Intro to the ACT-R subsymbolic level for declarative memory

Adrian Brasoveanu

April 20, 2015

1 Understanding the (basic) activation equation

(1) Activation equation: $A_i = B_i + \sum_{j \in C} W_j S_{ji}$, for a chunk i and elements j that are part of the current goal chunk.
This equation has three major components:

a. Base-level learning equation: $B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right) = \log\left(\sum_{k=1}^{n} \frac{1}{\sqrt{t_k}}\right)$ (since usually $d = 0.5$), where t_k is the time since the k-th practice / access of chunk i.

b. Attentional weighting equation: $W_j = \frac{W}{n}$

c. Associative strength equation: $S_{ji} \approx \log\left(\frac{prob(i|j)}{prob(i)}\right)$

1.1 The base-level learning equation

(2) Base-level learning equation: $B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right) = \log\left(\sum_{k=1}^{n} \frac{1}{\sqrt{t_k}}\right)$ (since usually $d = 0.5$), where t_k is the time since the k-th practice / access of chunk i.

(3) Anderson and Schooler (1991, p. 396):

> In this paper we explore the issue of whether human memory is behaving optimally with respect to the pattern of past information presentation. Each item in memory has had some history of past use. For instance, our memory for one person's name may not have been used in the past month but might have been used five times in the month previous to that. What is the probability that the memory will be needed (used) during the conceived current day? Memory would be behaving optimally if it made this memory less available than memories that were more likely to be used but made it more available than less likely memories.
>
> In this paper we examine a number of environmental sources to determine how probability of a memory being needed varies with pattern of past use.

Let's first examine the Ebbinghaus (1913) retention data presented in his chapter 7.

(4) **Ebbinghaus (1913, ch. 7) retention data**

a. Stimulus materials: nonsense CVC syllables, about 2300 in number; mixed together, randomly selected to construct series of different lengths.

b. Method: learning to criterion; the subject repeats the material as many times as necessary to reach a prespecified level of accuracy (e.g., one perfect reproduction).

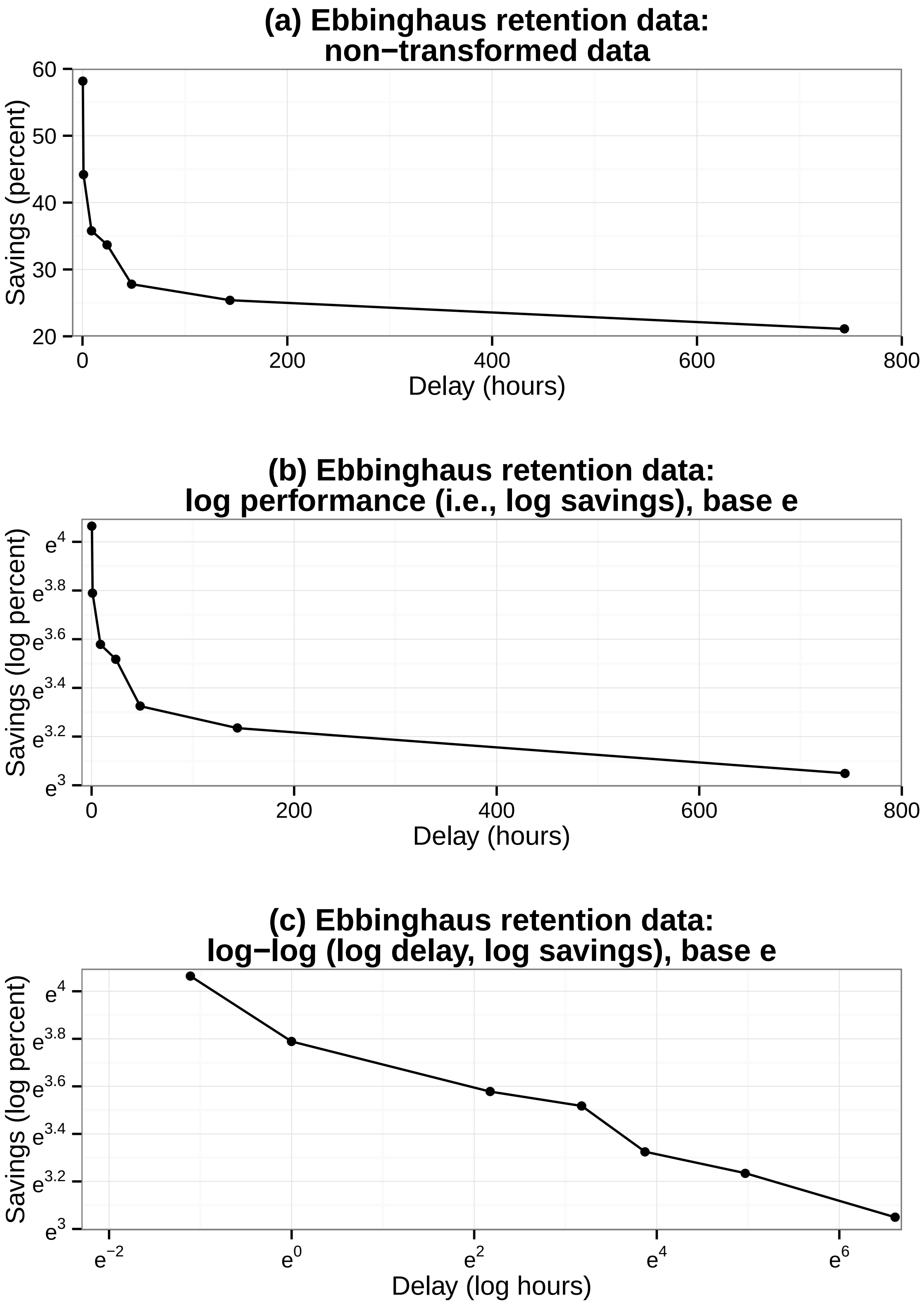
c. Retention measure: ‘savings’, i.e., subtracting the number of repetitions required to relearn material to a criterion from the number originally required to learn the material to the same criterion.

```
> ebbinghaus_data = read.csv("ebbinghaus_retention_data.csv", header=T)
> ebbinghaus_data

  delay_in_hours percent_savings
1           0.33            58.2
2           1.00            44.2
3           8.80            35.8
4          24.00            33.7
5          48.00            27.8
6         144.00            25.4
7         744.00            21.1

> summary(ebbinghaus_data)

 delay_in_hours   percent_savings
 Min.   :  0.3   Min.   :21.1
 1st Qu.:  4.9   1st Qu.:26.6
 Median : 24.0   Median :33.7
 Mean   :138.6   Mean   :35.2
 3rd Qu.: 96.0   3rd Qu.:40.0
 Max.   :744.0   Max.   :58.2
```


