Unbounded Human Learning: Optimal Scheduling for Spaced Repetition

Siddharth Reddy
Department of Computer Science
Cornell University
sgr45@cornell.edu

Igor Labutov
Electrical and Computer Engineering
Cornell University
iil4@cornell.edu

Siddhartha Banerjee
Operations Research and Information Engineering
Cornell University
sbanerjee@cornell.edu

Thorsten Joachims
Department of Computer Science
Cornell University
tj@cs.cornell.edu

ABSTRACT

In the study of human learning, there is broad evidence that our ability to retain a piece of information improves with repeated exposure, and that it decays with delay since the last exposure. This plays a crucial role in the design of educational software, leading to a trade-off between teaching new material and reviewing what has already been taught. A common way to balance this trade-off is via *spaced repetition* – using periodic review of content to improve long-term retention. Though widely used in practice, there is little formal understanding of the design of these systems. This paper addresses this gap. First, we mine log data from a spaced repetition system to establish the functional dependence of retention on reinforcement and delay. Second, based on this memory model, we develop a mathematical framework for spaced repetition systems using a queueing-network approach. This model formalizes the popular *Leitner Heuristic* for spaced repetition, providing the first rigorous and computationally tractable means of optimizing the review schedule. Finally, we empirically confirm the validity of our formal model via a Mechanical Turk experiment. In particular, we verify a key qualitative insight and prediction of our model – the existence of a sharp phase transition in learning outcomes upon increasing the rate of new item introductions.

Keywords

Spaced Repetition; Flashcard Scheduling; Queueing Models; Human Memory

1. INTRODUCTION

The ability to learn and retain a large number of new pieces of information is an essential component of human learning. Scientific theories of human memory, going all the way back to 1885 and the pioneering work of Ebbinghaus [9], identify two critical variables that determine the probability of recalling an item: *reinforcement*, i.e., repeated exposure to the item, and *delay*, i.e., time since the item was last recalled. Accordingly, scientists have long been proponents of the *spacing effect* for learning – the idea that periodic, spaced review of new information content improves their long-term retention.

A significant development in recent years has been a growing body of work that attempts to 'engineer' the process of human learning, creating tools that enhance the learning process by building on the scientific understanding of human memory. These schemes usually take the form of 'Flashcards' – small pieces of information content which are presented to the learner via a *spaced repetition* schedule [4], in order to combat the 'forgetting curve'. Though they have been around for a while in the physical form, a new generation of spaced repetition software such as SuperMemo [21], Anki [10], Mnemosyne [2], Pimsleur [19], and Duolingo [3] allow a much greater degree of control and monitoring of the process. Though these software applications are growing in popularity [4], there is a lack of formal mathematical models for reasoning about and optimizing these systems. In this work, we combine memory-models from psychology along with ideas from queueing theory to develop such a mathematical model for these systems – in particular, we focus on one of the simplest and most popular spaced repetition systems: the Leitner system [14] [1].

The Leitner system, first introduced in 1970, is a heuristic for prioritizing items for review. It is based on a series of decks of flashcards. After the user sees a new item for the first time, it enters the system at deck 1. The items at each deck form a first-in-first-out (FIFO) queue, and when the user requests an item to review, the system chooses a deck i according to some schedule, and presents the top item. If the user does not recall the item, the item is added to the bottom of deck $i-1$; else, it is added to the bottom

Preliminary work. Under review by the Conference on Knowledge Discovery and Data Mining (ACM SIGKDD). Do not distribute.

[1]The Leitner system actually emerges as the natural solution under our model – in a sense, our work helps explain its popularity in practice.

of deck $i+1$. The aim of the scheduler is to ensure that items from lower decks are reviewed more often than those from higher decks, so the user spends more time working on forgotten items and less time on recalled items. However existing schemes for assigning review frequencies to different decks are based on heuristics which are not founded on any formal reasoning, and hence, have no optimality guarantees. One of our main contributions is to provide a principled method for determining appropriate deck review frequencies.

The problem of deciding how frequently to review different decks in the Leitner system is a specific instance of the more general problem of *review scheduling* for spaced repetition software. The main tension in all settings is that schedules must balance competing priorities of introducing new items and reviewing old items, to ensure a maximum rate of learning. Most existing software use heuristics to make these trade-offs. We present a principled method for maximizing the rate of learning.

1.1 Related Work

The scientific literature on memory models is both old, dating back to more than a century, and also highly active. The basic memory model, the so-called *exponential forgetting curve*, was first studied by Ebbinghaus in 1885 [9] – it models the probability of recalling an item as an exponentially-decaying function of the time elapsed since previous review and the memory 'strength'. The exact nature of how strength evolves as a function of the number of reviews, length of review intervals, etc. is a topic of debate, though there is some consensus for the existence of a *spacing effect*, in which spaced reviews lead to greater strength than massed reviews (i.e., cramming) [8, 5]. Recent studies have proposed more sophisticated probabilistic models of learning and forgetting [18, 16], and there is a large body of related work on item response theory and knowledge tracing [15, 7]. Our work both contributes to this literature (via observational studies on log data from the Mnemosyne software system) and uses it as the basis for our queueing model and scheduling algorithm.

Though used extensively in practice (cf [4] for an excellent overview), there is very limited literature on the design of spaced repetition software. The seminal work in this regard is that of Novikoff et al. [17], who propose a theoretical framework for spaced repetition, based on a set of deterministic operations on an infinite string of content pieces. In particular, they assume identical items, and design schedules to implement deterministic spacing constraints, which are fixed in advance, based on an intuitive understanding of the effect of memory models on different learning objectives. The focus in [17] is on characterizing the combinatorial properties, in particular, the maximum asymptotic throughput, of schedules that implement various spacing constraints. Though our work shares the same spirit of formalizing the spaced repetition problem, we significantly improve upon their work in three ways: (1) in flexibility terms, as our model can easily incorporate various parameters such as the user's review frequency, non-identical item difficulties, different memory models, etc., (2) in computational terms, wherein, by using stochastic models and ideas from scheduling theory, we formulate optimization problems that are much easier to solve, and (3) in terms of scientific accuracy, as our work leverages both software log-data and large-scale experimentation to verify the memory models we use, and test the predictions made by our mathematical models.

