Computationally Efficient Forgetting via Base-Level Activation

Nate Derbinsky & John E. Laird

University of Michigan

supported by

ABSTRACT

Base-Level Activation (Anderson et al. 2004)
- Core to ACT-R declarative module
- Models historical retrieval bias and errors
- Used to model forgetting

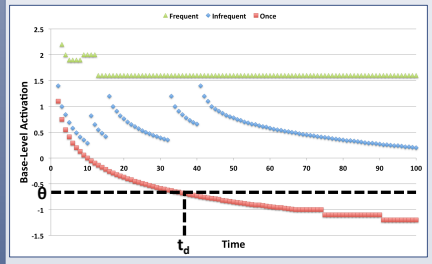
How to efficiently remove items from memory while preserving model fidelity?

PROBLEM FORMULATION

Base-Level Activation

$$B(t,d) = \ln(\sum_{i=1}^{n}[t-t_i]^{-d})$$

Activation at time *t* is the logarithm of an exponential decay over usage history, with respect to rate *d*.



Decay Problem

$$B(t_d,d) \overset{?}{<} \theta$$

When will activation fall below threshold?

NAÏVE APPROACH

Compute the activation of each memory at each point in time: *O*(# memories)

EFFICIENT IMPLEMENTATION

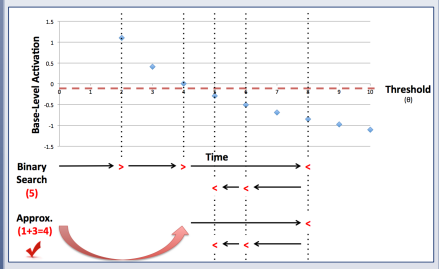
On activation, predict decay (t_d)
1. On access, *fast* via approximation
2. If needed, in future, *exact* via binary parameter search ~ $O(\log_2 t_d)$

Maintain decay map: M[t] = {m_1, m_2, …}
- Key: time step
- Value: set of elements predicted to decay at that time step

Memory events:
- New: insert into map at t_d
- Remove: find in map at t_d, remove
- Activate: find in map at $t_{d\text{-old}}$, remove, insert into map at $t_{d\text{-new}}$

At time step *t*, check elements at map[t]:
- If decayed, remove
- Else, predict via #2

EFFICIENT AND CORRECT



Novel Approximation

Sum of t_d for each time step independently.

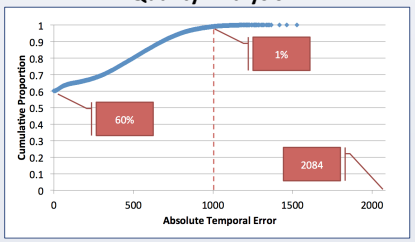
Notes
- Closed-form computation is *O*(1)
- Efficient to cache common values
- Guaranteed to underestimate t_d

SYNTHETIC EVALUATION

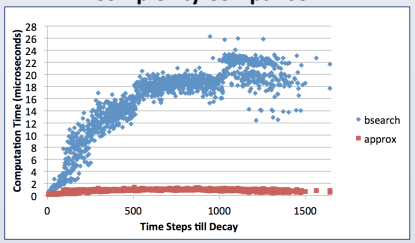
50,000 memories, valid @ t=1000
- *n* ~ *U*(1, 10)
- *# activations* ~ *U*(1,10)
- t_i ~ *U*(1,999), one @ t=999

Experimental conditions: *d* = 0.8, θ = -1.6
- Largest possible t_d=3332

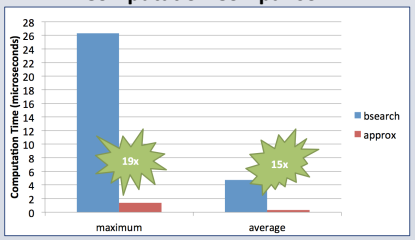
Quality Analysis



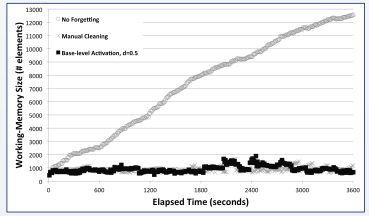
Complexity Comparison

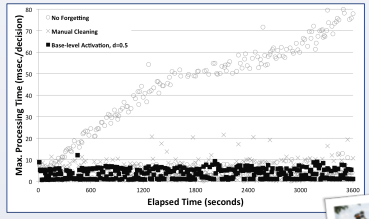


Computation Comparison

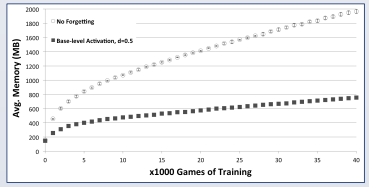


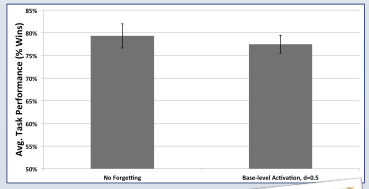
WORKING-MEMORY FORGETTING





PROCEDURAL FORGETTING





Maximum decision time = 6 msec.