A Comparison of Frequency Effects in Two Attitude Retrieval Models

Mark Orr[1] (morr@ihmc.org), Christian Lebiere[2] (cl@cmu.edu)
Don Morrison[2] (dfm2@cmu.edu), Peter Pirolli[1] (ppirolli@ihmc.org)

[1]Institute for Human and Machine Cognition
Pensacola, FL 32502 USA

[2]Department of Psychology, Carnegie Mellon Univ.
Pittsburgh, PA 15223 USA

Abstract

The psychological literature has put forth several auto-associative memory models of attitude formation and change. The status of frequency effects in such models is not well understood. We compare frequency effects in auto-associative memory models of attitudes to the well-established frequency effects found in the ACT-R cognitive architecture. We found striking differences between the model classes, but only under some conditions. We discuss future directions that might stem from this provisional work.

Keywords: attitudes, cognitive modeling, neural networks, memory, dynamical systems

The Problem

Attitude learning is divided into two camps. In one, we have memory processes as a central theoretical component for understanding how attitudes are formed and retrieved. These typically concern memory for the valence towards an attitude object. Although typically not formalized, a running debate in the social psychological literature stems from differentiating or not between simple associative learning or propositional learning in attitudes. This literature is rich in terms of evidence on learning (see Corneille & Stahl, 2019, for examples).

In the other camp, what we will call schema-like memory models, the primary interest is in attitudinal structure (Eagly & Chaiken, 1993). Recent work in this area uses auto-associative memory models to represent not only structure but also as models of attitudinal memory retrieval (e.g., Dalege et al., 2016, 2018). In this work, sets of beliefs are transformed from survey data into a network of associations (e.g., correlations) and modeled using Hopfield-like or Ising-like models. Learning is not well studied in such models. In its place are notions of persuasion: under what conditions will a person stray from their typical attitude retrieval pattern.

In short, little overlap exists between these two literatures. We attempt a kind of reconciliation between the two by studying attitude learning in the auto-associative memory case. Much is know about learning in auto-associative memory systems (e.g., Hopfield, 1982; Hertz et al., 1991). So, we thought it would be useful to directly compare learning in the auto-associative case to learning in an empirically-grounded cognitive architecture. For our comparison, we chose the currently prominent Causal Attitude Network (CAN) model (from social psychology, Dalege et al. (2016, 2018)) to the ACT-R cognitive architecture (Anderson et al., 2004). Our comparison method, thus, affords the following features: (i) it will ground the findings in human memory systems via ACT-R and (ii) it addresses learning in the structural approach to attitudes.

Design

Across two studies, we compared directly an ACT-R declarative memory model to the CAN attitude model. By directly, we mean that the input data, the model task and the analysis methods were identical. There were some differences in computed measures, but the semantics between them were close.

The Causal Attitude Network Model

The CAN model (Dalege et al., 2016; Dalege & van der Maas, 2020; Dalege et al., 2018) was motivated by the need to provide a dynamic attitude memory retrieval system that exhibits sensitivity to cues in the social environment. Virtually all theoretical work on the CAN model uses fixed, predetermined weights for its network (see below for the formal specification of the system). The CAN model literature references Hebbian learning as a potential candidate for learning attitudes, yet there have been no studies to date that implement learning. The heart of this theoretical work focuses on dynamical retrieval methods that are derived from Ising-like or discrete Hopfield models. The technical details of how a typical CAN model is implemented are as follows–we start with key definitions:

- There is a graph $G = G(V,E)$ consisting of a collection of beliefs (the set of n vertices V) and relations between them (the set of weighted edges E).
- The state of vertex $i \in V$ is $x_i \in K_i$ where K_i is the state set for that vertex.
- For all i we have $K_i \in \{0,1\}$
- The system state is $x = (x_1, x_2, \ldots, x_n)$.
- The system global energy H is defined using all $i \in V$ by $H(x) = -\sum_{i \in G} \tau_i x_i - \sum_{j \in N_G(i)} w_{ij} x_i x_j$ where $N_G(i) \subset V$ is the set of neighbors of i in G, *not* including i, w_{ij} is the weight of the edge $\{j,i\}$ and τ_i is the baseline parameter for vertex i. Assume that $w_{ij} = w_{ji}$.