1.2 Contributions and Argument Outline

The key contributions of this paper fall into two categories. First, the paper introduces a principled methodology for designing review scheduling systems with various learning objectives. Second, the models we develop provide qualitative insights and general principles for spaced repetition learning. To this effect, the argument in this paper consists of the following three steps:

- *Mining large-scale log data to identify human memory models*: First, we perform observational studies on data from Mnemosyne [2], a popular flashcard software, to compare different models of retention probability of items as a function of reinforcement and delay. Our results, presented in Section 2, add to the existing literature on memory models, and also provides the empirical foundation upon which we base our mathematical models.

- *Mathematical modeling of spaced-repetition systems*: Our main contribution lies in embedding the memory model empirically identified from the log-data analysis into a comprehensive stochastic model for spaced-repetition learning systems, and using this model to determine the optimal review schedule. Our framework, which we refer to as the *Leitner Queue Network*, is based on ideas from queueing theory and job scheduling. In particular, we leverage the theory of product-form networks [12, 6] to simplify the scheduling problem. Our Leitner Queue framework also allows us to study (both theoretically and via simulations) several related questions: the maximum rate of learning, the effect of item difficulties, the effect of a learner's review frequency on overall rate of learning, etc. We present our model, theory and simulations in Section 3.

- *Verification of mathematical model in controlled experiments*: Finally, we use Amazon Mechanical Turk [1] to perform large-scale experiments to test our mathematical models. In particular, we verify a critical qualitative prediction of our mathematical model – the existence of a *phase-transition* in learning-outcomes upon increasing the rate of introduction of new content beyond a maximum threshold. Our experimental results agree remarkably well with our model's predictions, thereby reaffirming the utility of our framework. Moreover, our experimental platform can also help serve as a testbed for future studies of other questions in spaced repetition systems. To this end, we release all data and software tools to replicate our experiments and facilitate follow-up studies. We document our experimental setup and results in Section 4.

2. MODELING HUMAN MEMORY

An essential component of any model of a spaced repetition system is to understand the effect of various system parameters on the ability of a human to recall a previously-seen item. For this, we adopt the widely accepted *exponential forgetting curve* model from the psychology literature. Roughly speaking, this model states that the retention probability decays exponentially, with a rate that increases with time elapsed since last review, and decreases with the 'strength' of knowledge of the item.

The increasing popularity of spaced repetition software affords us the chance of testing and verifying these memory models on log-data. In this section, we conduct such an analysis on log-data from the Mnemosyne [2] system.

Exponential Forgetting Curve.

The standard exponential forgetting curve model states that the probability of a user recalling an item obeys the functional form:

$$\mathbb{P}[\text{recall}] = \exp\left(-\theta \cdot d/s\right), \tag{1}$$

where $\theta \in \mathbb{R}^+$ is the item difficulty, $d \in \mathbb{R}^+$ is time elapsed since previous review, and $s \in \mathbb{R}^+$ is memory strength. Our formulation is slightly different from that of Ebbinghaus, in that we have added an explicit item difficulty parameter θ, which corresponds to the assumption that there is a constant, item-specific component of the memory decay rate. In the following, we explore how different instantiations of the exponential forgetting curve model fit empirical data. In particular, we explore the use of a global item difficulty θ vs. an item-specific difficulty θ_i, as well as several simple models of memory strength: a constant strength $s = 1$, a strength $s = n_{ij}$ equal to the number of repetitions of item i for user j (where $n_{ij} \geq 1$), and a strength $s = q_{ij}$ equal to the position of item i in the Leitner system for user j (where $q_{ij} \geq 1$). These results will form the basis of our scheduling algorithms.

Experiments on Log Data.

We use large-scale log data collected from Mnemosyne [2], a popular flashcard software tool, to validate our assumptions about the forgetting curve. After filtering out users and items with fewer than five interactions, we select a random subset of the data that contains 859,591 interactions, 2,742 users, and 88,892 items. Each interaction is annotated with a grade (on a 0-5 scale) that was self-reported by the user. Users are instructed by the Mnemosyne software to use a grade of 0 or 1 to indicate that they did not recall the item, and a grade of 2-5 to indicate that they did recall the item, with higher grades implying easier recall. We discretize grades into binary outcomes, where $1 = recalled$ and $0 = forgotten$, and observe an overall recall rate of 0.56 in the data. Additionally, we scale the time intervals between reviews to days.

We compare the exponential forgetting curve from Eq. (1) to three benchmark models: the zero- and one-parameter logistic item response theory models, and logistic regression. The 0PL IRT (users) model simply predicts the recall rate for the user observed in the training set, and the 0PL IRT (items) model simply predicts the recall rate for the item observed in the training set. The 1PL IRT model [15], a mathematical formulation of the Rasch cognitive model [20], has the following item recall likelihood: $\mathbb{P}[\text{recall}] = \phi(\theta_j - \beta_i)$ for user j and item i, where θ is user proficiency and β is item difficulty, and ϕ is the logistic link function. The logistic regression model uses the following statistics of the previous review intervals and outcomes to predict recall: mean, median, min, max, range, length, first, and last.

Logistic regression and 1PL IRT are trained using MAP estimation with an L2 penalty, where the regularization constant is selected to maximize log-likelihood on a validation set. All other models are trained using maximum-likelihood estimation (i.e., with an implicit uniform prior on model parameters). The hyperparameters in the IRT models are user abilities $\vec{\theta}$ and/or item difficulties $\vec{\beta}$. The hyperparameters in the exponential forgetting models are item-specific difficulties $\vec{\theta}$ or a global difficulty θ. We use ten-fold cross-validation to evaluate the memory models on the task of predicting held-out outcomes. Our performance metric is area under the ROC curve (AUC), which measures the discriminative ability of a binary classifier that assigns probabilities to class membership [11]. An advantage of using AUC instead of raw prediction accuracy is that AUC is not sensitive to class imbalance. On each fold, we train on the full histories of 90% of user-item pairs and the truncated histories of 10% of user-item pairs, and validate on the interactions immediately following the truncated histories. Truncations are made uniformly at random. See Figure 1 for an illustration of this scheme. After using cross-validation to perform model selection, we evaluate the models on a held-out test set of truncated user-item histories (20% of the user-item pairs in the complete data set) that was not visible during the earlier model selection phase.