(a) Ebbinghaus retention data: non−transformed data

(b) Ebbinghaus retention data: log performance (i.e., log savings), base e

(c) Ebbinghaus retention data: log−log (log delay, log savings), base e

Figure 1: Ebbinghaus retention data

The forgetting curve plotted in panel (a) of Figure 1 is sometimes taken to reflect an underlying negative exponential forgetting function of the form:

(5) $P = Ae^{-bT}$, where P is the performance measure (percent savings in the Ebbinghaus data), T is the delay in time, and A, b are the parameters of the model.

But this predicts that performance should be a linear function of time if we log-transform P, and panel (b) of Figure 1 shows that is not the case:

(6) $\log(P) = \log(A) - bT$

Instead, we see a power function, as panel (c) of Figure 1 shows. That is, performance is a linear function of time only if you log-transform both of them:

(7) $\log(P) = \log(A) - b\log(T)$, i.e., $\boxed{P = AT^{-b}}$

The base-level learning equation $B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right)$ reflects exactly this: the base-level activation B_i is basically a log-performance value.

The basic idea of the account in Anderson and Schooler (1991):

(8)

> The basic idea is that at any point in time, memories vary in how likely they are to be needed and the memory system tries to make available those memories that are most likely to be useful. The memory system can use the past history of use of a memory to estimate whether the memory is likely to be needed now. This view sees human memory in some sense as making a statistical inference. However, it does not imply that memory is explicitly engaged in statistical computations. Rather, the claim is that whatever memory is doing parallels a correct statistical inference.
>
> What memory is inferring is something we call the need probability, which is the probability that we will need a particular memory trace now. The basic assumption developed in Anderson (1990) is that memories are considered in order of their need probabilities until the need probability is so low that it no longer is worth considering any more. If we let p be the need probability, C be the cost of considering a memory, and G be the gain associated with a successful retrieval, one should stop when $C > pG$.
>
> Despite the description of this process in terms that evoke images of memories being considered one at a time, there are equivalent parallel processes. We prefer a parallel model in which different memories are allocated different resources according to their need probability.
>
> [. . .]
>
> This analysis does allow predictions to be derived about the relationship between need probability and the dependent measures of recall latency and recall accuracy. With respect to recall latency, the critical assumption is that there is a distribution of memories in terms of their estimated need probabilities. The reasonable assumption is that there will be a mass of need probabilities near zero with a tail of a few higher probability memories; that is, to say the distribution of memories will be J-shaped or highly skewed. It is more convenient to think about the shape of such a distribution in terms of need odds. If p is need probability, then

