Using large datasets to examine the performance space of ACT-R's base-level learning mechanism

Clayton Stanley and
Michael D. Byrne
Department of Psychology
Rice University
Houston, TX
clayton.stanley@rice.edu
byrne@rice.edu
http://chil.rice.edu

THE COMPUTER-HUMAN INTERACTION LABORATORY AT RICE UNIVERSITY

Overview

- Motivation for research
- Motivation for task
- Task descriptions
- Methods
    - Model descriptions
- Results
- Discussion
- Future Work

Motivation for Cognitive Modeling

- Recently become possible to evaluate psychological theories of declarative memory on large-scale real-world tasks
  - Theories can now be tested on hundreds of millions to billions of data points

- Now possible due to
  - Growth of social media and user created content
  - Publicly accessible large-scale datasets of user created content
  - Improved APIs and methods for extracting information
  - Improved data mining software and faster hardware

Motivation for Cognitive Modeling

- These information rich large-scale environments provide unique opportunities for research in declarative memory

- Can stress test and explore the impacts of the psychological constraints of each theory on a much larger set of data than has previously been possible

- Rapidly explore and evaluate different architectural constraints and their impact on retrieval accuracy

- Begin to test the declarative memory equations on a scale that is closer to the magnitude of chunks that are stored in human memory

Motivation for Task

- Social media sites are composed almost entirely of human-generated content
- There's a lot of it (over 400M tweets created per day)
- In addition to actively searching, users are subscribing to information streams
  - Followers and hashtags (Twitter), friends (Facebook), tags (StackOverflow)
- To support a user on these sites, how can we quickly and effectively connect users to the streams of content that they care about?

General Approach

- Frame the task of choosing a tag as a memory retrieval problem
  - Test how well two declarative memory theories can predict the correct tags

- If we can predict the chosen tags, then we can use this to recommend to them new information streams that match their interests

StackOverflow Example

stackoverflow Questions Tags Users Badges Unanswered

Why this is undefined behavior?

13

My answer to this question was this function:

```
inline bool divisible15(unsigned int x)
{
    //286331153 = (2^32 - 1) / 15
    //4008636143 = (2^32) - 286331153
    return x * 4008636143 <= 286331153;
}
```

It perfectly worked on my machine with VS2008 compiler, however here it doesn't work at all.

Does anyone has an idea, why it I get different results on different compilers? `unsigned` overflow isn't undefined behavior.

Important note: after some test it was confirmed it is faster than taking the remainder of the division by 15.

c++ | c | undefined-behavior

share | edit | flag

edited 13 mins ago
H2CO3
93.4k •11 •59 •127

asked 54 mins ago
user2623967
2,033 •5 •29

3 Is this faster than `(x % 15) == 0` ? – asveikau 50 mins ago

1 It doesn't show as undefined behavior to me? It probably integer overflows though. – PherricOxide 50 mins ago

@asveikau depends on compiler optimizations – user2623967 50 mins ago

1 Does `x * 4008636143` fit inside an int? – andre 49 mins ago

3 @millimoose Well... these are *unsigned* ints. The overflow behavior is specified. – Mystical 44 mins ago

add / show 9 more comments

Methods

- Analyze how a user's tag history influences chosen hashtags
- Use StackOverflow and Twitter popular-users dataset
- Use ACT-R base-level learning equations to model data
- Full-enumeration search to find best-fit base level activation term
  - Parameter might be specific to each user

Datasets

- StackOverflow
  - Used newest StackOverflow dataset released to public
    - Carved out 12 subsets of SO users across two dimensions: total number of questions and reputation
    - Sampled at various levels of reputation (100k, 10k, 1k, …) and total number of questions (500, 400, 300, …)
    - ~500 users per subset

- Twitter
    - Carved out 12 subsets of Twitter users across two dimensions: number of followers and total number of tweets
    - Sampled Twitter users at various levels of total popularity (10M, 1M, 100k, …) and total number of tweets (1M, 100k, …)
    - ~100 users per subset