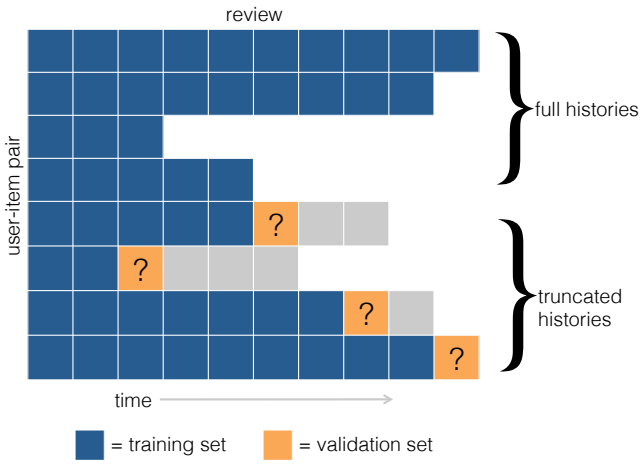

Figure 1: A schematic of the binary classification task used to evaluate memory models. Each square corresponds to a user-item interaction with a binary outcome. The gray squares are thrown out. This training-validation split occurs on each fold, where the sets of full and truncated histories change across folds.

Table 2 summarizes all the models that were evaluated, with rows 1-4 representing the benchmarks and rows 5-14 variants of the exponential forgetting curve model. Rows 5-9 use a global value for difficulty θ, while rows 10-14 use item-specific difficulties θ_i. Rows 5, 7, 10, and 12 assume that memory strength is equal to the number of reviews, while rows 8, 9, 13, and 14 assume that memory strength is equal to the position of the item in the Leitner system; rows 6 and 11 assume that memory strength is constant.

The predictive performance of the models on the test data is given in Figures 2 and 3. The results lead to four key findings:

1. Including the delay term in the recall model generally improves performance (model 5 vs. 7, 8 vs. 9, 10 vs. 12, 13 vs. 14); in Figure 2, compare the solid lines (models with the delay term) to the dashed lines (models without

	$\mathbb{P}$[recall]	Model
1	$\phi(\theta_j)$	0PL IRT (users)
2	$\phi(-\beta_i)$	0PL IRT (items)
3	$\phi(\theta_j - \beta_i)$	1PL IRT
4	$\phi(\beta \cdot x)$	Logistic regression
5	$\exp(-\theta \cdot d_{ij}/n_{ij})$	Exponential forgetting curve
6	$\exp(-\theta \cdot d_{ij})$	
7	$\exp(-\theta/n_{ij})$	
8	$\exp(-\theta \cdot d_{ij}/q_{ij})$	
9	$\exp(-\theta/q_{ij})$	
10	$\exp(-\theta_i \cdot d_{ij}/n_{ij})$	
11	$\exp(-\theta_i \cdot d_{ij})$	
12	$\exp(-\theta_i/n_{ij})$	
13	$\exp(-\theta_i \cdot d_{ij}/q_{ij})$	
14	$\exp(-\theta_i/q_{ij})$	

Table 1: In rows 5-14, q_{ij} is the position of item i in the Leitner system for user j, n_{ij} is the number of reviews of item i for user j, d_{ij} is the time elapsed since previous review of item i for user j, θ is the global item difficulty, and θ_i is the difficulty of item i. In rows 1-3, θ_j is the ability of user j and β_i is the difficulty of item i. In row 4, x is the feature vector of review interval and outcome statistics described earlier in this section, and β is a vector of coefficients. ϕ is the logistic function.

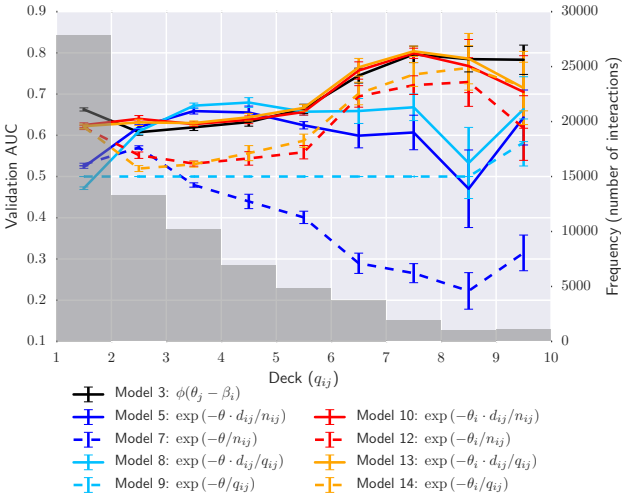

Figure 2: To evaluate the memory models' ability to predict outcomes in the Leitner system, validation AUC is computed for separate bins of data that control for an item's position q_{ij} in the Leitner system. The error bars show the standard error of validation AUC across the ten folds of cross-validation. Each curve corresponds to a row in Table 2. We have included only the best-performing benchmark model, 1PL IRT (model 3), to reduce clutter.

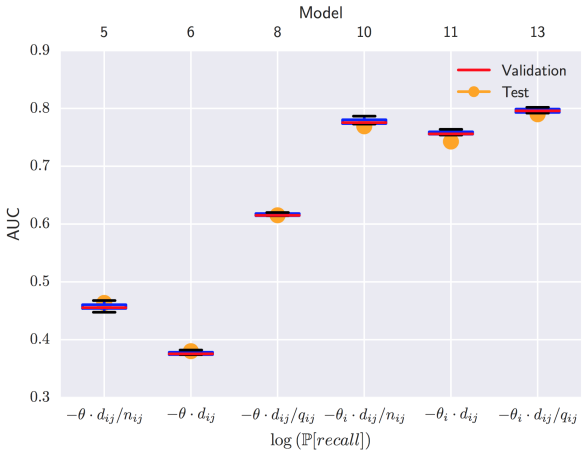

Figure 3: To compare the three memory strength models $s = n_{ij}$, $s = 1$, and $s = q_{ij}$, we compute AUCs for the full data set (instead of separate bins of data, as in Figure 2). The box-plots show the spread of validation AUC across the ten folds of cross-validation, and the orange circles show AUC on the test set.

the delay term).