> $q = p/(1-p)$ will be need odds. An odds measure has the advantage of varying from zero to infinity. Thus, the expectation is that most memories will have near-zero odds and a rapidly diminishing few will have higher odds. (Anderson and Schooler, 1991, p. 400)

In sum:

(9) The base-level activation equation encodes that (see Anderson and Schooler 1991, p. 407, and Anderson et al. 2004, p. 1042):

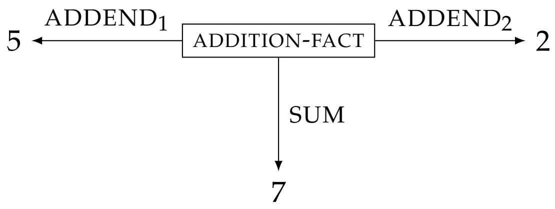
a. the strength of a memory trace provides an encoding of its need odds memory performance (base-level activation tracks log odds);

b. the strengths from individual presentations sum to produce a total strength (each presentation has an impact on odds, and the impacts of different presentations add up);

c. strengths of individual presentations decay as a power function of the time (the fact that the impact on odds of an individual presentation decays as a power function produces the power law of forgetting).

Let's work through some examples. Assume we have a fact – it can be an addition fact like the one below, or the lexical representation of a word etc.

(10) a. A chunk of type ADDITION-FACT with slots ADDEND$_1$, ADDEND$_2$ and SUM which models the fact 5 + 2 = 7. The slot values are the primitive elements 5, 2 and 7, respectively. Chunks are boxed, whereas primitive elements are simple text. A simple arrow ($\longrightarrow$) signifies that the chunk at the start of the arrow has the value at the end of the arrow in the slot with the name that labels the arrow.



b. The same chunk represented as an attribute-value matrix (AVM). We'll use only AVM representations from now on. The various components of the activation equation have been added.

$$\text{ADDITION-FACT}\left(B_i\right)\begin{bmatrix} \text{ADDEND}_1\left(S_{ji}\right): & 5\left(W_j\right) \\ \text{ADDEND}_2\left(S_{ji}\right): & 2\left(W_j\right) \\ \text{SUM}: & 7 \end{bmatrix}$$

Assume this chunk is presented 5 times, once every 300 ms, starting at time 0 ms. We want to plot its base-level activation for the first 3500 ms.

We define a `base_activation` function: its inputs are the presentation times for the chunk, and also the moments of time at which to obtain activation. The output is the base-level activation values at the corresponding moments of time.

```
> base_activation <- function(pres_times, moments) {
+     base_act = numeric(length=length(moments))
+     for (i in 1:length(moments)) {
+         base_act[i] = sum(1/sqrt(moments[i] - pres_times[pres_times<moments[i]]))
+     }
+     base_act[which(base_act!=0)] = log(base_act[which(base_act!=0)])
+     return(base_act)
+ }
>
> pres_times = seq(0, 1200, length.out=5)
> moments = 0:3500
> base_act = base_activation(pres_times, moments)
```