Models

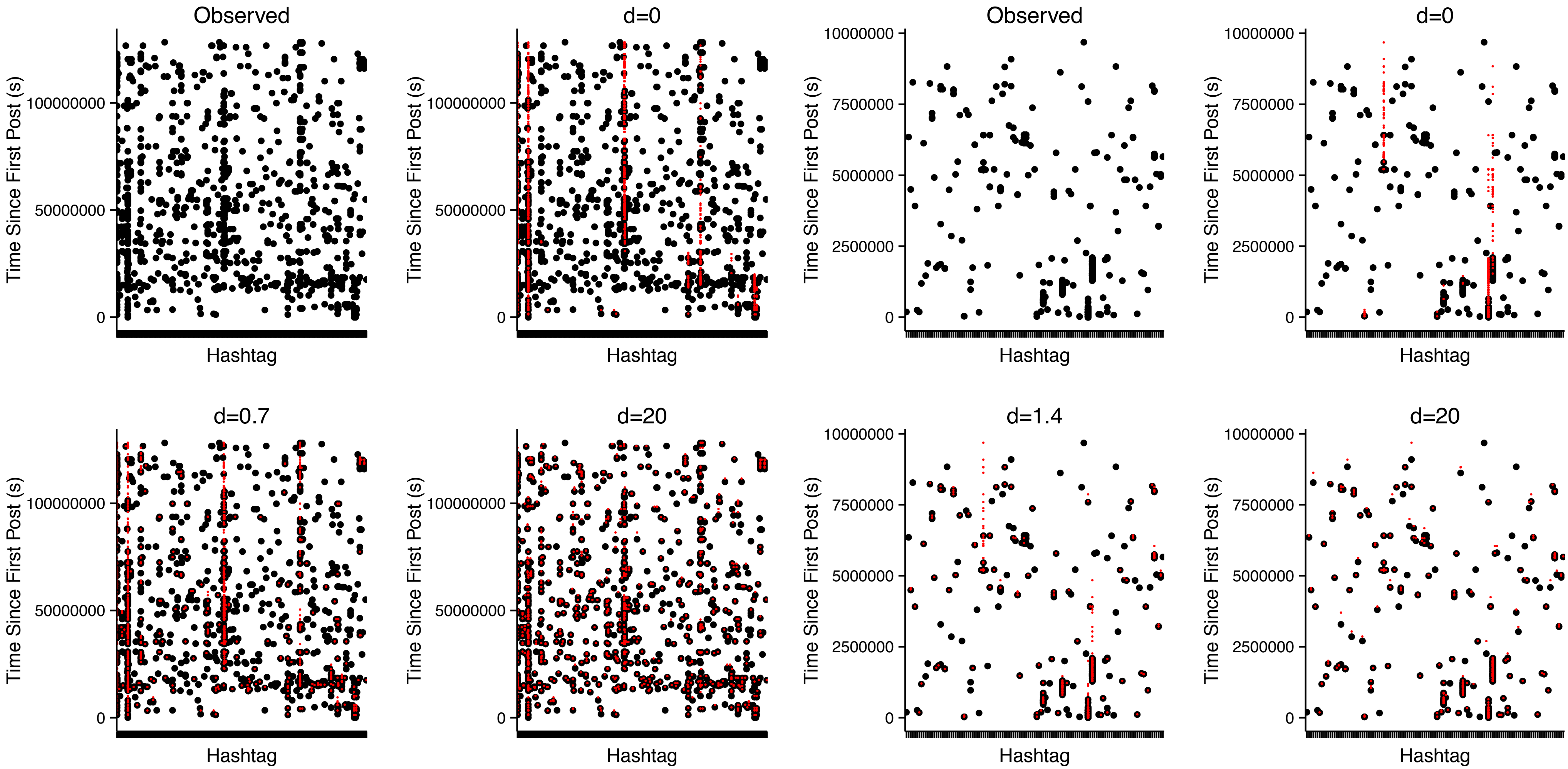
Common Name	Equation
Base Level Activation	$B_i = log \sum_{j=1}^{n} t_j{}^{-d}$
Optimized Learning	$B_i = log \frac{n}{1-d} - d * logL$

- Two different models: The standard equation and simplified form
- Simplified form assumes equal spacing of presentations within each chunk, and consequently does not have to store every observation in DM

- Terms:
  - *t*: Time since presentation of each chunk
  - *d*: Decay rate parameter
  - *n*: Number of presentations of chunk
  - *L*: Time since first presentation of chunk