2. The item-specific difficulty θ_i outperforms the global item difficulty θ for lower decks ($q_{ij} \leq 2$) and higher decks ($q_{ij} > 5$), but the global difficulty performs better for intermediate decks (model 5 vs. 10, 8 vs. 13); in Figure 2, compare the solid cool colors (models with the global item difficulty) to the solid warm colors (models with the item-specific difficulty).

3. The memory strength model $s = q_{ij}$ performs better than $s = n_{ij}$ and $s = 1$ (model 8 vs. 5-6, and 13 vs. 10-11); see Figure 3.

4. Exponential forgetting models that include the delay term (models 5, 8, 10, and 13) perform comparably to the best-performing benchmark model: 1PL IRT (model 3).

Our overall conclusion is that exponential forgetting curve models with delay term d and memory strength based on the deck in the Leitner system q_{ij} provide the best fit of the empirical data. Whether to use item-specific difficulties θ_i or a global difficulty θ is less clear. We will therefore consider both variants in the development of our overall theory of spaced-repetition learning systems in the following section.

3. A STOCHASTIC MODEL FOR SPACED REPETITION SYSTEMS

Based on the memory model developed in the previous section, we now present a stochastic model for a spaced repetition system, and outline how we can use it to design good review-scheduling policies.

We focus on a regime where the learner wants to memorize a very large set of items – in particular, the number of available items is much larger than the potential amount of time spent by the learner in memorizing them. A canonical example of such a setting is that of learning words in a foreign language. From a technical point of view, this translates to assuming that new items are always available for introduction into the system, similar to an *open queueing system* (i.e., one with a potentially infinite stream of arrivals) – moreover, we can use the *steady-state* performance

of the queue as a measure of performance of the scheduler as an appropriate metric in this setting. We refer to this as the *long-term learning* regime.

Since we found in Section 2 that the deck of the Leitner system is a good indicator of memory strength, we base our mathematical model on the Leitner system [14]. The Leitner system is one of the most commonly used spaced repetition systems. It comprises of a series of n decks of flashcards, indexed as $\{1, 2, \ldots, n\}$, where new items enter the system at deck 1, and items upon being reviewed either move up a deck if recalled correctly or down if forgotten. In principle, the deck numbers can extend in both directions; in practice however, they are bounded both below and above – we follow this convention and assume that items in deck 1 are *reflected* (i.e., they remain in deck 1 if they are incorrectly reviewed), and all items which are recalled at deck n (which in experiments we take to $n = 5$), are declared to be 'mastered' and removed from the system.

Following the results of Section 2, we use the following memory model in our mathematical analysis. For an item in Leitner deck i, and assuming that d units of time have elapsed since the item was last viewed, we model that it is recalled with probability:

$$\mathbb{P}[\text{recall}] = \exp\left(-\theta \cdot d/i\right), \tag{2}$$

where θ is the item difficulty parameter. For simplicity of exposition, we assume that the difficulty is a global constant (model 8 in Table 2), but we will relax this constraint later to allow for item-specific difficulties (model 13 in Table 2).

We note again that all existing schemes for assigning review frequencies to decks in the Leitner system, and in fact, in all other spaced repetition systems, are based on heuristics with no formal optimality guarantees. One of our main contributions is to provide a principled method for determining appropriate deck review frequencies. While we focus on the Leitner system in this paper, we conjecture that the techniques can be extended to other spaced repetition systems as well.

3.1 A Preliminary Queueing Model

We can model the set of items in various stages of an n-deck Leitner system via a network of n inter-connected queues, as depicted in Figure 4. New items are *pulled* into the system following a Poisson process with rate λ_{ext}, henceforth referred to as the *learning rate*. A new item is first shown to the user (who we assume does not know it beforehand, hence cannot recall it), then inserted into deck 1.

Our model of the Leitner system can be parametrized by the following state: at time t, we associate with each deck i the vector $S_i(t) = (Q_i(t), \{T_{i,1}(t), T_{i,2}(t), \ldots, T_{i,Q_i}(t)\})$, where Q_i is the number of items in the deck at time t, and $T_{i,j} < t$ is the time at which the j^{th} item in the deck first entered the system (note that the times are sorted). Next, we assume that the learner has a maximum review frequency budget (e.g., the maximum rate at which the user can review items) of U, which is to be divided between reviewing the decks as well as viewing new items. Formally, we assume that review instances are created following a Poisson process at rate U. Our aim is to design a scheduler which at each review instant chooses an item to review. When a deck is chosen for review, we assume that items are chosen from it according to FIFO. Note though that when an item comes up for review, its transition to the next state depends on the recall probability, which depends on the deck number and delay (i.e., time elapsed since the last review of that item). In particular, at time t, for any deck i, let $D_i = t - T_{i,1}$ denote the delay (i.e., the time elapsed since that item was last reviewed) of the head-of-the-line (HOL) item in deck i. Then, using the memory model from Equation 2, we get that upon reviewing the HOL item from deck i (for $i \in \{1, 2, 3, \ldots, n-1\}$), its transition follows:

$$\mathbb{P}[i \to i+1] = \exp\left(-\theta \cdot D_i/i\right)$$
$$\mathbb{P}[i \to \max\{i-1, 1\}] = 1 - \exp\left(-\theta \cdot D_i/i\right)$$

Note that items in deck 1 return to the same deck upon incorrect recall. Finally, items coming for review from deck n exit the system with probability $\exp\left(-\theta \cdot D_n/n\right)$ (i.e., upon correct recall), else transition to deck $n-1$.

Given any scheduling policy that depends only on the state $\mathbf{S}(t) = (S_1(t), S_2(t), \ldots, S_n(t))$, it can be easily verified that $\mathbf{S}(t)$ forms a Markov chain. Given this Markovian description of the system, the most general scheduler design problem is that of choosing a *dynamic state-dependent schedule*, wherein review instances are created following a Poisson process at rate U, and at each review instant, the scheduler defines a map from the state $\mathbf{S}(t)$ to a control decision which involves choosing either to introduce a new card, or a deck from which to review an item. Coming up with such an optimal dynamic schedule appears impossible in our setting owing to two reasons: (1) the state space of the system has *very high dimensionality* – in fact, it is infinite dimensional unless we impose some restriction on the maximum number of cards allowed in the network at any instant; and (2) the Markov chain over $\mathbf{S}(t)$ is *time-inhomogeneous*, as the transition probabilities change with t. In the next section, we show how we can modify this model via two assumptions in order to obtain a tractable program for optimizing the review schedule.