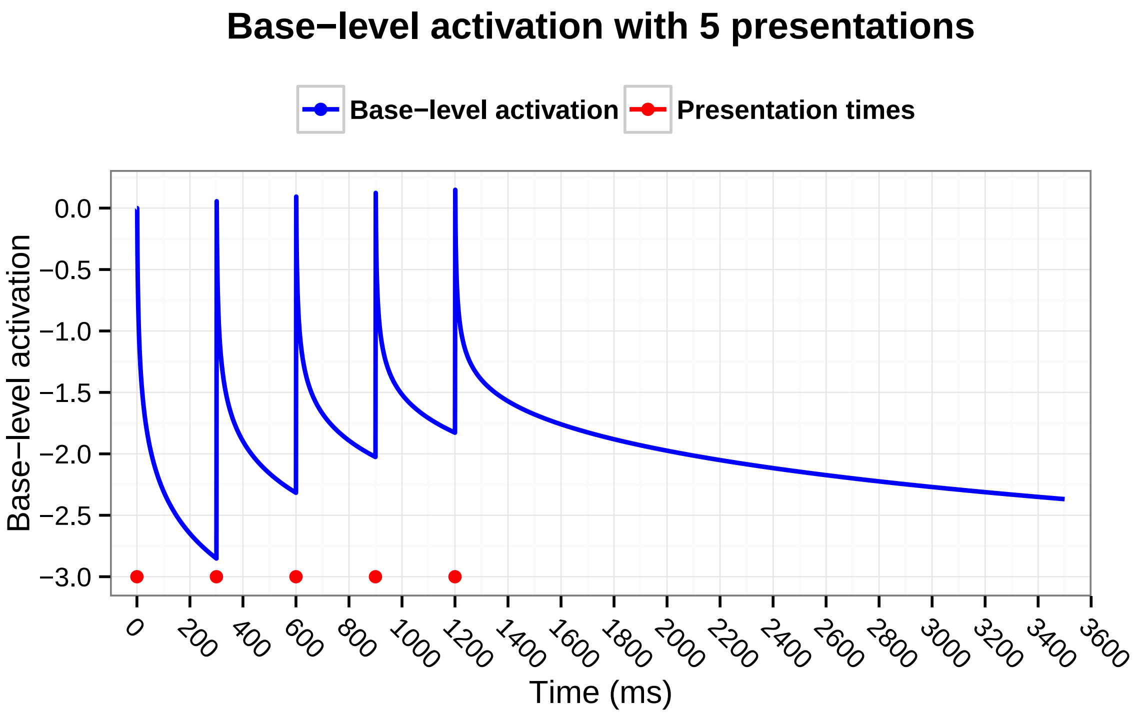


Figure 2: Base-level activation as a function of time

1.2 The attentional weighting equation

(11) Attentional weighting equation: $W_j = \frac{W}{n}$

W is usually set to 1, so the attention weights are usually $\frac{1}{n}$, where n is the number of sources of activation / terms.

1.3 The associative strength equation

(12) Associative strength equation: $S_{ji} \approx \log\left(\frac{prob(i|j)}{prob(i)}\right)$

S_{ji} is usually set to $S - \log(fan_j)$, where fan_j is the number of facts associated with term j. S is usually set to 2.

2 Activation, probability of retrieval, and latency of retrieval

(13) Probability of retrieval equation: $P_i = \frac{1}{1+e^{-\frac{A_i-\tau}{s}}}$, where s is the noise parameter and is typically set at about 0.4, and τ the retrieval threshold.

(14) Latency of retrieval equation: $T_i = Fe^{-A_i}$, where F is the latency factor.

(15) The threshold τ and the latency factor F vary from model to model, but there is a general relationship between them:
$F \approx 0.35e^{\tau}$
i.e., the retrieval latency at threshold (when $A_i = \tau$) is approximately 0.35 seconds.

Let's plot the probability and latency of retrieval for the same hypothetical case as above, assuming the activation of the items is just the base-level activation. We assume:

- noise $s = 0.4$
- threshold $\tau = -2$
- latency factor $F = 50$ (ms)

Note that according to the above equation, $F \approx 0.35e^{-2} \approx 0.35 \times 0.1353 \approx 0.04736$ (s), so our value of 50 ms is very close to this. Also note that this value is different from $F = 0.46$ in Vasishth et al. (2008, p. 692)), or $F = 0.14$ in Lewis and Vasishth (2005, p. 382).

```
> pres_times = seq(0, 1200, length.out=5)
> moments = 0:3500
> base_act = base_activation(pres_times, moments)
>
> s = 0.4
> tau = -2
> F = 50 # in ms
>
> prob_retrieval = 1/(1 + exp(-(base_act - tau)/s))
> latency_retrieval = F * exp(-base_act)
```