- OL is set to default in ACT-R, *d* is almost always set to 0.5

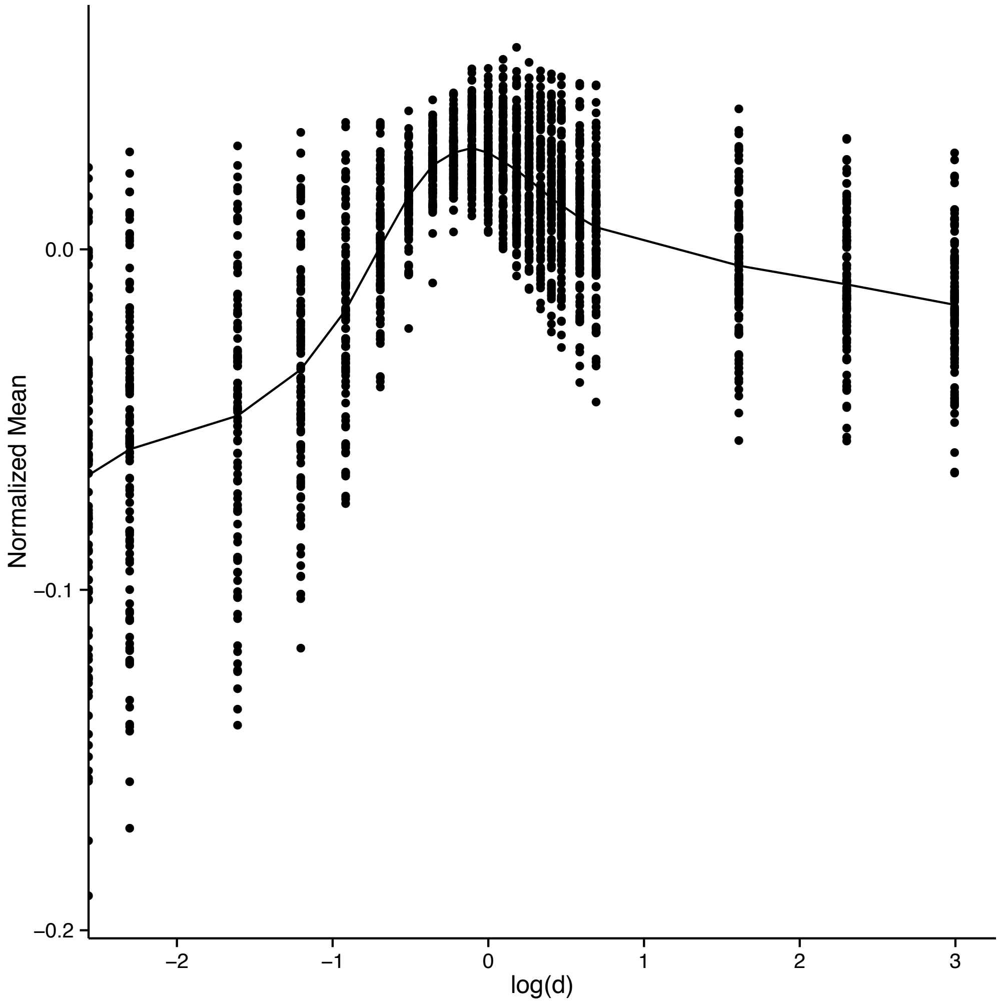
Results



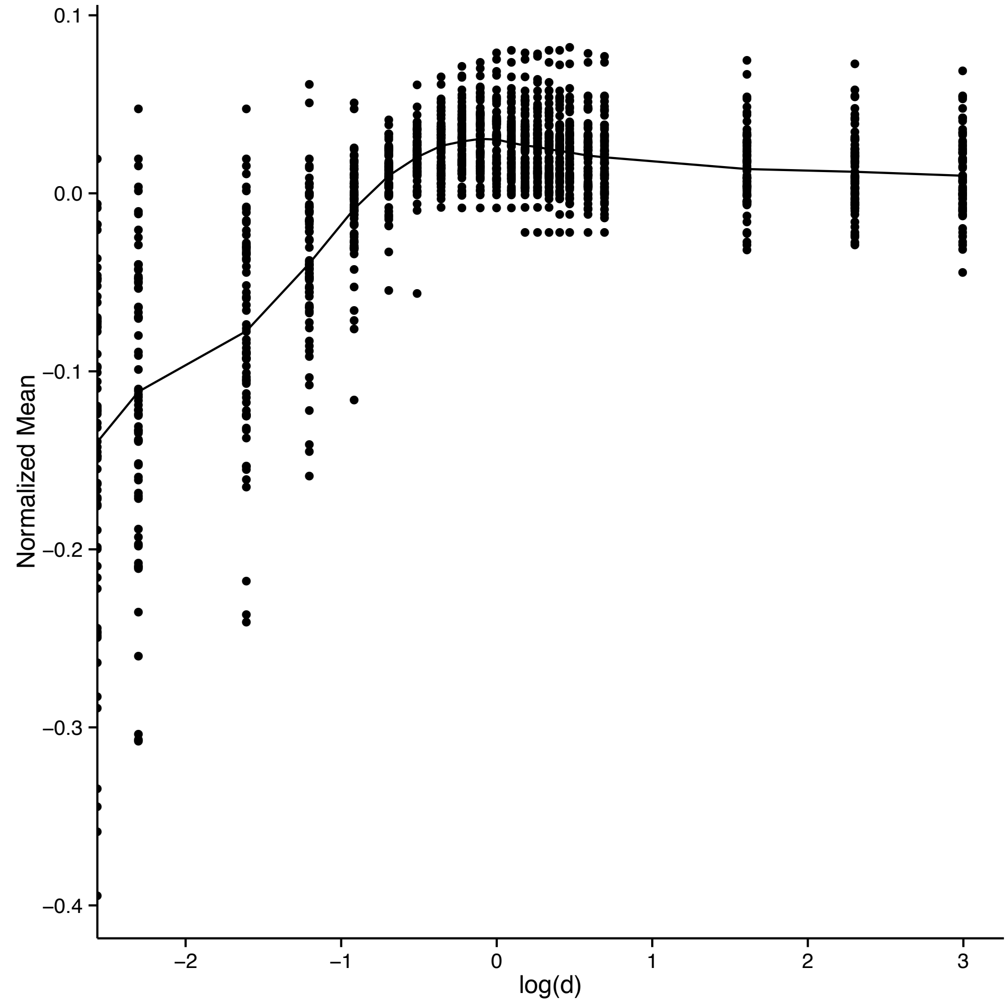
StackOverflow User

Twitter User

Results

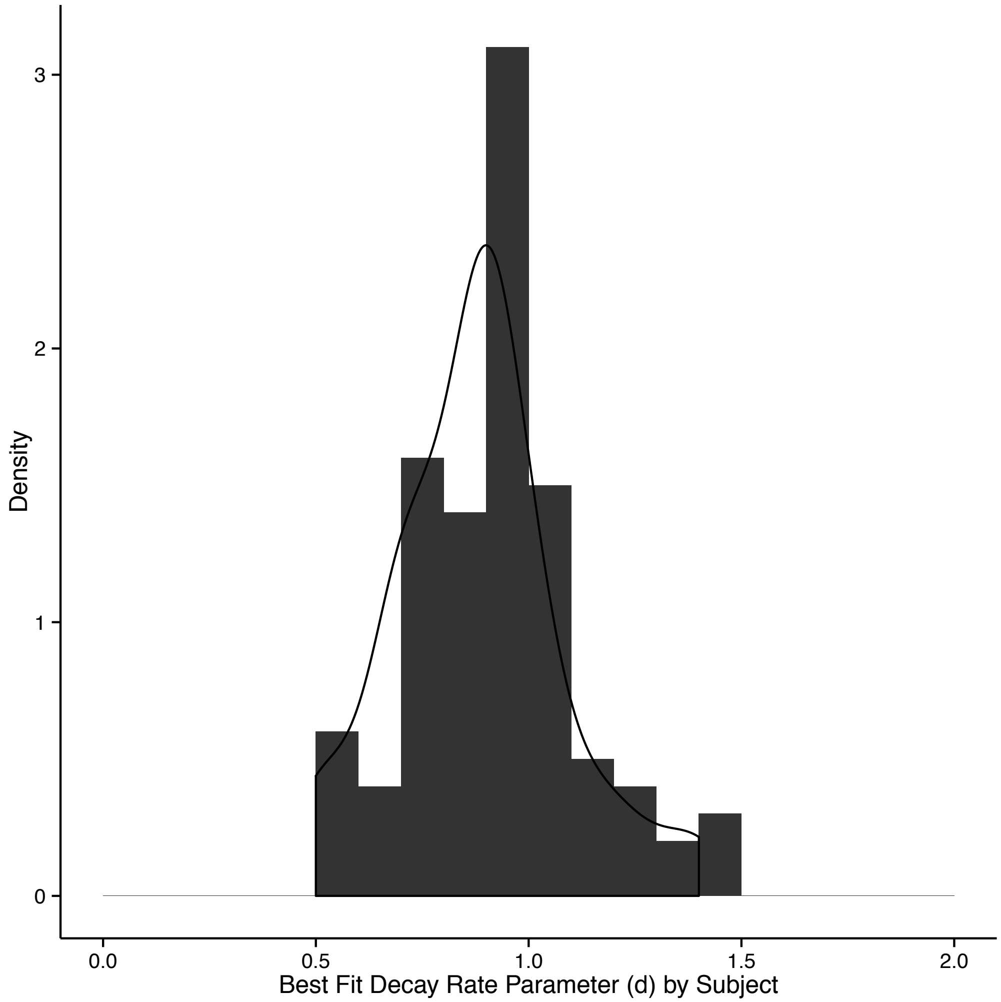


StackOverflow Subset

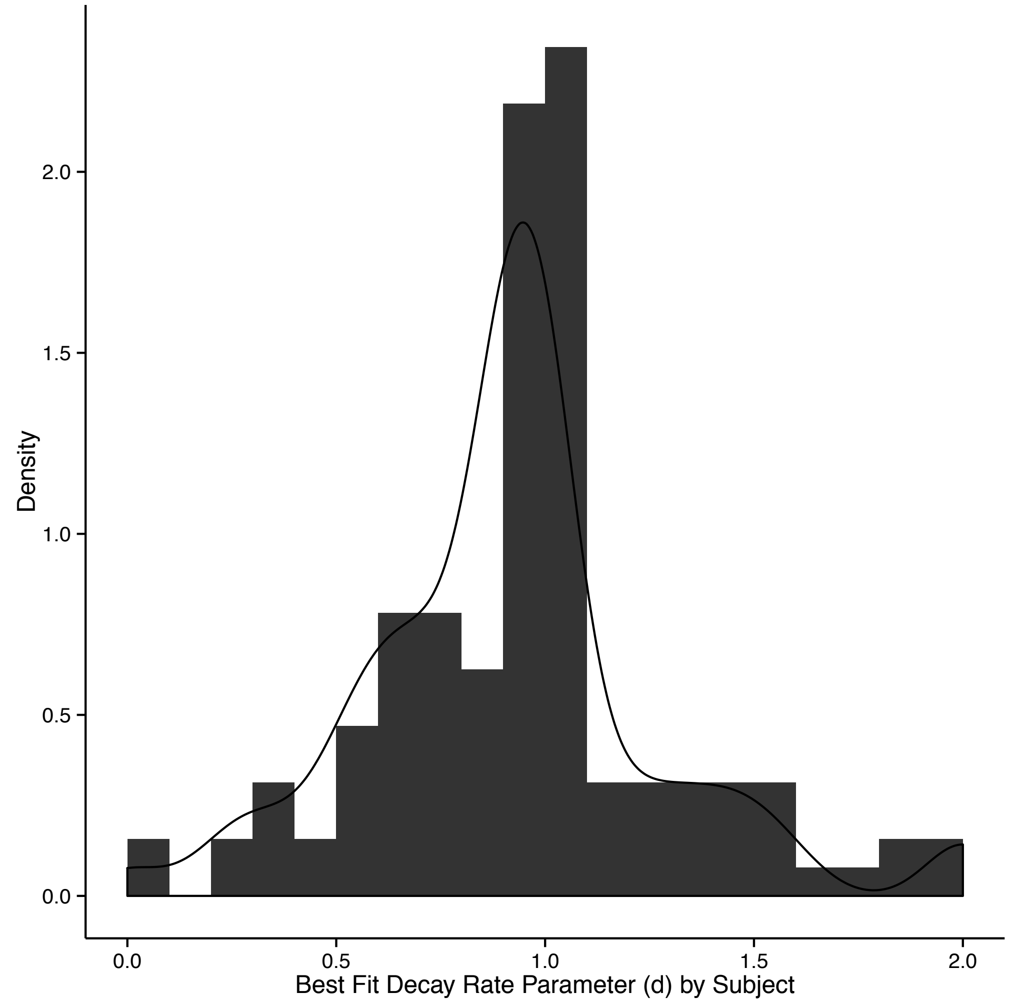


Twitter Subset

Results

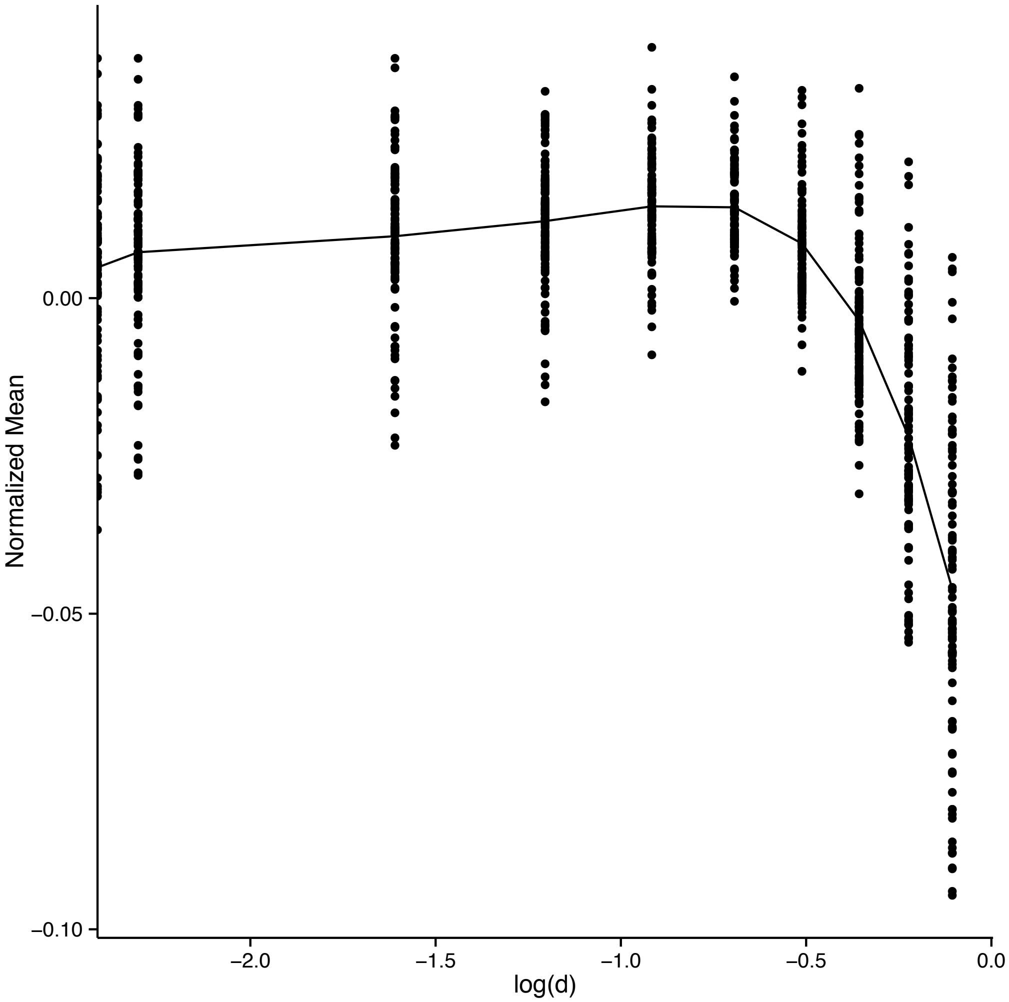


StackOverflow Subset