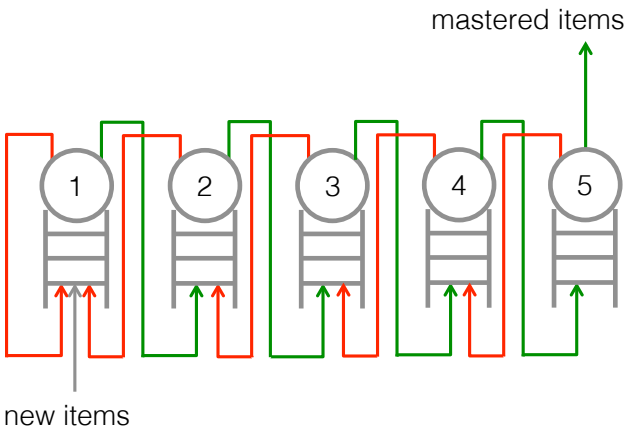

Figure 4: The Leitner Queue Network: Each queue represents a deck in the Leitner system. New items enter the network at deck 1. Green arrows indicate transitions that occur when an item is correctly recalled during review, and red arrows indicate transitions for incorrectly recalled items. Queue i is served (i.e., chosen for review) at a rate μ_i, and selects items for review in a FIFO manner.

3.2 The Leitner Queue Network

The stochastic model we created in Section 3.1 captured all the important details of the Leitner system – however, it was not amenable for use in designing a review schedule owing to its high dimensionality and time-inhomogenous nature. In this section, we propose an alternate stochastic model for the Leitner system, under which the problem of choosing an optimal review schedule *reduces to a low-dimensional deterministic optimization problem*. Nevertheless, simulation results in Section 3.3 demonstrate that the two models match closely.

Our new approximate model is based on two assumptions:

1. First, we reduce the dimensionality of the stochastic control problem by restricting ourselves to solving for optimal *static scheduling policies*. In particular, for each deck i, we choose a *service rate* μ_i, which represents the rate at which items from that deck come up for review. We now need to ensure that the learning rate λ_{ext} and deck service rates together satisfy the users review frequency budget constraint, i.e., $\lambda_{ext} + \sum_i \mu_i \leq U$ [2].

2. Switching to static schedules does not remove the issue of time-inhomogeneity of the chain. In order to convert this to a time-homogeneous model, for an item from deck i coming under review, *we replace the delay term D_i in the exponential memory model with its expectation* $\mathbb{E}[D_i]$:

$$\mathbb{P}[\text{Recall}|\text{ Deck } i] = \exp\left(-\theta \cdot \frac{\mathbb{E}[D_i]}{i}\right)$$

We henceforth call this the *mean-delay approximation*, and moreover, we refer to the queueing model with static scheduling policy and under the mean-delay approximation as the *Leitner Queue Network*.

To understand the use of the above assumptions, suppose we have that under an given choice of λ_{ext} and $\{\mu_i\}_{i=1}^n$, the resulting Markov chain is ergodic – then classical Markov chain theory dictates that the chain has a well-defined steady-state distribution. Moreover, in steady-state, for each deck i, the mean delay $\mathbb{E}[D_i]$ converges to a constant, and consequently, the probability of recalling an item from deck i also converges to a constant. Now using standard ideas from the theory of reversible Markov chains [12], we see that when the Leitner queue network is ergodic, then *its steady-state distribution is identical to that of a Jackson network of n $M/M/1$ queues Q_i* [13], with service times $\{\mu_i\}$, and arrival rates $\{\lambda_i\}$ satisfying the following *flow-balance equations*:

$$\begin{aligned}
\lambda_1 &= \lambda_{ext} + P_1\lambda_1 + (1-P_2)\lambda_2 \\
\lambda_i &= P_{i-1}\lambda_{i-1} + (1-P_{i+1})\lambda_{i+1} \quad \text{, for } i \in \{2,3,\ldots,n-1\} \\
\lambda_n &= P_{n-1}\lambda_{n-1} + (1-P_n)\lambda_n,
\end{aligned}$$

From basic properties of the $M/M/1$ queue, we have that:

$$\mathbb{E}[D_i] = \frac{1}{\mu_i - \lambda_i}$$

Finally, ergodicity of the Leitner queue network is guaranteed by the following stability conditions:

$$\lambda_i < \mu_i \quad \text{, } \forall i \in \{1,2,\ldots,n\}$$

[2] In practice, this corresponds to the following: for each review instance, with probability $\frac{\mu_i}{\lambda_{ext}+\sum_i \mu_i}$, we sample deck i, and select the item at the head of the sampled deck for review; else we introduce a new item.

If however one or more of these do not hold, then the resulting queue length(s) (i.e., deck sizes), and thus, the delays between reviews for that deck, grow unbounded for the decks for which the condition is violated.

Now we are interested in finding μ_i that maximize the *throughput* of the system (i.e., the long-term rate of mastering items – in an ergodic system in steady-state, this is equivalent to the learning rate λ_{ext}) such that the budget constraint is satisfied. Putting everything together, we get the following *static planning problem*:

$$\begin{aligned}
\underset{\{\mu_i\}_{i=1}^n}{\text{Maximize}} \quad & \lambda_{ext} \qquad (3) \\
\text{Subject to} \quad & U \geq \lambda_{ext} + \sum_{i=1}^n \mu_i, \\
& \lambda_1 = \lambda_{ext} + P_1\lambda_1 + (1-P_2)\lambda_2, \\
& \lambda_i = P_{i-1}\lambda_{i-1} + (1-P_{i+1})\lambda_{i+1} \text{ , for } i \neq 1,n, \\
& \lambda_n = P_{n-1}\lambda_{n-1} + (1-P_n)\lambda_n, \\
& P_i = \exp\left(\frac{-\theta}{i\cdot(\mu_i - \lambda_i)}\right) \quad \forall i \in [n], \\
& \lambda_i \leq \mu_i \quad \forall i \in [n], \\
& \mu_i \geq 0 \quad \forall i \in [n]
\end{aligned}$$

Thus, as desired, the Leitner Queue Network helps convert the stochastic control problem of designing an optimal review schedule, to a low ($O(n)$) dimensional, deterministic optimization problem. Now, we can use a nonlinear solver (e.g., IP-OPT) to solve the static planning problem. Note that this planning problem is unusual compared to typical network control problems, since the *routing probabilities depend on the service rates μ_i*.