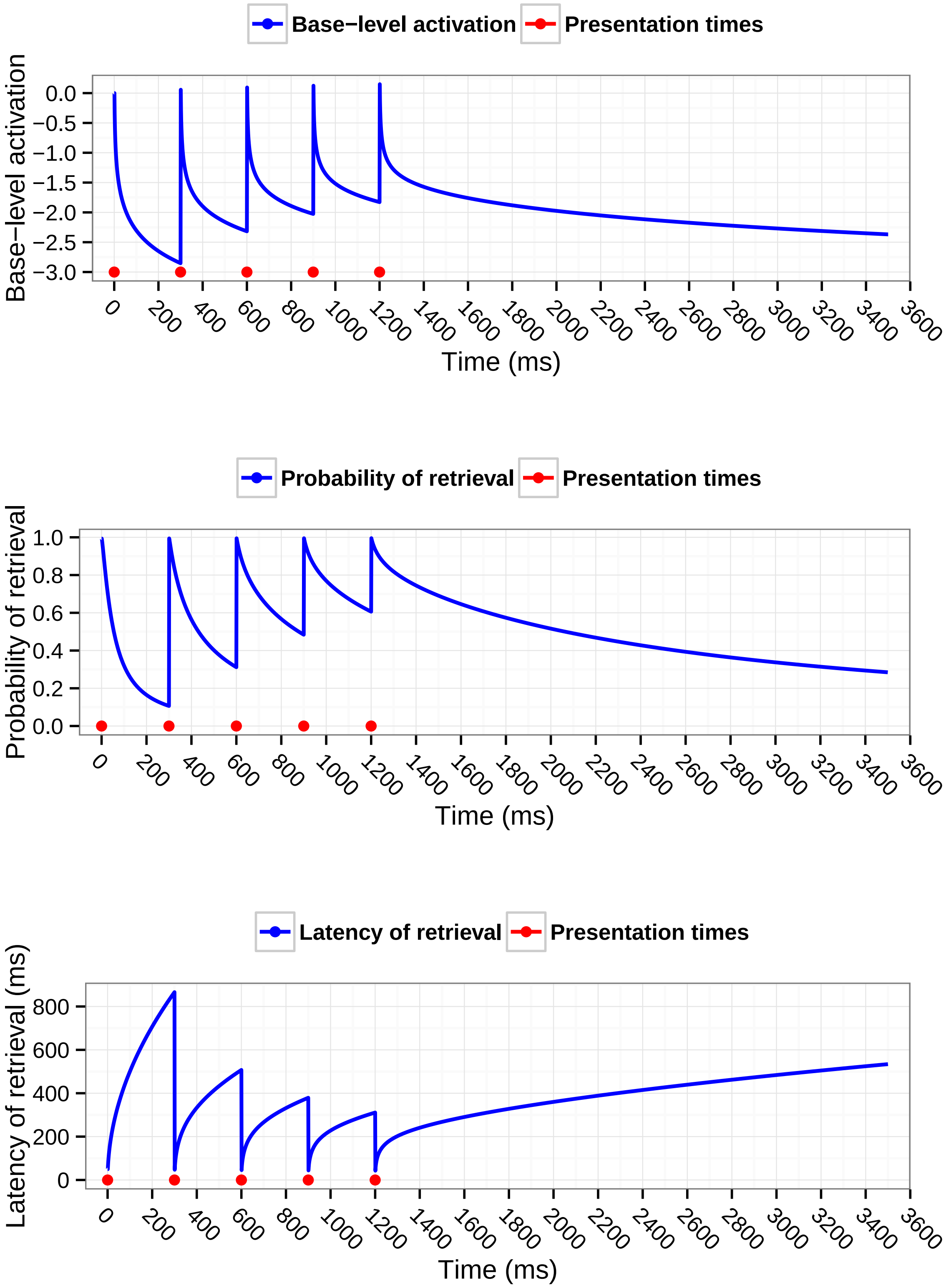


Figure 3: Base-level activation, probability of retrieval, and latency of retrieval as a function of time

2.1 Probability of retrieval

Let's take a closer look at probability of retrieval. We plot the odds of retrieval in addition to probability of retrieval, and also plot odds against activation.

```
> pres_times = seq(0, 1200, length.out=5)
> moments = 0:3500
> base_act = base_activation(pres_times, moments)
>
> s = 0.4
> tau = -2
>
> prob_retrieval = 1/(1 + exp(-(base_act - tau)/s))
> odds_retrieval = exp((base_act - tau)/s)
```

