

# Computationally Efficient Approximation of the Base-Level Learning Equation in ACT-R

Alexander A. Petrov (apetrov@alexpetrov.com)  
 Department of Psychology, Colorado University, Boulder  
 1905 Colorado Ave, Boulder, CO 80309 USA

## Introduction

Availability of human memories for specific items is sensitive to frequency and recency. Sensitivity to these statistics has high adaptive value because they predict the likelihood of encountering the same items in the future (Anderson & Milson, 1989; Anderson & Schooler, 1991). This property of memory accounts for a range of cognitive phenomena and is implemented in various models (e. g., Anderson & Milson, 1989; Petrov & Anderson, 2005). The ACT-R theory has been particularly successful and influential (Anderson & Lebiere, 1989; Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004). The declarative memory in ACT-R is organized in chunks, and each chunk has a *base-level activation* (BLA) reflecting the frequency and recency of its use. The equation that updates these quantities has been “*the most successful and frequently used part of the ACT-R theory*” (Anderson et al., 2004, p. 1042, emphasis added). One serious practical drawback, however, is that it is very expensive to compute. ACT-R models have ground to a halt because of its complexity. This article describes an efficient approximation that scales up well and preserves all desirable properties of the exact equation.

## Base-Level Learning in ACT-R

The ACT-R theory postulates that the BLA of each chunk is a logarithm of a sum of powers (Equation 1; Anderson & Lebiere, 1998). Each new use of the chunk adds another term to the sum, which then decays independently with rate  $d$ . The total count so far is denoted by  $n$ , and  $t_i$  is the time since the  $i$ -th use of the item. This equation gives rise to the activation dynamics illustrated by the dashed line in Figure 1, top. Note the three distinctive features of this curve: sharp transient peak immediately after each use, decay in the absence of use, and gradual accretion with frequent use.

$$B = \ln \left[ \sum_{i=1}^n t_i^{-d} \right] \quad (1)$$

Equation 1 is analytically intractable and its implementation requires a complete record of the exact time stamps of each use of each chunk. Slowly but surely, this excessive storage saturates the program workspace and aborts the simulation. A memory-efficient approximation is needed.

$$B \approx \ln \left[ \frac{n}{1-d} t_n^{-d} \right] \quad (2)$$

Such an approximation exists and has been in use in the ACT-R community for years. It is based on the insight that

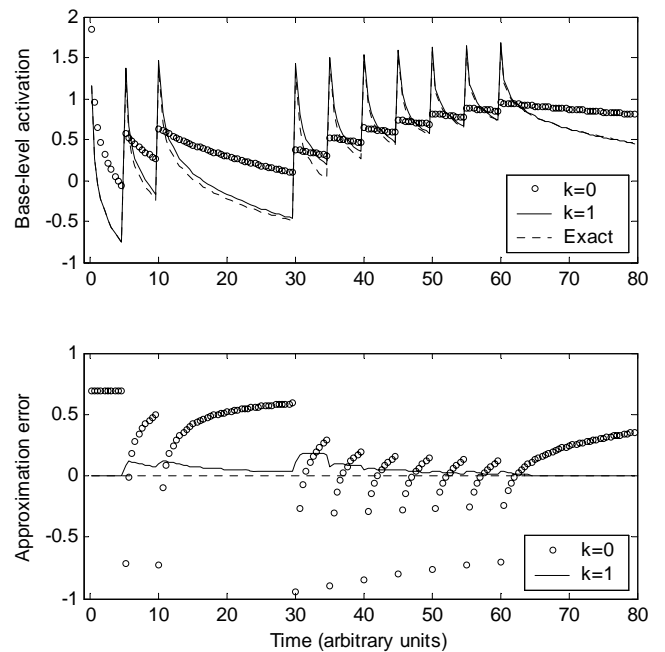


Figure 1: *Top*: Activation dynamics under exact base-level learning in ACT-R (Eq. 1) and approximations of different depths  $k$  (Eq. 3). *Bottom*: Approximation error, same data.

distant events diminish in importance as new events are added to the record. Equation 2 (Anderson & Lebiere, 1998) ignores the history of the chunk and retains only two pieces of information: its total lifetime  $t_n$  and number of uses  $n$ . The resulting learning curves are illustrated by the open circles in Figure 1.

Equation 2 accounts for the slow, frequency-driven component of the activation dynamics. Frequently used chunks grow stronger than infrequently used ones. It also gives rise to a weak decay effect when  $t_n$  increases while  $n$  does not. However, Equation 2 underestimates the decay effect, particularly during quiescent periods following bursts of activity. This is because it implicitly assumes all events are evenly spaced, and thus “fills in” the quiescent periods.

Equation 2 has a more serious flaw, however. It misses the rapid, recency-driven component manifested in sequential assimilation, priming, and other transient effects. It smoothes the activation dynamics too much because it ignores the timing of the most recent events. The improved approximation proposed in this article takes this critical information into account.