- For $i \in V$ let $\sigma_i\colon \prod_{i=1}^{n} K_i \longrightarrow \mathbb{R}$ be the function defined by $\sigma_i(x) = H(x)^c - H(x)^o$ where c and o are the current and opposite state of vertex i.

- For each vertex i we define its vertex function as $\phi_i(x) = 1/(1+e^{-\sigma_i(x)/t})$ where t is the temperature of the system; this defines the probability that at any point in time a vertex i will flip to its opposite state: $P(c \longrightarrow o) = \phi_i(x)$.

A typical instance of CAN is a discrete-time, asynchronous simulation. For each time step: (i) select a vertex i, (ii) compute $P(c \longrightarrow o) = \phi_i(x)$ and (iii) use $P(c \longrightarrow o)$ directly to decide if vertex i will change its state. Another common implementation is to draw n samples of the system state x from the Gibbs probability distribution. This is computed as: (i) compute the Gibbs probability distribution of all system states x_i such that each is $P(x = x_i) = e^{-H(x_i)}/Z$ where $H(x_i) = -\tau_i x_i - \sum_{j \in N_G(i)} w_{ij} x_i x_j$ and $Z = \sum_X e^{-H(x)}$, (ii) sample from this distribution n times. In our CAN simulations below, we leverage the latter.

ACT-R Declarative Memory

For this article, we develop a comparison to the CAN model using the declarative memory module of the ACT-R cognitive architecture implemented in the PyACTUp Python package[1].

Declarative memory is a module in the ACT-R cognitive architecture comprised of discrete data objects called *chunks*. Each chunk contains a number l of slots which contain attribute-value pairs. The attribute is the slot name and the value is the slot content. Access to this symbolic content is controlled by a subsymbolic quantity called activation, which reflects the characteristics of the knowledge including its history and semantics. The activation calculus determining declarative memory access works as follows:

- The activation A of a chunk is defined as: $A_i = B_i + \varepsilon_i + P_i + S_i$ where B_i is the base level activation, ε_i is stochastic noise, P_i is the partial matching correction, and S_i is the spreading activation . The latter term was not used in the work presented here.

- The base level activation B_i is defined as: $B_i = ln(\sum_i t_{ij}^{-d})$ where t is the time lag since the jth reference to chunk i and d is the time decay parameter, typically set at 0.5.

- Retrieval from memory is computed by selecting the chunk with the highest activation value, after noise has been added. Analytically, the probability P_i of retrieving chunk i can be characterized by the Boltzmann (softmax) distribution as $P_i = e^{A_i/t} / \sum_j e^{A_j/t}$ where the sum is over all chunks j matching the retrieval request and the temperature t is a function of the noise parameter. This is equivalent to viewing the activation of a chunk as an estimate of the log odds of retrieval need (Anderson (1990)).

- The latency T_i of a chunk retrieval is inversely proportional to its activation as: $T_i = Fe^{-A_i}$ when F is a time scaling parameter.

Although attitudes have been modeled using ACT-R in prior work (Orr et al., 2021; Pirolli, 2016a,b; Pirolli et al., 2020), there exists no direct comparison to prominent models in the social psychology literature.

Data

We generated synthetic data for both studies in this article using two bit vectors as the basis for the synthetic data. The intent is for those vectors to represent two distinct attitudes competing in belief space. To generate the basis bit vectors, we used the following procedure: Take any random bit vector of length 16 with exactly eight bits with a state of 1 as the first pattern ζ^1. Then, generate another pattern ζ^2 from ζ^1 by flipping four of the 1 bits and four of the 0 bits. This procedure results in the two patterns ζ^1 and ζ^2 that are exactly the expected Hamming distance among all possible vectors in the configuration space of size 2^{16}. For ease of analysis, we fixed ζ^1 to 1111111100000000 and generated ζ^2 as 1111000011110000; these were our two basis bit vectors.