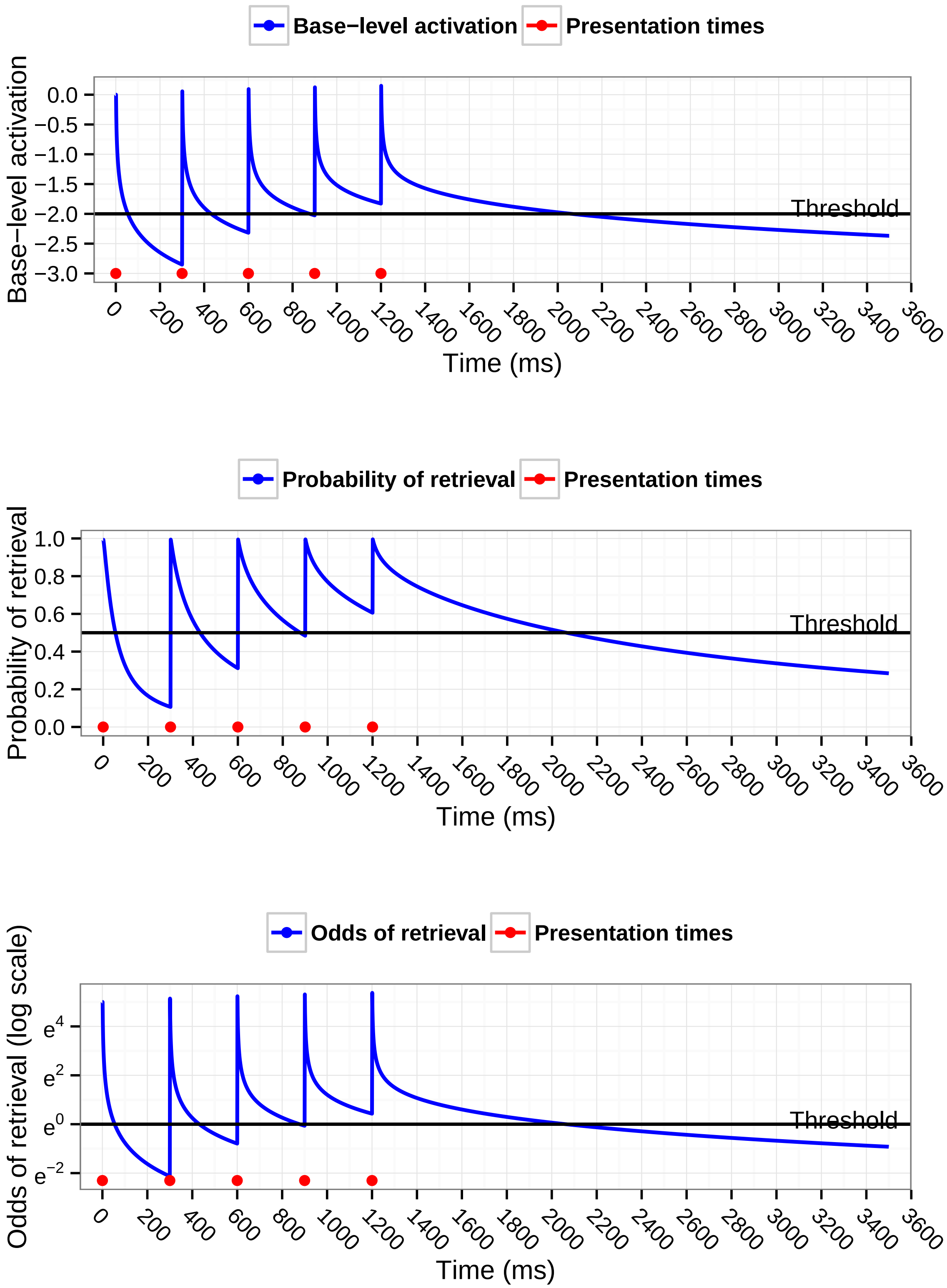


Figure 4: Base-level activation, probability of retrieval, and odds of retrieval as a function of time

Let's plot probability and odds of retrieval against activation. Note the *linear* relationship between activation and odds of retrieval on the log scale, i.e., log-odds, i.e., logits.

