A Queueing-Theoretic Foundation for Optimal Spaced Repetition

S. Reddy[1] I. Labutov[2] S. Banerjee[3] T. Joachims[1]

[1]Department of Computer Science
Cornell University

[2]Department of Electrical and Computer Engineering
Cornell University

[3]School of Operations Research and Information Engineering
Cornell University

ICML Workshop on ML for Digital Education 2016

Motivation

- Spaced repetition software is becoming popular ($\sim 100,000$ users)
- Need models and formalizations that allow us to reason about their operation

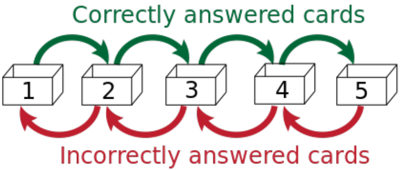
Outline

What is spaced repetition?

- Periodic, spaced review improves long-term retention
- Flashcard software uses spaced repetition
- **Key challenge**: review scheduling (which item should the user review at any given time?)
- **Goal**: ?
- **Constraints**: ?

Heuristic approach

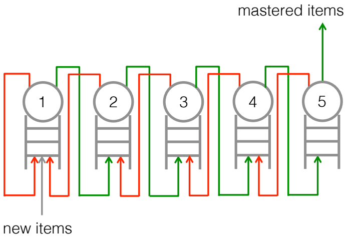
- **Leitner system**: review forgotten items more frequently than recalled items [Leitner, 1974]



- Very little formal understanding of how or why it should work
- Simple theoretical framework proposed by [Novikoff et al., 2012]

Formalizing the Leitner system using queueing theory

- Leitner Queue Network



- New items arrive over time. Mastered items exit the system.
- **Goal**: maximize throughput (the rate at which new items are mastered)
- **Constraints**: user has a review frequency budget (only has time to review U items per day)

Maximizing the rate of learning

deck review rates

throughput

$$\underset{\{\mu_k\}_{k=1}^{n}}{\text{Maximize}} \quad \lambda_{ext} \tag{1}$$

budget constraint

Subject to $U \geq \lambda_{ext} + \sum_{k=1}^{n} \mu_k,$

flow-balance constraint

$$\lambda_1 = \lambda_{ext} + (1 - P_1)\lambda_1 + (1 - P_2)\lambda_2,$$

$$\lambda_k = P_{k-1}\lambda_{k-1} + (1 - P_{k+1})\lambda_{k+1}, \text{ for } k \neq 1, n,$$

$$\lambda_n = P_{n-1}\lambda_{n-1},$$

stability constraint

$$0 \leq \lambda_k \leq \mu_k \quad \forall k \in [n],$$

routing probability

$$P_k = \frac{\mu_k - \lambda_k}{\mu_k - \lambda_k + \theta/k} \quad \forall k \in [n],$$

Maximizing the rate of learning

deck review rates

throughput

$$\underset{\{\mu_k\}_{k=1}^{n}}{\text{Maximize}} \quad \lambda_{ext} \tag{1}$$

budget constraint

$$\text{Subject to} \quad U \geq \lambda_{ext} + \sum_{k=1}^{n} \mu_k,$$

flow-balance constraint

$$\lambda_1 = \lambda_{ext} + (1 - P_1)\lambda_1 + (1 - P_2)\lambda_2,$$

$$\lambda_k = P_{k-1}\lambda_{k-1} + (1 - P_{k+1})\lambda_{k+1}, \text{ for } k \neq 1, n,$$

$$\lambda_n = P_{n-1}\lambda_{n-1},$$

stability constraint

$$0 \leq \lambda_k \leq \mu_k \quad \forall k \in [n],$$

routing probability

$$P_k = \frac{\mu_k - \lambda_k}{\mu_k - \lambda_k + \theta/k} \quad \forall k \in [n],$$