We constructed five sets of data, all using the same procedure. The basic unit of data was called an example, a single 16-bit vector. We first defined five frequency ratios, each mapping to one of the five sets of data: 50:50, 60:40, 70:30, 80:20, 90:10. The first term of each ratio referenced the number of examples of ζ^1 in the data set; the second term did the same for ζ^2. Each of the five sets of data also contained one example from the full configuration space of 2^{16} (that is 65,536 distinct examples define the configuration space). Thus, each of the five data sets contained a total of 65,636 examples, 100 of which were some ratio of ζ^1 and ζ^2.

The CAN model assumes that each node in a Hopfield network captures the endorsement or not of a belief that references an attitude object (e.g., 'has claws' is a belief about cats that is either endorsed or not). We use the same abstraction in our simulations and will call each bit in the bit vector an attitudinal belief.

Simulations

The two models (CAN and ACT-R) learned the data via a single pass through all examples in a data set (for both Study 1 and 2). The notion, in attitude research, is that each example is an abstraction of a social exposure to a set of beliefs (e.g., from an acquaintance or from mass media). We will call this the learning phase, which was identical in all conditions across Studies 1 and 2 (except for the distinct frequency distributions of each condition). We ran two separate studies.

Study 1: Frequency Effects in Free Recall. The objectives of Study 1 were to understand how each of the model types (CAN and ACT-R) represents differences in frequency of inputs and how this affects retrieval under free-recall. For each model type there were five conditions, one for each of the five data sets, which determined the data that the model

[1] https://github.com/dfmorrison/pyactup/

learned. Following learning, each model generated a non-cued retrieval probability for each of the 2^{16} bit vectors in the full configuration space. (See the section *Design* for computation of these probabilities.) Due to stochasticity in retrieval in ACT-R, we computed the set of retrieval probabilities for each model for each condition 30 times, the average of which was reported for the two basis patterns ζ^1 and ζ^2.

Study 2: Frequency Effects in Cued Recall. For Study 2, we used the same method as for Study 1 with one exception, cuing. In Study 2, we ran the full set of simulations used in Study 1 two separate times, each with a different cue. The first time used the more frequent basis pattern ζ^1 as the cue; the second time used the less frequent ζ^2.

The ACT-R Model: We defined all chunks to have one slot for each of the 16 attitudinal beliefs (16 bits in the bit pattern). Each slot had two valid values, 0 and 1. For the learning procedure, the model encoded all examples in its condition. The frequency of each chunk was reflected in the data so chunks were reinforced in proportion to their frequency by separate chunk encodings (i.e., each chunk was reinforced as many times as there were examples in the data). We used the functions `pyactup.learn()` to learn chunks and `pyactup.advance()` to advance time. All chunks were learned prior to advancing time and thus retrieval was not subject to time-dependent decay across chunks. For the simulation procedure we used the `pyactup.retrieve()` function. In Study 1, all retrievals were non-cued. For Study 2, each cue condition was realized by providing the cue of the full pattern of interest, either ζ^1 or ζ^2 e.g., `pyactup.retrieve(`$\{\zeta^1\}$`)`. All parameters of the cognitive architecture were left at their default values, i.e., the decay rate was 0.5, the activation noise was 0.25 and the retrieval threshold was 0.0.

The CAN Model: The Hopfield model was constructed by (i) mapping each of the bits x_i to a network node, (ii) generation of weights w_{ij} using Hebbian learning (Hertz et al., 1991), (iii) assigning a baseline parameter for each x_i as τ_i. Cuing (or not) was controlled by the set of τ_i. In Study 1, all τ_i were set to zero, to reflect no cuing free-recall. In Study 2, cuing was defined as providing the following mapping: $x_i = 1 \longmapsto \tau_i = 1$ if $x_i = 1$ was learned; else $x_i = 0 \longmapsto \tau_i = -1$; the latter condition provided a strong bias for $x_i = 0$.