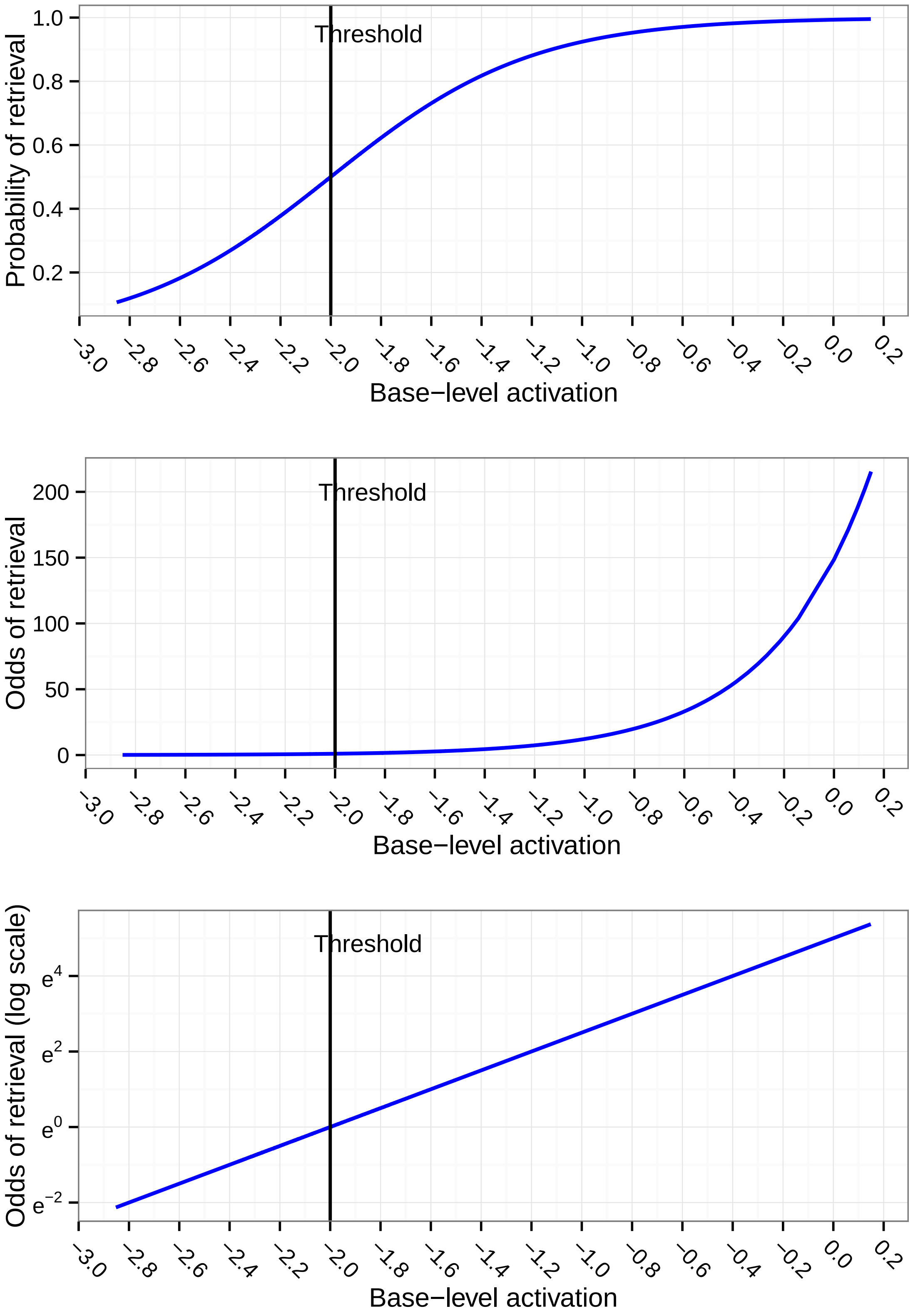


Figure 5: Probability and odds of retrieval as a function of activation

2.2 Latency of retrieval

Let's plot time of retrieval and log time of retrieval against activation – and also against log odds of retrieval. Note the *linear* relationship between activation and time of retrieval (or odds of retrieval) on the log scale.

You can get an intuitive interpretation for the latency scale parameter F by looking at how much time it takes to retrieve a chunk that has a threshold (τ) activation.

```
> pres_times = seq(0, 1200, length.out=5)
> moments = 0:3500
> base_act = base_activation(pres_times, moments)
>
> s = 0.4
> tau = -2
>
> prob_retrieval = 1/(1 + exp(-(base_act - tau)/s))
> odds_retrieval = exp((base_act - tau)/s)
>
> F = 50 # in ms
> latency_retrieval = F * exp(-base_act)
```