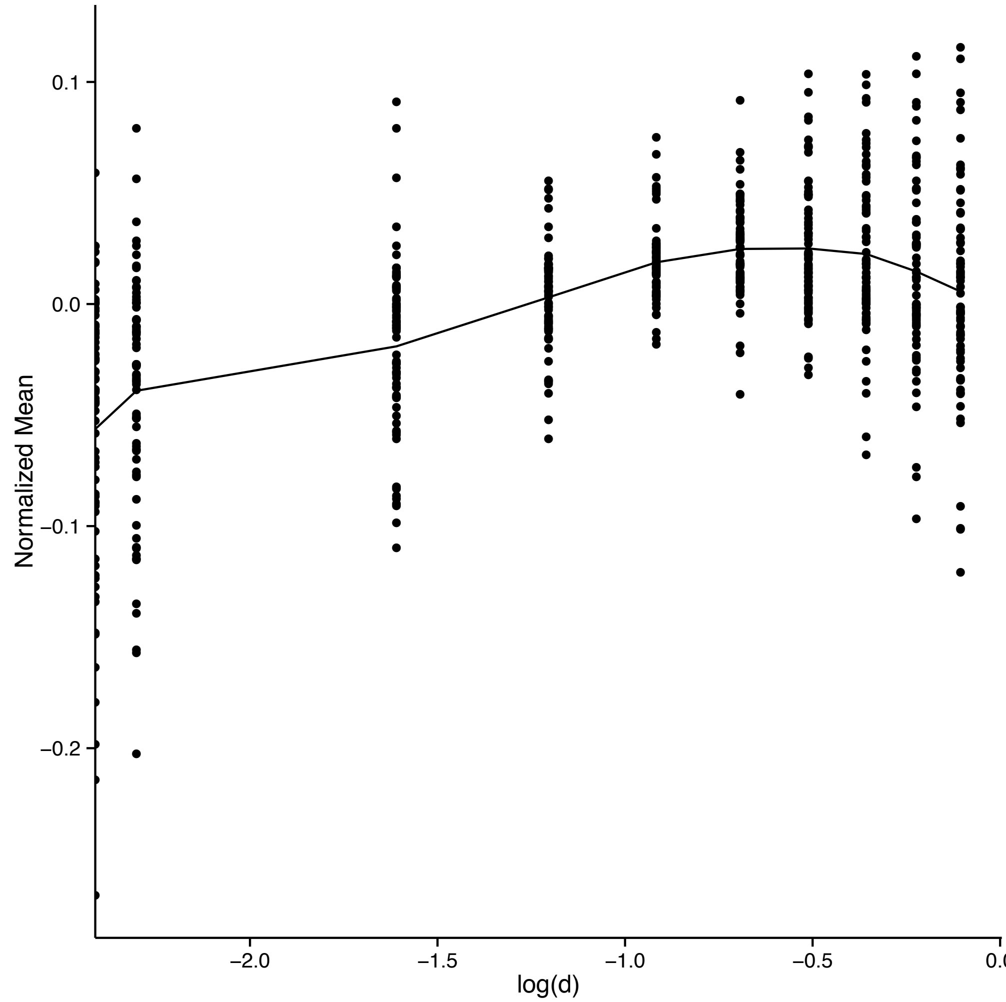


Twitter Subset

OL Results

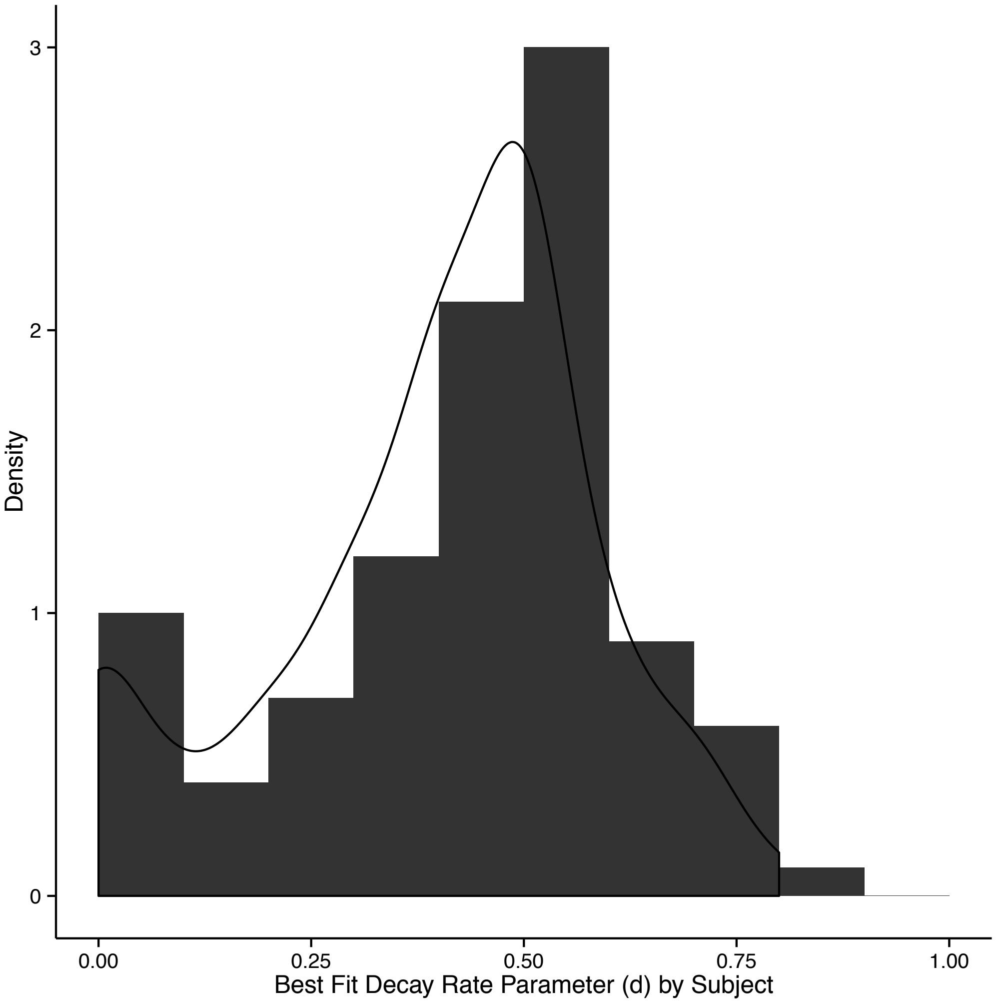


StackOverflow Subset

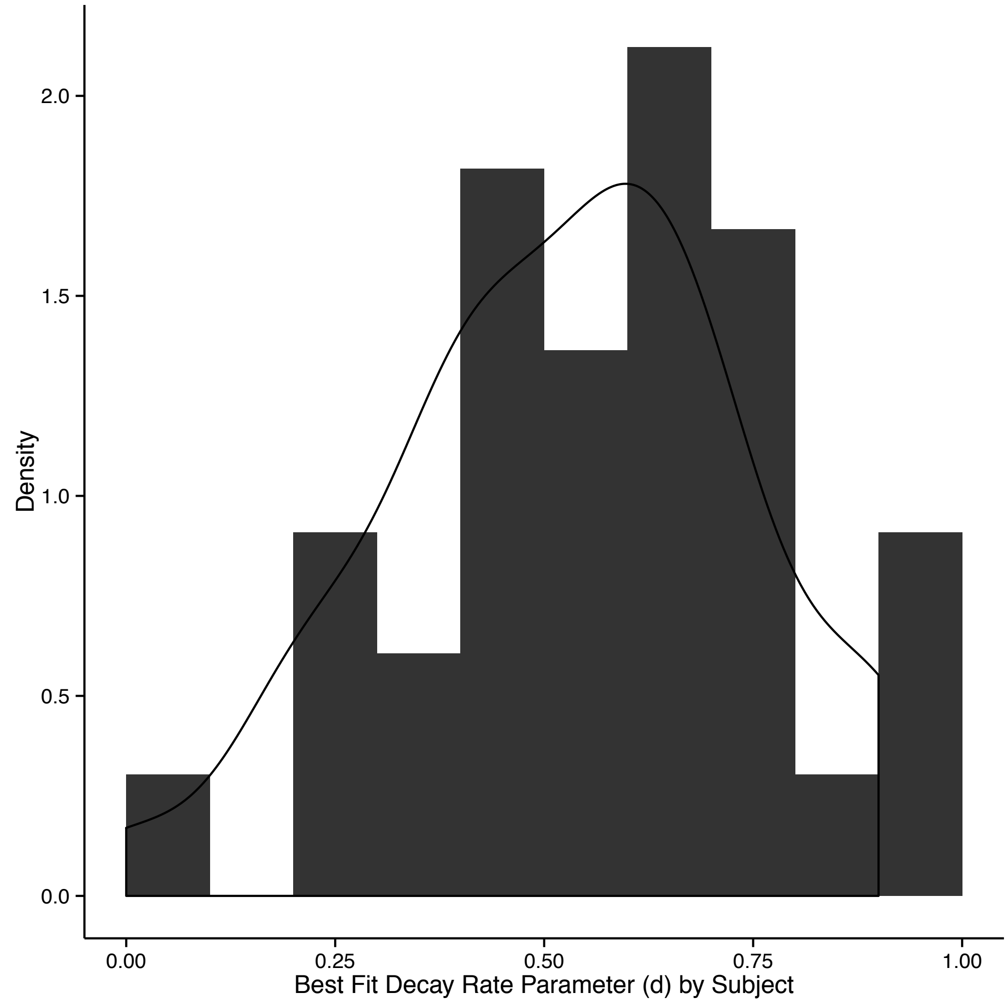


Twitter Subset

OL Results

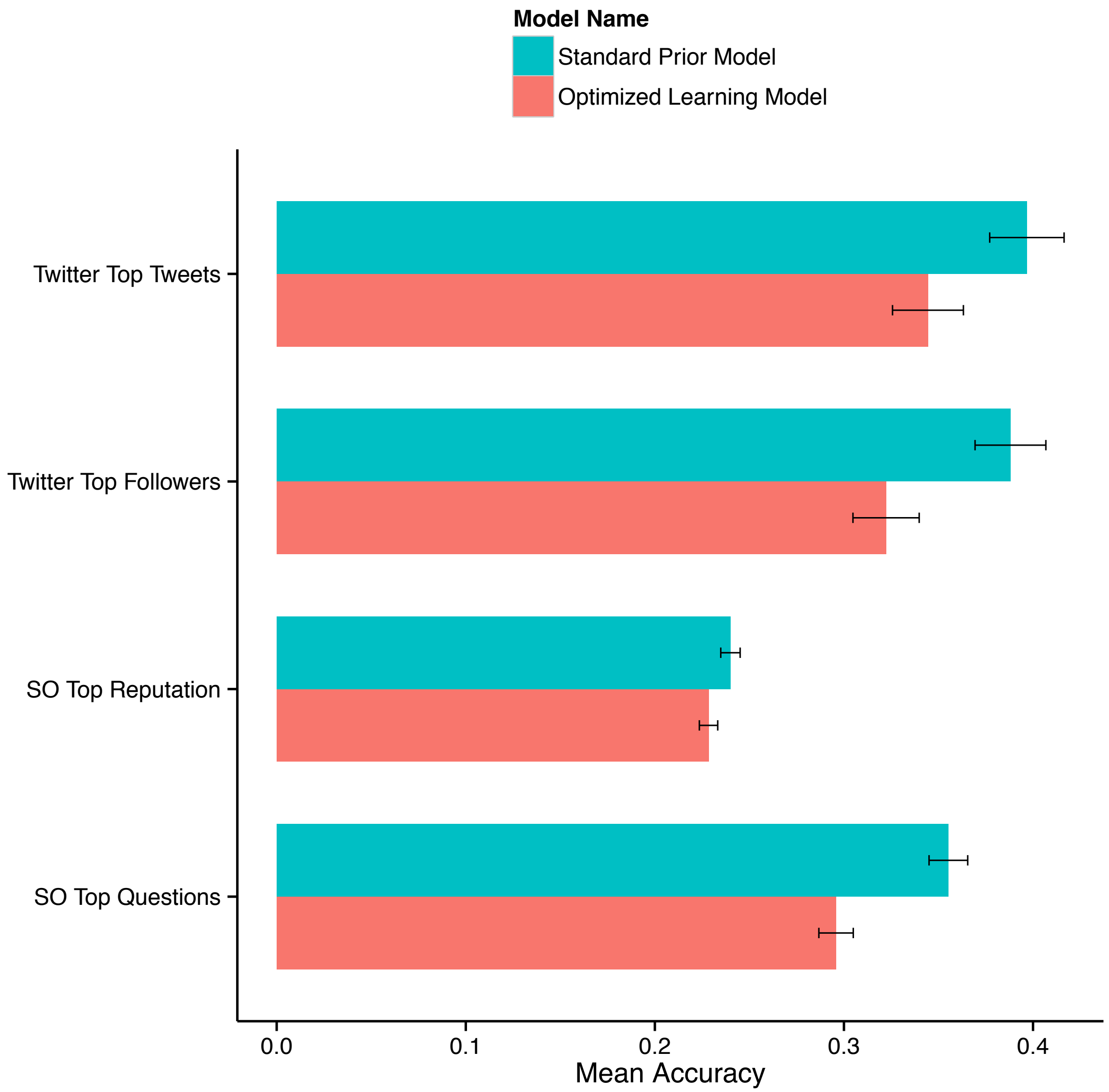


StackOverflow Subset

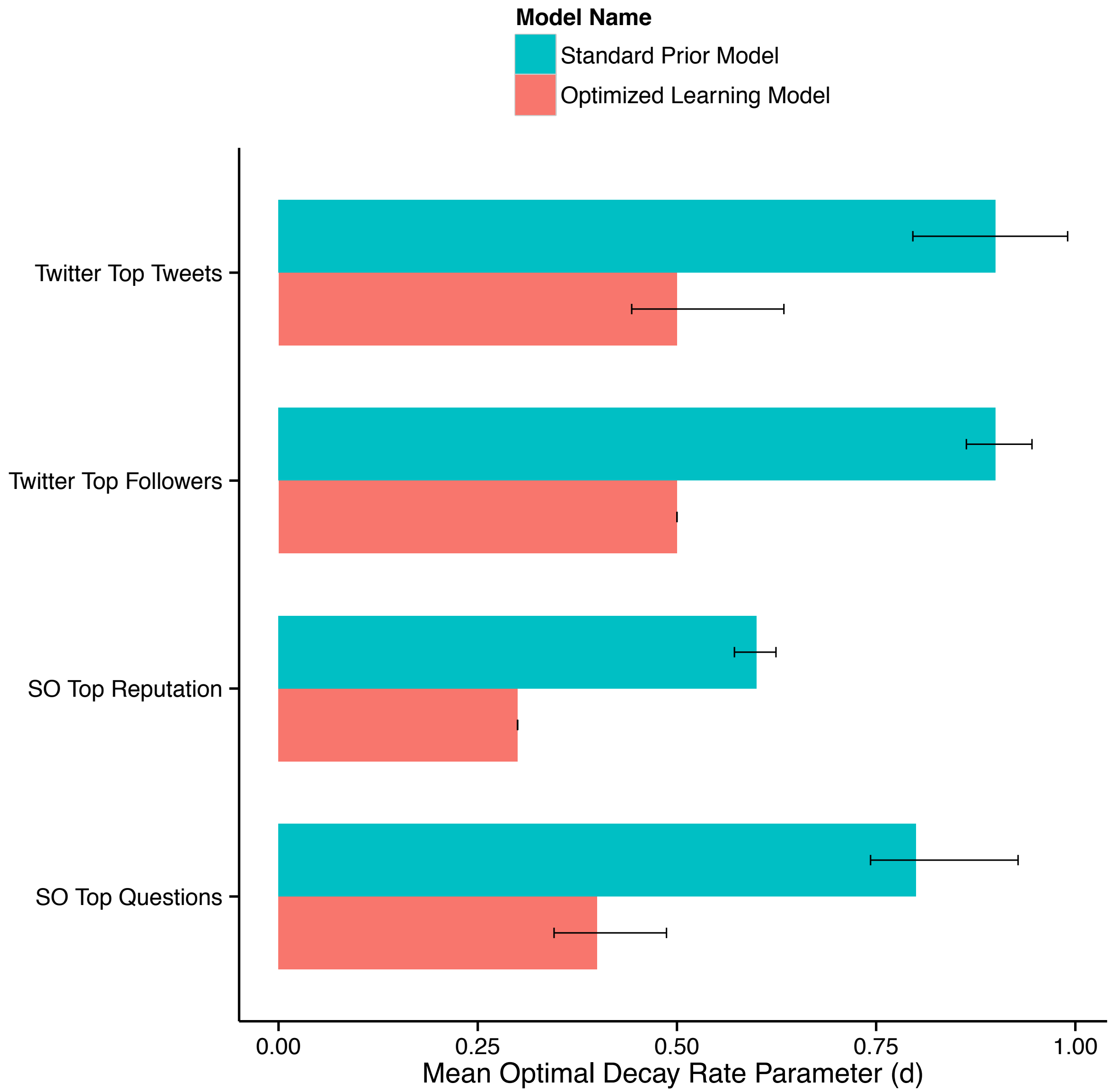


Twitter Subset

Overall Results



Overall Results



Discussion

- Surprising that retrieval accuracy is relatively high (34%) when *no context* and *only prior user tag history* is taken into account
  - Especially given the large tag space
  - Highlights the importance of a user's past tag history on future tag selections

- Accuracy is not as high when OL form of equation is used (29%)
  - Hard to tell if OL form of equation is any better than a pure frequency-based model, especially as total length of time increases
  - Makes sense given that this equation collapses to frequency model as time increases

Discussion

- Standard form of equation can be implemented efficiently
  - Retrieval times much less than what is required for the co-occurrence (context) component

- May be worthwhile to look into using the standard form of the base-level learning equation for a broader range of tasks
  - And increasing the decay rate slightly when doing so

- ACT-R's base-level learning mechanism is further validated
  - Efficiently handles large amounts of presentations (100k-1M per dataset)
  - Robust to noise inherent in messy real-world data (findings consistent across subsets)
  - Optimal d nicely blends frequency and recency information to generate relatively accurate predictions

Future Work

- Add in contextual information for each model
- Evaluate two different models for computing activation based on context: ACT-R's strength of association and a vector-based implementation
- Attempt to incorporate strengths of each model into the other
  - Word order from vector-based, base-level learning from ACT-R
- Explore and validate their performance space

Questions

- Thanks for your time