Results

Study 1: Frequency Effects in Free Recall

The primary result in Study 1, shown in Figure 1, was the comparison between the ACT-R and CAN attitude models under no cuing conditions. Both models responded in a way that captured the frequency ratio between the basis patterns ζ^1 and ζ^2 (note: ζ^1 is more frequent). When the ratio was $50:50$ the probability of recall was nearly equal between the two basis patterns for both models. As the ratio increased, for both models, the separation in probability of recall grew as a function of the size of the ratio between the two basis patterns. Two features distinguish the two models. First, the CAN model had lower probabilities of retrieval overall. Second, also for the CAN model, the probability of retrieval for the less frequent basis pattern ζ^2 was very close to zero for any condition other than the $50:50$ ratio. It is not clear whether these two features of the CAN model indicate a functional difference between it and the ACT-R model.

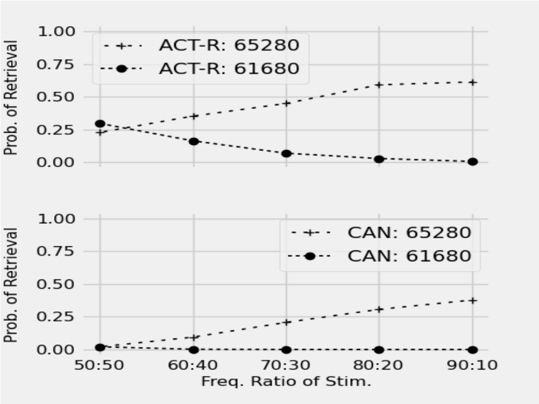

Figure 1: A comparison between the ACT-R (top panel) and CAN (bottom panel) attitude models in the probability of retrieval as a function of each of five conditions of the frequency ratio of the two basis patterns ζ^1 and ζ^2 (the former is 65280; the latter is 61680) in Study 1. No cue was given in this study. Note: ζ^1 is more frequent.

Figure 2 provides some insight into the way the models operate; it shows the results of a single simulation in the condition $50:50$. Both models cleanly separated the two basis patterns ζ^1 and ζ^2 from the other patterns. For the CAN model, the point shown with the highest probability of retrieval captured the two basis patterns (this is occluded because of overlap). For ACT-R, one of the basis patterns was clearly favored, something that was due to the stochastic nature of activation noise in each chunk. Figure 3 shows results for the $80:20$ condition. We see that with a high frequency ratio, both models showed strong separation of the most frequent basis pattern ζ^1. Comparing the two conditions ($50:50$ to $80:20$) surfaces one potentially interesting difference between the two models in terms of their operation. For the CAN model, a larger frequency ratio between the two basis patterns significantly affected the range of the energy surface via reducing the minimum energy of the system (it deepened the attractor); the corresponding effect in terms of activation in ACT-R was much more muted.

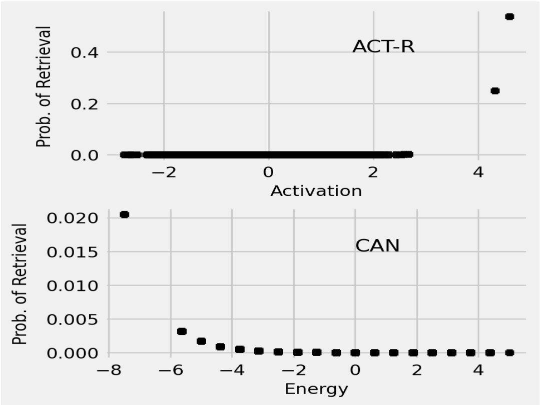

Figure 2: The relation between energy (CAN model) or activation (ACT-R model) (x-axis) and the probability of retrieval (y-axis) for each of the examples in the full configuration space (2^{16} examples). Each panel represents a simulation of the 50 : 50 condition. Note the different scales of the y-axis in each panel.

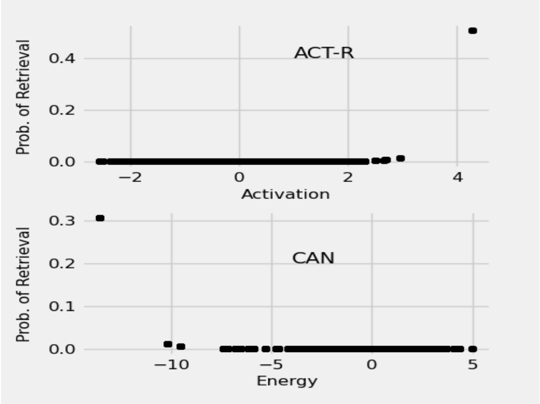

Figure 3: The relation between energy (CAN model) or activation (ACT-R model) (x-axis) and the probability of retrieval (y-axis) for each of the examples in the full configuration space (2^{16} examples). Each panel represents a simulation of the 80 : 20 condition. Note the different scales of the y-axis in each panel.

