Theory vs. Practice of Parameter values

Niels Taatgen
Two issues

- The joy of estimating :rt and :lf (and sometimes :bll and :ga and :ans)
- Meaning of parameters
Estimating parameters

- In many ACT-R models, declarative chunks are created that you might want to retrieve later on.

- In such a model, the desired behavior is that new chunks are hard to recall, unless they have a very strong context association, or have already been recreated a number of times.
Estimating parameters

- Problem: this balance is hard to achieve, because sometimes you end up in a situation where you retrieve everything right away, or a situation in which you retrieve nothing at all.

- To get into the desired situation, you have to carefully estimate, often through trial-and-error values for the retrieval threshold, base-level decay, goal activation, etc.

- Not easy to predict in advance due to the relatively unpredictable influence of spreading activation.
Earlier ACT-R books: The activation of a chunk represents an estimation of the log odds that you need that chunk in your current context:

\[ A_i \sim \log P(\text{chunk}_i \text{ needed} / \text{chunk}_i \text{ not needed}) \]

That means an activation of zero represents a probability that you need that chunk is 50%, which is extremely high.
Why are activation too high?

- The culprit is base-level learning
- Based on one experience:
  \[ A = \log t^{-d}, \text{ with } d \text{ by default } 0.5 \]
Default estimate

*Basis: single reference, $d = 0.5$*
What about higher d’s?

- Single reference, \( d = 1 \)
Probability after 1 second is still always 50%, which seems too high.

Hidden parameter in base-level learning equation:

\[ A = \log xt^{-d} \]

\[ x = 1 \] might just be too high.
Try \( d=1 \) and \( x=0.2 \)
Consequences

- Maybe solves the $d=0.5$ or $0.4$ problem
- Retrieval threshold will be negative
- Have to look at parameters in other equations
- Have to look at associations (maybe get the learning there going again?)
- Pay-off: true zero-parameter estimation?