Maximizing the rate of learning

deck review rates

throughput

$$\underset{\{\mu_k\}_{k=1}^n}{\text{Maximize}} \quad \lambda_{ext} \tag{1}$$

budget constraint

$$\text{Subject to} \quad U \geq \lambda_{ext} + \sum_{k=1}^{n} \mu_k,$$

flow-balance constraint

$$\lambda_1 = \lambda_{ext} + (1 - P_1)\lambda_1 + (1 - P_2)\lambda_2,$$

$$\lambda_k = P_{k-1}\lambda_{k-1} + (1 - P_{k+1})\lambda_{k+1}, \text{for } k \neq 1, n,$$

$$\lambda_n = P_{n-1}\lambda_{n-1},$$

stability constraint

$$0 \leq \lambda_k \leq \mu_k \quad \forall k \in [n],$$

routing probability

$$P_k = \frac{\mu_k - \lambda_k}{\mu_k - \lambda_k + \theta/k} \quad \forall k \in [n],$$

Maximizing the rate of learning

deck review rates

throughput

$$\underset{\{\mu_k\}_{k=1}^{n}}{\text{Maximize}} \quad \lambda_{ext} \tag{1}$$

budget constraint

$$\text{Subject to} \quad U \geq \lambda_{ext} + \sum_{k=1}^{n} \mu_k,$$

flow-balance constraint

$$\lambda_1 = \lambda_{ext} + (1 - P_1)\lambda_1 + (1 - P_2)\lambda_2,$$

$$\lambda_k = P_{k-1}\lambda_{k-1} + (1 - P_{k+1})\lambda_{k+1}, \text{ for } k \neq 1, n,$$

$$\lambda_n = P_{n-1}\lambda_{n-1},$$

stability constraint

$$0 \leq \lambda_k \leq \mu_k \quad \forall k \in [n],$$

routing probability

$$P_k = \frac{\mu_k - \lambda_k}{\mu_k - \lambda_k + \theta/k} \quad \forall k \in [n],$$

Maximizing the rate of learning

deck review rates

throughput

$$\underset{\{\mu_k\}_{k=1}^n}{\text{Maximize}} \quad \lambda_{ext} \tag{1}$$

budget constraint

$$\text{Subject to} \quad U \geq \lambda_{ext} + \sum_{k=1}^{n} \mu_k,$$

flow-balance constraint

$$\lambda_1 = \lambda_{ext} + (1 - P_1)\lambda_1 + (1 - P_2)\lambda_2,$$

$$\lambda_k = P_{k-1}\lambda_{k-1} + (1 - P_{k+1})\lambda_{k+1}\ , \text{for } k \neq 1, n,$$

$$\lambda_n = P_{n-1}\lambda_{n-1},$$

stability constraint

$$0 \leq \lambda_k \leq \mu_k \quad \forall k \in [n],$$

routing probability

$$P_k = \frac{\mu_k - \lambda_k}{\mu_k - \lambda_k + \theta/k} \quad \forall k \in [n],$$

Maximizing the rate of learning

deck review rates

throughput

$$\underset{\{\mu_k\}_{k=1}^{n}}{\text{Maximize}} \quad \lambda_{ext} \tag{1}$$

budget constraint

$$\text{Subject to} \quad U \geq \lambda_{ext} + \sum_{k=1}^{n} \mu_k,$$

flow-balance constraint

$$\lambda_1 = \lambda_{ext} + (1 - P_1)\lambda_1 + (1 - P_2)\lambda_2,$$

$$\lambda_k = P_{k-1}\lambda_{k-1} + (1 - P_{k+1})\lambda_{k+1}, \text{ for } k \neq 1, n,$$

$$\lambda_n = P_{n-1}\lambda_{n-1},$$

stability constraint

$$0 \leq \lambda_k \leq \mu_k \quad \forall k \in [n],$$

routing probability

$$P_k = \frac{\mu_k - \lambda_k}{\mu_k - \lambda_k + \theta/k} \quad \forall k \in [n],$$

Can we model human memory?