Before proceeding to study various aspects of the optimal review schedule for the Leitner Queue Network (as calculated from (3)), we briefly comment on the effect of the mean-delay approximation. First, using Jensen's inequality, we have:

$$\mathbb{E}\left[\exp\left(-\theta \cdot \frac{D_i}{i}\right)\right] \geq \exp\left(-\theta \cdot \frac{\mathbb{E}[D_i]}{i}\right)$$

The above LHS however corresponds to the average recall probability (in steady-state) under the 'clocked' delay model. Moreover, a simple coupling argument suggests that using a lower bound on the expected recall probability will eventually give us a lower bound on the optimal throughput λ^*_{ext} (since decreasing the probability of items moving up through the network will lead to lower throughput). This argument suggests that *the throughput under the Leitner Queue Network is a lower bound on the throughput under the clocked delay model*. Moreover, simulations comparing the two models (cf. Figure 5) suggest that they are very close in practice.

3.3 Properties of the Leitner Queue Network

Given the Leitner Queue network, we now explore the properties of the resulting optimal review schedule.

The main qualitative prediction from our mode is the *existence of a phase transition* in learning outcomes when the arrival rate is increased beyond the rate of the optimal review schedule. To study this, we simulate a review session with 500 reviews and 50 unique items, under different learning rates. Binary outcomes are sampled according to the exponential forgetting curve (Model 8 in Table 2). In Figure 6, we observe that a sharp phase transition indeed occurs as

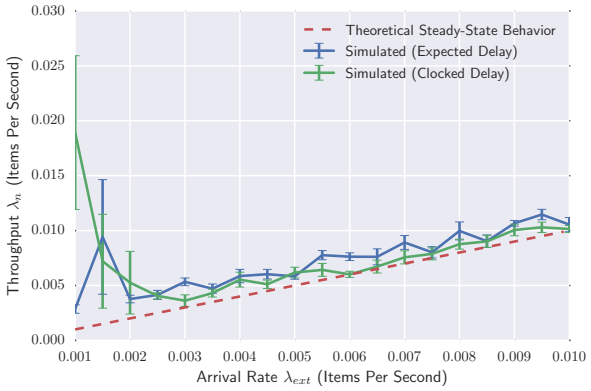

Figure 5: Exit rate λ_n vs. arrival rate λ_{ext}, where number of decks $n = 5$, review frequency budget $U = 0.1902$, and global item difficulty $\theta = 0.0077$. Transient behavior in the simulations causes the deviation between the simulated and steady-state throughputs.

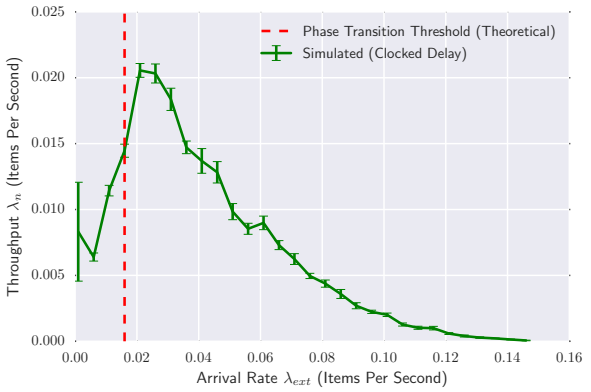

Figure 6: The exit rate λ_n vs. arrival rate λ_{ext}, where number of decks $n = 5$, review frequency budget $U = 0.1902$, and global item difficulty $\theta = 0.0077$.

the arrival rate is increased: throughput initially increases linearly with arrival rate, then sharply decreases. We note that the transition occurs at an arrival rate greater than the maximum rate predicted by the model – this arises dues to a combination of the fact that our predicted rate is based on a steady-state analysis and assumes that the retention probability depends on the average delay, while the simulated system uses clocked delays (which as we argued above is stochastically dominated by the Leitner queue network) and moreover is still in transient state.

In addition to the phase transition, we present several additional predictions made by the Leitner Queue Model and its optimal schedule. In Figure 7, we see that the Leitner Queue Model spends more time on lower decks than on higher decks. This is partly a result of the network topology, where items enter the system through deck 1 and exit the system through deck n. However, in Figure 8 we observe that the Leitner Queue Network also increases the expected delay between subsequent reviews as an item moves up through the system. We note here that there is empirical support in the literature for expanding intervals between repetitions [5]. The queue lengths in Figure 9 are the corresponding expected steady-state deck sizes.

The result in Figure 10 is interesting, because, to the best of our knowledge, there is currently little understanding of how the user should adjust deck review frequencies when the general difficulty of items changes. The Leitner Queue Network shows that when items are generally easy, the user should spend a roughly uniform amount of time on each deck. But when items are generally difficult, the user should spend more time on lower decks than higher decks. Figure 11 gives us a sense for how the maximum throughput of the system depends on the general difficulty of items.

In Figure 12 we study the maximum learning rate as a function of a user's review frequency budget U. The result is encouraging, as it suggests that the function is convex for small U, implying that there are *increasing* returns to throughput as the user increases their review frequency budget.

3.4 Extensions

The assumption that all items have the same difficulty θ can be easily relaxed by discretizing difficulties into a fixed number of bins b, creating b parallel copies of the network. Thus we now have b parallel Leitner queue network, coupled via the budget constraint which applies to the sum of service rates across the networks for each θ. We can assume that the θ_i are known a priori (e.g., from log data or expert annotations).

A complementary regime to long-term learning is *cramming* [17], wherein the number of items to be learnt is smaller than the total number of reviews the user does (for example, learning a syllabic/non-ideogram alphabet for a foreign language). This leads to a non-stationary queueing model, which is more challenging to optimize [12]. We have some ongoing parallel work that focuses on this setting.

4. EXPERIMENTAL VALIDATION

Amazon Mechanical Turk (Mturk) was used as platform to evaluate the fidelity of the queueing model in a realistic setting of memorizing vocabulary from a foreign language. In particular, the following study investigates whether the proposed queueing model captures the dynamics of memory sufficiently well to make verifiable predictions about the dynamics of the overall learning system, specifically the existence and the nature of the phase transition.