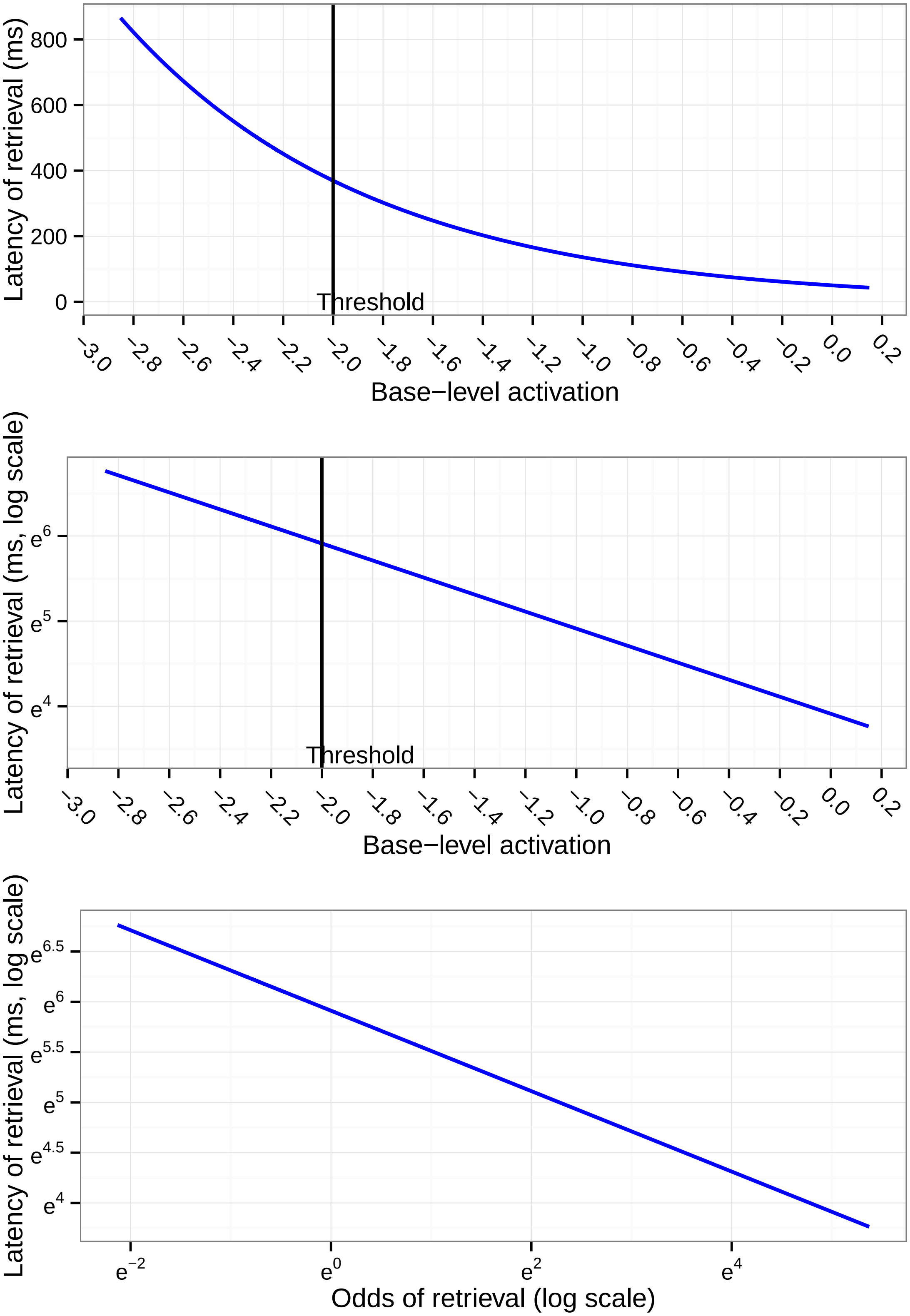


Figure 6: Time of retrieval as a function of activation and as a function of odds of retrieval

References

Anderson, John R. (1990). *The Adaptive Character of Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, John R. and Lael J. Schooler (1991). “Reflections of the Environment in Memory”. In: *Psychological Science* 2.6, pp. 396–408.

Anderson, John R. et al. (2004). “An Integrated Theory of the Mind”. In: *Psychological Review* 111.4, pp. 1036–1060.

Ebbinghaus, Hermann (1913). *Memory: A Contribution to Experimental Psychology*. Trans. by Henry A. Ruger and Clara E. Bussenius. New York: Teachers College, Columbia University. URL: http://psychclassics.yorku.ca/Ebbinghaus/index.htm.

Lewis, Richard and Shravan Vasishth (2005). “An activation-based model of sentence processing as skilled memory retrieval”. In: *Cognitive Science* 29, pp. 1–45.

Vasishth, Shravan et al. (2008). “Processing Polarity: How the Ungrammatical Intrudes on the Grammatical”. In: *Cognitive Science* 32, pp. 685–712.