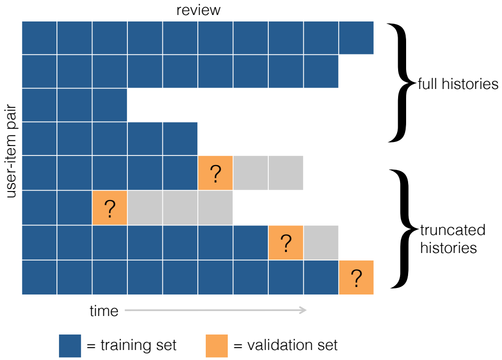
- Assume recall is binary. We would like to **predict the probability of recall**.
- Ebbinghaus proposed the **exponential forgetting curve** in the 19th century [Ebbinghaus, 1913]:

$$\mathbb{P}[\mathrm{recall}] = \exp\left(-\theta \cdot \frac{D}{s}\right),$$

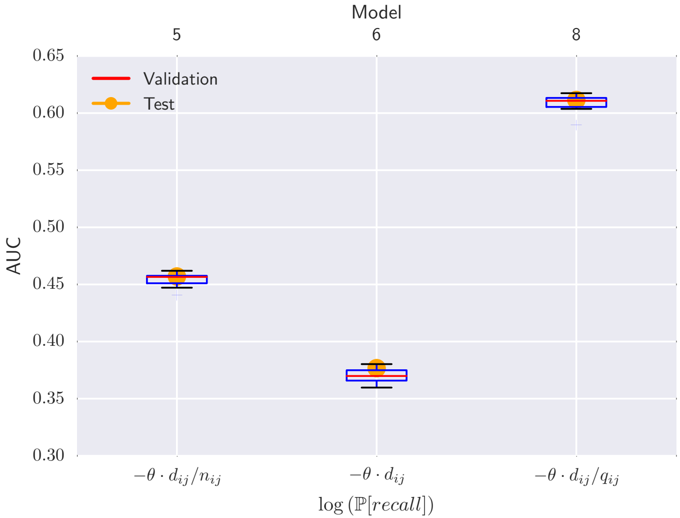
  where θ is the item difficulty, D is the delay since previous review, and s is the memory strength
- Memory strength s is not fully understood

Can we evaluate memory models?

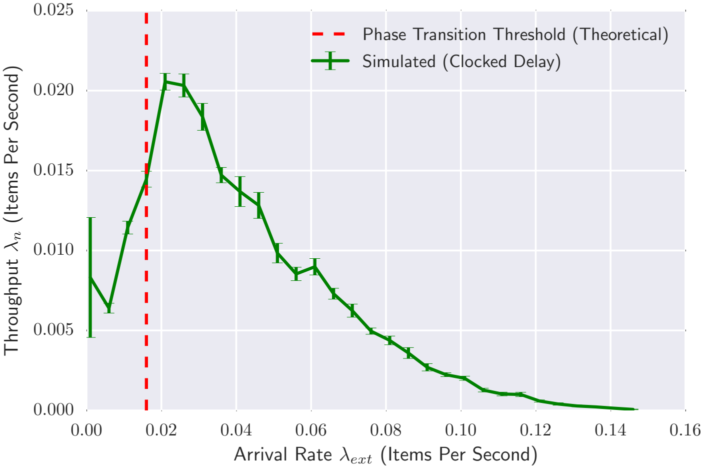
- Large-scale log data from Mnemosyne spaced repetition software [Bienstman, 2006]
- **Prediction task**: given historical outcomes, will the user recall or forget an item?