4.1 Experiment Setup

A total of 331 users (turkers) of the Mechanical Turk platform were solicited to participate in a vocabulary learning task. At the beginning of the task, vocabulary used in the task was selected randomly from one of two categories: *Japanese* (words) and *American Sign Language* (animated gestures). Items for the experiment were sampled from the list of common words in both languages [3]. Each task was timed to last 15 minutes, and turkers were compensated $1.00 for completing the task regardless of their performance. The task used the interface depicted in Figure

[3] https://en.wiktionary.org/wiki/Appendix:1000_Japanese_basic_words for *Japanese* and http://www.lifeprint.com/asl101/gifs-animated/ for *American Sign Language*

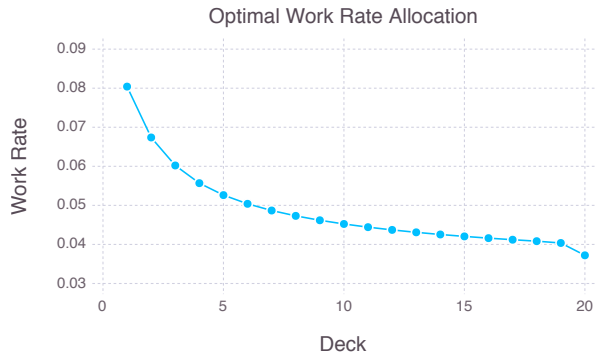

Figure 7: Optimal review schedule $\{\mu_i\}_i$ for $n = 20$, $U = 1$, $\theta = 0.01$

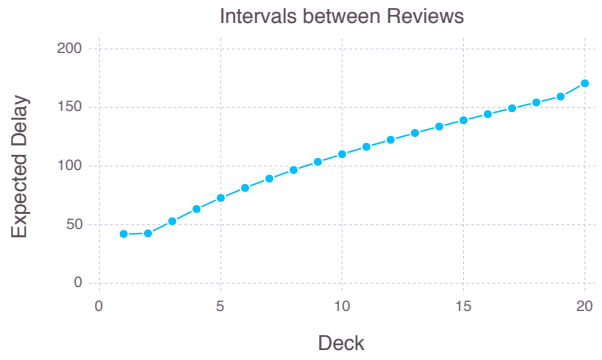

Figure 8: Expected delays $1/(\mu_i - \lambda_i)$ under optimal schedule for $n = 20$, $U = 1$, $\theta = 0.01$.

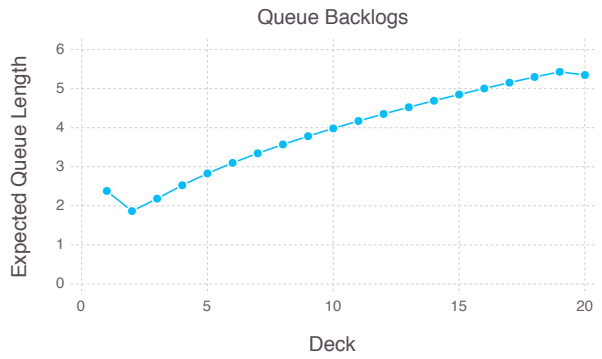

Figure 9: Expected queue size $\lambda_i/(\mu_i - \lambda_i)$ under optimal schedule for $n = 20$, $U = 1$, $\theta = 0.01$. The kinks at the boundaries, in this and the previous plot, arise from having a bounded number of decks.

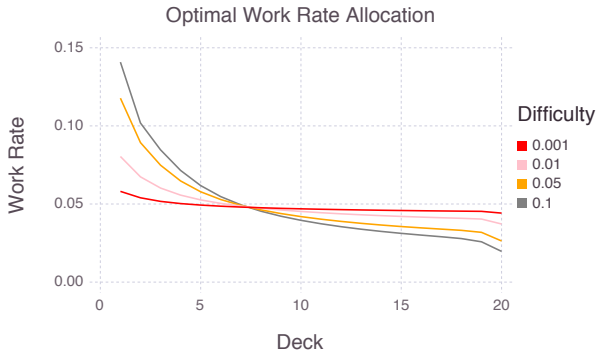

Figure 10: Optimal review schedule $\{\mu_i\}_i$ for $n = 20$, $U = 1$ and varying item difficulties θ.

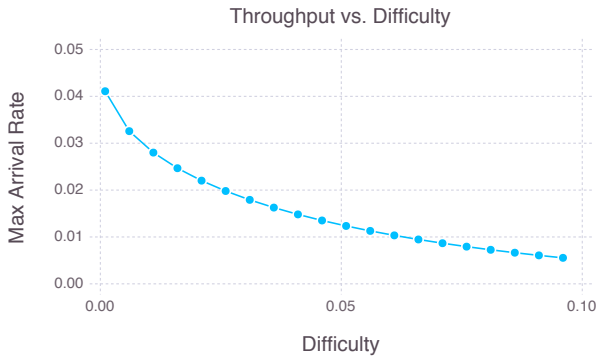

Figure 11: Variation in maximum learning rate λ^*_{ext} with item difficulty θ, for $n = 20, U = 1$.

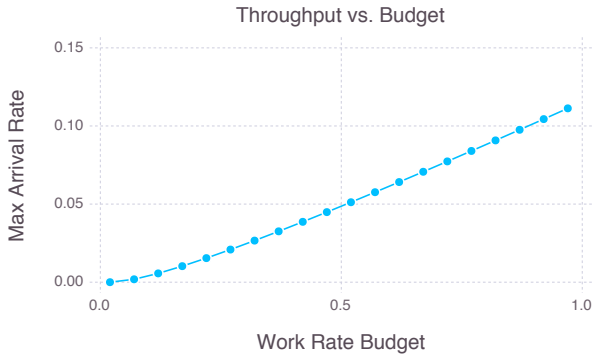

Figure 12: Variation in learning rate λ^*_{ext} with review frequency budget U for $n = 5, \theta = 0.01$. Note the convexity (hence, increasing returns) for low U.

