Computationally Efficient Forgetting via Base-Level Activation

Nate Derbinsky (nlderbin@umich.edu)
John E. Laird (laird@umich.edu)
University of Michigan, 2260 Hayward Street
Ann Arbor, MI 48109-2121 USA

Keywords: large-scale cognitive modeling

Introduction

As we apply cognitive models to complex, temporally extended tasks, removing declarative knowledge from memory, or *forgetting*, will become important both to model human behavior, as well as to scale computationally. The *base-level activation* (BLA) model predicts that the availability of specific memories is sensitive to frequency and recency of use. Memory decay based on this model has long been a core commitment of the ACT-R theory (Anderson et al., 2004), as it has been shown to account for a class of memory retrieval errors (Anderson, Reder, & Lebiere, 1996), and has been used in Soar (Laird, 2012) to investigate task-performance effects of forgetting short-term (Chong, 2003) and procedural (Chong, 2004) knowledge. Prior work has addressed many of the computational challenges associated with retrieving a single memory according to the BLA model (Petrov, 2006; Derbinsky, Laird, & Smith, 2010; Derbinsky & Laird, 2011). However, efficiently removing items from memory, while preserving BLA-model fidelity, is a different problem, which we address here. We formally describe the computational problem; present a novel approach to forget according to BLA in large memories; and evaluate using synthetic data.

Problem Formulation

Let memory M be a set of elements, $\{m_1, m_2, \ldots\}$. Let each element m_i be defined as a set of pairs (a_{ij}, k_{ij}), where k_{ij} refers to the number of times element m_i was activated at time a_{ij}. We assume $|m_i| \leq c$: the number of activation events for any element is bounded. These assumptions are consistent with the ACT-R declarative memory when bounding chunk-history size (Petrov, 2006). This is also consistent with the semantic memory in Soar (Laird, 2012).

We assume that activation of an element m at time t is computed according to the BLA model (Anderson et al. 2004), where d is a fixed decay parameter:

$$B(m,t,d) = \ln(\sum_{j=1}^{|m|} k_j \cdot [t - a_j]^{-d})$$

We define an element as *decayed*, with respect to a threshold parameter θ if $B(m,t,d) < \theta$. Given a static element m, we define L as the fewest number of time steps required for the element to decay, relative to time step t:

$$L(m,t,d,\theta) := \inf\{t_d \in \aleph : B(m,t+t_d,d) < \theta\}$$

For example, element $x = \{(3, 1), (5, 2)\}$ was activated once at time step three and twice at time step five. Assuming decay rate 0.5 and threshold -2, x has activation about 0.649 at time step 7 and is not decayed: $L(x,7,0.5,-2) = 489$.

During a model time step t, the following actions can occur with respect to memory M:

S1. A new element is added to M.
S2. An existing element is removed from M.
S3. An existing element is activated y times.

If S3 occurs with respect to element m_i, a new pair (t, y) is added to m_i. To maintain a bounded history size, if $|m_i| > c$, the pair with smallest a (i.e. the oldest) is removed from m_i.

Thus, given a memory M, we define that the *forgetting* problem, at each time step, t, is to identify the subset of elements, $D \subseteq M$, that have decayed since the last time step.

Efficient Approach

Given this problem definition, a naïve approach is to determine the decay status of each element every time step. This test requires computation $O(|M|)$, scaling linearly with average memory size. The computation expended upon each element, m_i, will be linear in the number of time steps where $m_i \in M$, estimated as $O(L)$ for a static element.

Our approach draws inspiration from the work of Nuxoll, Laird, and James (2004): rather than checking memory elements for decay status, "predict" the future time step when the element will decay. First, at each time step, examine elements that either (S1) weren't previously in the memory or (S3) were activated. The number of items requiring inspection is bounded by the total number of elements ($|M|$), but may be a small subset. For each of these elements, predict the time of future decay (discussed shortly) and add the element to a map, where the map key is the predicted time step and the value is a set of elements predicted to decay at that time. If the element was already within the map (S3), remove it from its old location before adding to its new location. All insertions/removals require time at most logarithmic in the number of distinct decay time steps, which is bounded by the total number of elements ($|M|$). At any time step, the set D is those elements in the set indexed by the current time step that are decayed.

To predict element decay, we perform a novel, two-phase process. After a new activation (S3), we first employ an *approximation* that is guaranteed to underestimate the true value of L. If, at a future time step, we encounter the element in D and it has not decayed, we then compute the exact prediction using a binary parameter search.

We approximate L for an element m as the sum of L for each independent pair $(a, k) \in m$. Here we derive the closed-form calculation: given a single element pair at time t, we solve for t_p, the future time of element decay…

$$\ln(k \cdot [t_p + (t-a)]^{-d}) = \theta$$

$$\ln(k) - d \cdot \ln(t_p + (t-a)) = \theta$$

$$t_p = e^{\frac{\theta - \ln(k)}{-d}} - (t-a)$$

Since k refers to a single time point, a, we rewrite the summed terms as a product. Furthermore, we time shift the decay term by the difference between the current time step, t, and that of the element pair, a, thereby predicting L.