Results of memory model evaluations



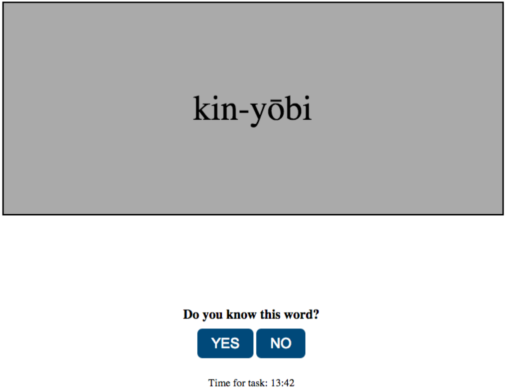
Simulations



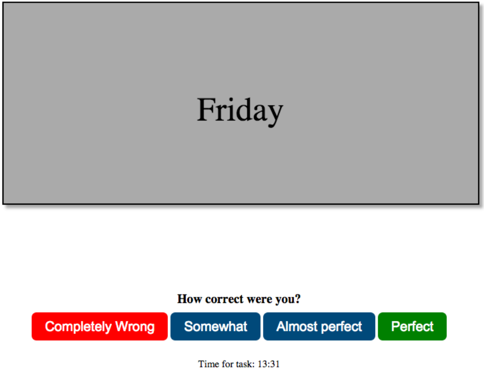
Controlled Experiment

- $>$ 300 users on Mechanical Turk [mtu, 2005]
- 15-minute sessions, working through flashcards on Japanese and Ameslan vocabulary
- Users are subjected to different arrival rates of new items
- Can we verify the existence of the phase transition predicted by the Leitner Queue Network?

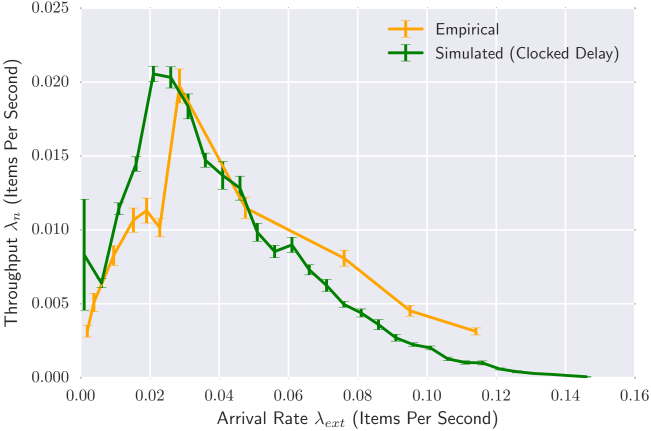
Controlled Experiment



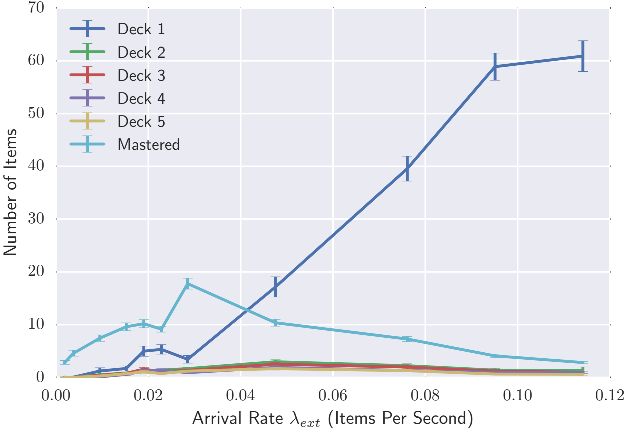
Controlled Experiment



Controlled Experiment



Controlled Experiment



Punch line

Don't procrastinate!

...but if you do, slow down your incoming workload

Ongoing work

Extensions to the Leitner Queue Network:

- Better approximations for routing probabilities
- More sophisticated memory models [Lindsey, 2014]
- Control policies for the transient regime
- Interventions for preventing positive-feedback instabilities

Contact:

- **Email**: sgr45@cornell.edu
- **Paper, Code, Data**: `http://siddharth.io/leitnerq`

References

(2005).
Amazon mechanical turk.
https://www.mturk.com.

Bienstman, P. (2006).
The mnemosyne project.
http://mnemosyne-proj.org.

Ebbinghaus, H. (1913).
Memory: A contribution to experimental psychology.
Number 3. University Microfilms.

Leitner, S. (1974).
So lernt man lernen.
Herder.

Lindsey, R. V. (2014).
Probabilistic models of student learning and forgetting.

Novikoff, T. P., Kleinberg, J. M., and Strogatz, S. H. (2012).
Education of a model student.
Proceedings of the National Academy of Sciences, 109(6):1868–1873.