In summary, the first order comparison between the ACT-R and CAN attitude models showed functional similarity, to a first approximation, in terms of reflecting the frequencies of the learning environment (see Figure 1). Both models were good at separating the two basis patterns and their respective frequencies in terms of probability of retrieval (see Figures 2 and 3). The only notable difference, one for future study, was that the change in the energy space with an increased frequency ratio was much more significant for the CAN model than for the ACT-R model.

Study 2

The results for Study 2 were markedly different from Study 1. Figure 4 shows the set of simulations that cued the more frequent bit pattern ζ^1 (we will call these *Study 2a*). The high-level feature of these data is that both models operated well under cue in the sense that under all frequency ration conditions the cue was likely to be retrieved. This was to be expected because we cued the most frequent bit pattern. Further, the behavior of the ACT-R model was completely dependent on the cue; its behavior was the same for all five frequency ratios. In contrast, the CAN model exhibited a strong frequency effect across the frequency ratio spectrum. We will come back to this latter point shortly.

The set of simulations (*Study 2b*) that cued the less frequent bit pattern ζ^2 are shown in Figure 5. The comparison between ACT-R and CAN showed clear differences. As in Study 2a, the ACT-R model was completely driven by the cue and showed no effect across the frequency ratio conditions. In other words, the partial matching term overwhelmed the base-level activation, partly due to the large size of the fully-specified pattern (16 slots). However, for the CAN model we see see an interaction (of sorts) between the context of the cue and the frequency ratio of what was learned. For lower frequency ratios, the CAN model cued accurately but for higher frequency ratio conditions, the frequency factor drove the probability of retrieval. This, in fact, is the same effect we saw in Study 2a for the CAN model–the probabilities of retrieval decreased as the frequency ratio became smaller, conditions for which the learning context was against, in a relative sense, the more frequent bit pattern ζ^1.

In summary, in both Study 2a and 2b, we see a strong frequency effects for the CAN model and not for the ACT-R model under cuing conditions.

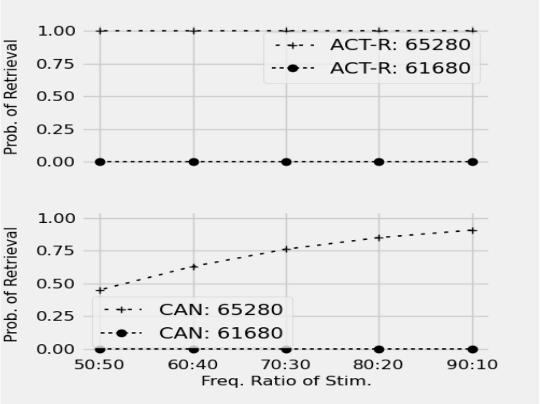

Figure 4: A comparison between the ACT-R (top panel) and CAN (bottom panel) attitude models in the probability of retrieval as a function of each of five conditions of the frequency ratio of the two basis patterns ζ^1 and ζ^2 (the former is 65280; the latter is 61680) in Study 2a. The cue was the more frequent pattern ζ^1.

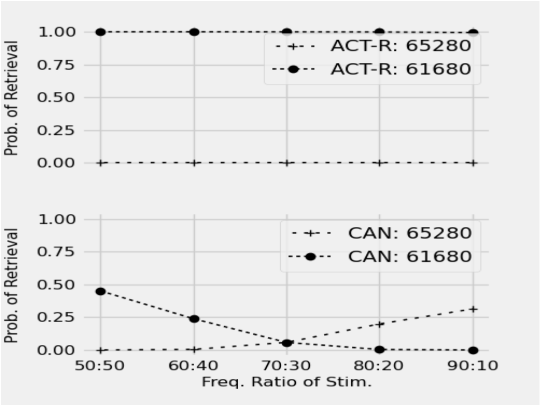

Figure 5: A comparison between the ACT-R (top panel) and CAN (bottom panel) attitude models in the probability of retrieval as a function of each of five conditions of the frequency ratio of the two basis patterns ζ^1 and ζ^2 (the former is 65280; the latter is 61680) in Study 2b. The cue was the less frequent pattern ζ^2.