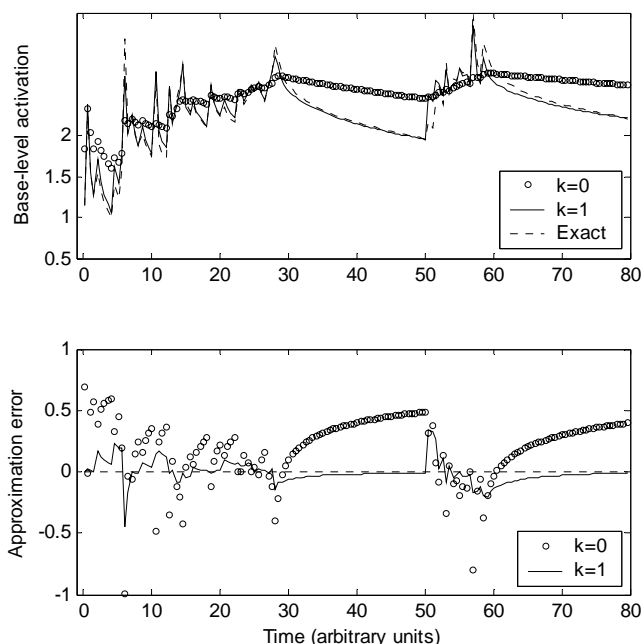


Figure 2: The approximation is robust to violations of the equal spacing assumption for depths  $k \geq 1$  (Equation 3).

### The New, Improved Approximation

The main idea is to keep the exact timing of a few most recent events and ignore the details of the distant past. A *depth* parameter  $k$  determines the cutoff point. When a new use of the chunk occurs, it assumes index  $i = 1$ , all previous uses shift one index up, and time stamp  $t_{k+1}$  passes into oblivion. Thus, the system keeps only a small amount of information about each chunk: its total lifetime  $t_n$ , the total number of uses  $n$ , and the most recent time lags  $t_1 \dots t_k$ . This allows the implementation to allocate a fixed amount of storage for each chunk, leading to dramatic efficiency improvement and scaling up.

$$B \approx \ln \left[ \sum_{i=1}^k t_i^{-d} + \frac{(n-k)(t_n^{1-d} - t_k^{1-d})}{(1-d)(t_n - t_k)} \right] \quad (3)$$

Equation 3 formalizes these ideas. See Appendix for derivation details. Note that the old approximation (Equation 2) is a special case for  $k=0$ . The improvement becomes apparent at greater depths. Even with  $k=1$ , Equation 3 accounts for the rapid, recency-driven component of the activation dynamics. That is because the transient BLA spike and subsequent decay are driven predominantly by the most recent event. Overall, the approximation is nearly perfect even for  $k=1$  (Figures 1 and 2). Greater depths should hardly ever be necessary.

$$B \approx \ln \left[ \sum_{i=1}^k \frac{1}{\sqrt{t_i}} + \frac{2(n-k)}{\sqrt{t_n} + \sqrt{t_k}} \right] \quad (4)$$

Equation 4 optimizes Equation 3 for the important special case  $d=0.5$ , which is the default decay parameter in ACT-R.

In conclusion, the new approximation captures all qualitative properties of the base-level activation dynamics: sharp transient peak immediately after each use, decay in the absence of use, and gradual accretion with frequent use. It tracks the theoretical Equation 1 very closely, but does not suffer from its computational complexity. Thus, it can be very useful for large simulations, allowing the ACT-R architecture to scale up to more realistic memory sizes and more prolonged learning periods. It has already been used successfully in a memory-based model of category rating and absolute identification (Petrov & Anderson, 2000, 2005).

MATLAB software is available at <http://alexpetrov.com>.

### 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*, 1036-1060.
- Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought*. Hillsdale, NJ: LEA.
- Anderson, J. R. & Milson, R. (1989). Human memory: An adaptive perspective. *Psychological Review*, *96*, 703-719.
- Anderson, J. R. & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychol. Science*, *2*, 396-408.
- Petrov, A. A. (2004). Additive or multiplicative perceptual noise? Two equivalent forms of the ANCHOR model. *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (pp. 922-927). Hillsdale, NJ: LEA.
- Petrov, A. A. & Anderson, J. R. (2000). A memory-based model of category rating. *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society* (pp. 369-374). Hillsdale, NJ: LEA.
- Petrov, A. A. & Anderson, J. R. (2005). The dynamics of scaling: A memory-based anchor model of category rating and absolute identification. *Psychological Review*, *112*, 383-416.

### Appendix

#### Mathematical Derivation of Equation 3

Split the sum in Equation 1 into “recent” and “distant” parts:

$$\sum_{i=1}^n t_i^{-d} = \sum_{i=1}^k t_i^{-d} + \sum_{i=1}^{n-k} t_{k+i}^{-d} \quad (5)$$

Assume equal spacing of the distant events:

$$t_{k+i} \approx t_k + ai \quad \text{for } i = 1 \dots n-k, \text{ where} \quad (6)$$

$$a = (t_n - t_k) / (n-k) \quad (7)$$

Approximate the distant sum with an integral quadrature:

$$\sum_{i=1}^{n-k} t_{k+i}^{-d} \approx \sum_{i=1}^{n-k} (t_k + ai)^{-d} \approx \int_0^{n-k} (t_k + ax)^{-d} dx \quad (8)$$

$$\sum_{i=1}^{n-k} t_{k+i}^{-d} \approx \frac{(n-k)(t_n^{1-d} - t_k^{1-d})}{(1-d)(t_n - t_k)} \quad (9)$$

Equation 3 in the main text follows from Equations 1, 5, and 9. The special case for  $d=0.5$  follows from Equation 3 and the factoring  $(t_n - t_k) = (\sqrt{t_n} - \sqrt{t_k})(\sqrt{t_n} + \sqrt{t_k})$ .