text{hippy}}$ and any cue is 55˚ and the cosine is 0.58.

Where f is the fan, the cosine between a cue and a memory vector is $f^{-1/2}$ if the vectors are perfectly orthogonal, or

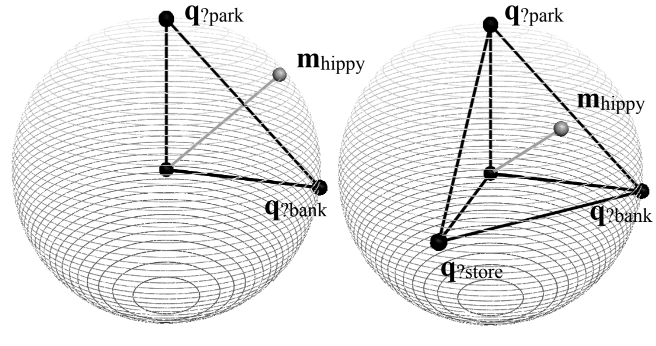

Figure 3: $\mathbf{m}_{\text{hippy}}$ with a fan of 2 (left) or 3 (right).

approximates $f^{-1/2}$ for the random vectors used by HDM. Thus HDM predicts that as the fan increases, the cosine decreases, but by diminishing amounts with each increase in fan. As the fan approaches infinity, the cosine approaches zero. HDM makes the intuitive prediction that increases in the fan has a steadily diminishing effect on reaction time, such that knowing 100 facts about the hippy is not appreciably different from knowing 101.

The cosine in HDM approximates the square-root of the probability only when the events are equiprobable. For n events with frequencies v_1 to v_n, the cosine of event i is

$$\text{cosine} = \frac{v_i}{\sqrt{v_1^2 + \ldots + v_i^2 + \ldots v_n^2}} \quad (3)$$

When given events of unequal probabilities, HDM will behave as if the most frequent events are disproportionately likely and the least frequent events are disproportionately unlikely. This is a testable and possibly erroneous prediction of HDM. The quantum probability model of human judgements (Busemeyer, Pothos, Franco, & Trueblood, 2011) also uses vector algebra to calculate probabilities, but uses the square-roots of the frequencies, then squares the cosine, such that Equation 3 is equal to classical probability. Using the square-roots of the frequencies is not possible for HDM as it would require HDM to know a priori how frequently each event will occur.

Future Work and Applications of HDM

We have presented in this paper an HDM model of the fan effect and compared it to Anderson and Reder's (1999) DM model of the fan effect. However, we have only discussed fitting to the reaction time of targets, sentences presented at the recognition phase that occurred in the study set. Anderson and Reder's (1999) model for foils, sentences that were not in the study set, fails on a variant of the fan effect task (West, Pyke, Rutledge-Taylor, & Lang, 2010). As the foil is difficult to model, we leave developing an HDM model of the foil for future research.

At present, HDM does not model recency effects, that is, more recent information is not recalled better than less recent information. However, other holographic models in the literature (e.g., Franklin & Mewhort, 2015; Murdock 1993) can account for recency effects, so such a mechanism could be incorporated into the model.

At present, interfacing with ACT-R chunks imposes an information bottleneck on HDM. Detailed sensory information cannot be feasibly stored in ACT-R chunks, but can be stored in holographic vectors (Kelly, Blostein, & Mewhort, 2013). Reimplementing the entirety of ACT-R as a holographic system would improve ACT-R's ability to interface with real world environments and to match situations to procedures. Some of that work has already been done: A holographic model similar to ACT-R's procedural memory system already exists as the basal ganglia model of the SPAUN brain model (Eliasmith, 2013; Stewart, Bekolay, & Eliasmith, 2012). However, a holographic procedural memory consistent with HDM and ACT-R would necessarily differ from SPAUN's to meet the demands of integration with a different architecture.

HDM is a powerful tool for cognitive modellers because it inherits the abilities of holographic models such as BEAGLE (Jones & Mewhort, 2007) and DSHM (Rutledge-Taylor, Vellino, & West, 2008) to store large quantities of data in memory and use it to make intelligent predictions in knowledge-heavy tasks. In Rutledge-Taylor et al. (2014) we show that DSHM can be used to model a difficult but small-scale decision-making task. HDM could be applied to a large-scale, knowledge-driven decision-making task.

Conclusion

We present a new module for ACT-R, Holographic Declarative Memory (HDM). We substitute HDM for DM in the ACT-R model of the fan effect and find that without changing any parameters HDM provides a good fit to the fan effect. We present an analysis that allows us to specify the mathematical relationship between the DM and HDM models of the fan effect.

HDM, by virtue of being a holographic model, has a number of capabilities for which DM is less suited, such as analogical or case-based reasoning, learning associations between concepts without having association strengths set by the modeller, and performing tasks that require large amounts of knowledge. We hope that by integrating a holographic memory model into ACT-R, we can bring the capabilities of vector space modelling into the ACT-R research community and enhance the capability of the ACT-R cognitive architecture to model human cognition.

Acknowledgments

This research is supported by an Ontario Graduate Scholarship awarded to the first author and a grant from the National Science and Engineering Research Council of Canada to the third author.

References

Anderson, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology, 6,* 451-474.

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*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General, 128,* 186-197.

Busemeyer, J. R., Pothos, E. M., Franco, R., & Trueblood, J. (2011). A quantum theoretical explanation for probability judgement errors. *Psychological Review, 118,* 193-218.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford University Press.

Eliasmith, C., & Thagard, P. (2001). Integrating structure and meaning: a distributed model of analogical mapping. *Cognitive Science, 25*, 245-286.

Franklin, D. R. J., & Mewhort, D. J. K. (2015). Memory as a hologram: An analysis of learning and recall. *Canadian Journal of Experimental Psychology, 69*, 115-135.

Gabor, D. (1969). Associative holographic memories. *IBM Journal of Research and Development, 13*, 156–159.

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.

Murdock, B. B. (1993). TODAM2: a model for the storage and retrieval of item, associative and serial-order information. *Psychological Review, 100*, 183–203.

Longuet-Higgins, H. C. (1968). Holographic Model of Temporal Recall. *Nature, 217*, 104.doi: 10.1038/217104a0

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

Plate, T. A. (2000). Analogy retrieval and processing with distributed vector representations. Expert Systems: The International Journal of Knowledge Engineering and Neural Networks, 17, 29-40.

Rutledge-Taylor, M. F., Pyke, A. A., West, R. L., Lang, H. (2010). Modeling a three term fan effect. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 211-216). Philadelphia, PA: Drexel University.

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.

Rutledge-Taylor, M. F., Vellino, A., & West, R. L. (2008). A holographic associative memory recommender system. In *Proceedings of the 3rd International Conference on Digital Information Management* (pp. 87-92). London, UK.

Stewart, T. C., Bekolay, T., and Eliasmith, C. (2012) Learning to select actions with spiking neurons in the basal ganglia. *Frontiers in Neuroscience, 6:2*, 1-14.

West, R. L., Pyke, A. A., Rutledge-Taylor, M. F., & Lang, H. (2010). Interference and ACT-R: New evidence from the fan effect. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 277-281). Philadelphia, PA: Drexel University.