Conclusions

In conclusion, the differences in the retrieval behavior of the ACT-R and CAN attitude models were greater than the similarities. Cued recall, a more realistic conceptualization of the human attitude problem, showed marked differences between the two models. The ACT-R attitude model was driven by the cue; the CAN model was driven by both cue and learning frequency, sometimes to the extent that the cue was effectively ignored. Although ignoring cues can be adaptive in some tasks, we do not see the value in the context of attitudes unless other social processes or motives were modeled in conjunction.

To what extent does this stand as an indictment of the CAN attitude model? On one hand, the declarative memory model in ACT-R could be seen to serve as a kind of validation comparison: it represents human memory in a way that is not justifiable for the CAN model. In the CAN model's defense, we note that the CAN model was not developed in the context of memory models. The CAN model was an outgrowth of what is called the psychological networks approach, an approach for using graph structure as an alternative measurement approach for psychological survey or clinical data.

We see our work presented here as highly provisional, a useful first step in reconciling learning to the structural approach to attitudes. Future work should study the following issues: (i) the degree of learning in the Hopfield network would impact the results, yet it is not clear to what extent or precisely how, (ii) formal mathematical analysis and comparison of learning and retrieval in both the ACT-R and CAN models, (iii) evaluating the impact of different representations in the ACT-R model, e.g., by representing each belief as a separate chunk, (iv) whether the results generalize to partial cueing of a subset of the full belief set, and (v) the impact of factors such as recency if a real time learning schedule is used.

References

Anderson, J. R. (1990). *The adaptive character of thought*. Lawrence Erlbaum Associates.

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.

Corneille, O., & Stahl, C. (2019). Associative attitude learning: A closer look at evidence and how it relates to attitude models. *Personality and Social Psychology Review*, *23*(2), 161–189.

Dalege, J., Borsboom, D., van Harreveld, F., van den Berg, H., Conner, M., & van der Maas, H. L. (2016). Toward a formalized account of attitudes: The causal attitude network (can) model. *Psychological review*, *123*(1), 2.

Dalege, J., Borsboom, D., van Harreveld, F., & van der Maas, H. L. J. (2018, Oct). The attitudinal entropy (ae) framework as a general theory of individual attitudes. *Psychological Inquiry*, *29*(4), 175–193. doi: 10.1080/1047840X.2018.1537246

Dalege, J., & van der Maas, H. L. J. (2020, Nov). Accurate by being noisy: A formal network model of implicit measures of attitudes. *Social Cognition*, *38*(Supplement), s26–s41. doi: 10.1521/soco.2020.38.supp.s26

Eagly, A., & Chaiken, S. (1993). *The psychology of attitudes*. Orlando, FL: Harcourt Brace Jovanovich.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Addison-Wesley.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554–2558.

Orr, M., Stocco, A., Lebiere, C., & Morrison, D. (2021). Attitudinal polarization on social networks: A cognitive architecture perspective. In *Proceedings of the 19th international conference on cognitive modelling.*

Pirolli, P. (2016a). A computational cognitive model of self-efficacy and daily adherence in mhealth. *Translational behavioral medicine*, *6*(4), 496–508.

Pirolli, P. (2016b). From good intentions to healthy habits: Towards integrated computational models of goal striving and habit formation. In *2016 38th annual international conference of the ieee engineering in medicine and biology society (embc)* (pp. 181–185).

Pirolli, P., Bhatia, A., Mitsopoulos, K., Lebiere, C., & Orr, M. (2020). Cognitive modeling for computational epidemiology. In *2020 international conference on social computing, behavioral-cultural modeling & prediction and behavior representation in modeling and simulation (spb-brims 2020).*