Computing this approximation for a single pair takes constant time (and common values can be cached). Overall approximation computation is linear in the number of pairs, which is bounded by c, and therefore $O(1)$. The computation required for binary parameter search of an element is $O(\log_2 L)$. However, this computation is only necessary if the element has not decayed, or removed from M.

Evaluation

This approach has been empirically evaluated for long-term tasks in the procedural and working memories of Soar (Derbinsky & Laird, 2012). In this paper, we focus on the quality and efficiency of our prediction approach and utilize synthetic data. Our data set comprises 50,000 memory elements, each with a randomly generated pair set. The size of each element was randomly selected from between 1 and 10, the number of activations per pair (k) was randomly selected between 1 and 10, and the time of each pair (a) was randomly selected between 1 and 999. We verified that each element had a valid history with respect to time step 1000, meaning that each element would not have decayed before t=1000. Also, each element contained a pair with at least one access at time point 999, which simulated a fresh activation (S3). For all synthetic experiments we used decay rate d=0.8 and threshold θ=-1.6. Given these constraints, the largest possible value of L for an element is 3332.

We first evaluate the quality of the decay approximation. In Figure 1, the y-axis is the cumulative proportion of the elements and the x-axis plots absolute temporal error of the approximation, where a value of 0 indicates that the approximation was correct, and non-zero indicates how many time steps the approximation under-predicted. We see that the approximation was correct for over 60% of the elements, but did underestimate over 500 time steps for 20% of the elements and over 1000 time steps for 1% of the elements. Under the constraints of this data set, it is possible for this approximation to underestimate up to 2084 time steps. We also compared the prediction time, in microseconds, of the approximation to an exact calculation using binary parameter search. The maximum computation time across the data set was >19x faster for the approximation (1.37 vs. 26.28 μsec./element) and the average time was >15x faster (0.31 vs. 4.73 μsec./element).

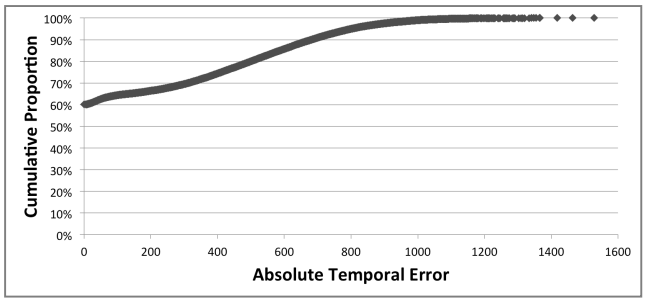

Figure 1. Evaluation of decay-approximation quality.

We did not compare these results with a naïve approach, as results would depend upon a model of memory size ($|M|$).

In conclusion, we presented a novel, two-phase forgetting approach that maintains fidelity to the base-level activation model and scales to large memories. The experimental results show that the first phase is a high-quality approximation and is an order of magnitude less costly than the exact calculation in the second phase.

Acknowledgments

We acknowledge the funding support of the Air Force Office of Scientific Research, contract FA2386-10-1-4127.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., Qin, Y. (2004). An Integrated Theory of the Mind. *Psychological Review, 111 (4)*, 1036-1060.

Anderson, J. R., Reder, L., Lebiere, C. (1996). Working Memory: Activation Limitations on Retrieval. *Cognitive Psychology, 30*, 221-256.

Chong, R. (2003). The Addition of an Activation and Decay Mechanism to the Soar Architecture. *Proc. of the 5th Intl. Conf. on Cognitive Modeling* (pp. 45-50).

Chong, R. (2004). Architectural Explorations for Modeling Procedural Skill Decay. *Proc. of the 6th Intl. Conf. on Cognitive Modeling*.

Derbinsky, N., Laird, J. E. (2011). A Functional Analysis of Historical Memory Retrieval Bias in the Word Sense Disambiguation Task. *Proc. of the 25th AAAI Intl. Conf. on Artificial Intelligence* (pp. 663-668).

Derbinsky, N., Laird, J. E. (2012). Competence-Preserving Retention of Learned Knowledge in Soar's Working and Procedural Memories. *Proc. of the 11th Intl. Conf. on Cognitive Modeling*.

Derbinsky, N., Laird, J. E., Smith, B. (2010). Towards Efficiently Supporting Large Symbolic Declarative Memories. *Proc. of the 10th Intl. Conf. on Cognitive Modeling* (pp. 49-54).

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

Nuxoll, A., Laird, J. E., James, M. (2004). Comprehensive Working Memory Activation in Soar. *Proc. of the 6th Intl. Conf. on Cognitive Modeling* (pp. 226-230).

Petrov, A. (2006). Computationally Efficient Approximation of the Base-Level Learning Equation in ACT-R. *Proc. of the 7th Intl. Conf. on Cognitive Modeling* (pp. 391-392).