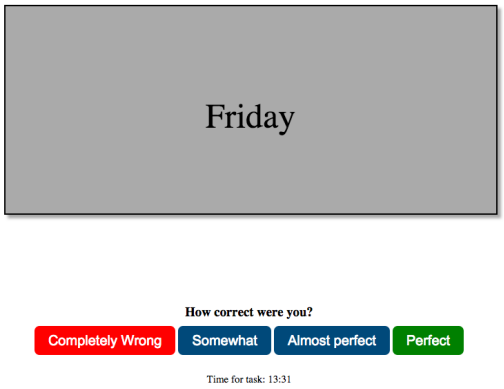

Figure 13: Screenshot of the Mechanical Turk interface for a single flashcard. The user sees the word in *Japanese* (not shown) and enters their guess. The card is then flipped (shown above) to reveal the meaning of the word in English. The user then assesses herself on a 4 point scale.

13: 1) a flashcard initially displaying the item in the foreign language (either word or gesture), and a pair of *YES/NO* buttons to collect input from the user in response to the question: *"Do you know this word?"*. If the user selected *YES*, he or she was then asked to type the translation of the word in English. Once the word was entered, the flashcard was "flipped" revealing the correct English word. The user was then asked to self-assess their correctness on a scale of 1 (completely wrong) to 4 (perfect). Following the submission of the rating, the next card is sampled from the deck. In all our experiments we consider self-assessment score of 3 (almost perfect) and 4 (perfect) as a "pass", and all other scores as a "fail".

At the beginning of each task, each turker was assigned to one of the 11 conditions corresponding to the arrival rate of new items (λ) (items per second): [0.002, 0.004, 0.010, 0.015, 0.020, 0.023, 0.029, 0.050, 0.076, 0.095, 0.11, 0.19]. The resulting data set consists of a total of 77,034 logs, 331 unique users, 446 unique items, an overall recall rate of 0.663, a fixed session duration of 15 minutes, and an average session length of 171 logs, where each log is a tuple (turkerID, itemID, score, timestamp). We heuristically set deck review rates to $\mu_i \propto 1/\sqrt{i}$, so that users review lower decks more frequently than higher decks (this heuristic roughly follows the shape of the optimal allocation in Figure 7). Note that we cannot optimize the review rates μ_i prior to the start of the experiments, since we do not know the item difficulty θ or the review budget U a priori, and do not yet have the data to estimate those parameters. During the experiments, we set the number of decks in the system to $n = \infty$, so items never exit the system during a review session. Items forgotten at deck 1 are 'reflected' and stay in deck 1. In our post-hoc analysis of the data, we consider an item to be 'mastered' if its final position is in a deck greater than $n = 5$.

We estimate the empirical review budget U using the formula (average number of logs in a session) / (session dura-

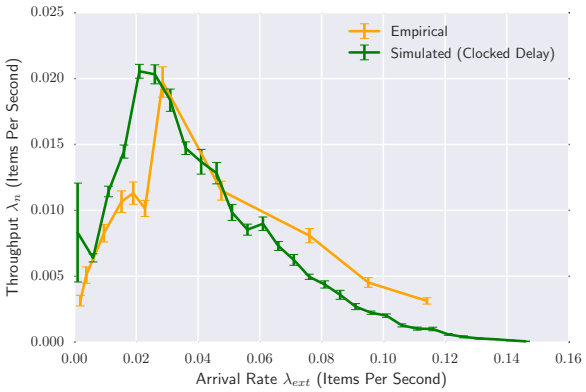

Figure 14: The exit rate λ_n vs. arrival rate λ_{ext}, where number of decks $n = 5$, review frequency budget $U = 0.1902$, and global item difficulty $\theta = 0.0077$.

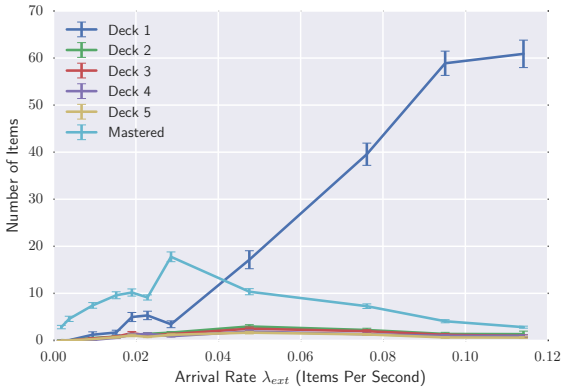

Figure 15: The number of items that finish in each deck vs. arrival rate λ_{ext}, where number of decks $n = 5$, review frequency budget $U = 0.1902$, and global item difficulty $\theta = 0.0077$.

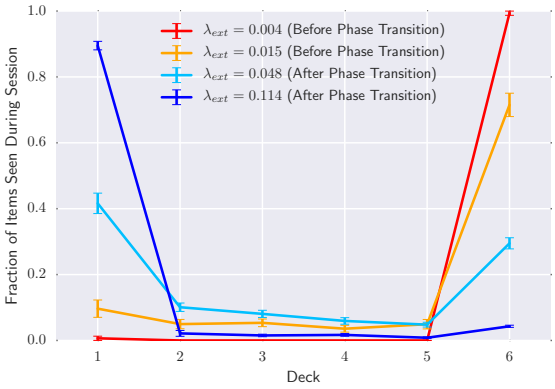

Figure 16: The fraction of items seen during a session that finish in each deck for different arrival rates λ_{ext}, where number of decks $n = 5$, review frequency budget $U = 0.1902$, and global item difficulty $\theta = 0.0077$. Deck 6 refers to the pile of mastered items.

tion), and the empirical item difficulty θ using maximum-likelihood estimation. We measure throughput λ_n as (average number of items mastered in a session) / (session duration).

4.2 Results

Figure 14 shows a plot of the average empirical throughput observed in the Mechanical Turk data for each arrival rate condition, along with a plot of the average throughput observed in simulations (using the parameter values for θ and U measured from the Mechanical Turk data). The simulations are identical to those presented in Figure 6: review sessions are simulated for 500 reviews and 50 items, and outcomes are sampled using the exponential forgetting curve (model 8 in Table 2). The simulated throughput fits closely with the empirical throughput. In particular, the Mechanical Turk data shows the phase transition in learning throughput that was predicted by the theoretical model.

In addition to computing the observed throughput for the various arrival rates in the Mechanical Turk data, we compute the average distribution of items across the five decks and the pile of mastered items at the end of a session. This gives insight into where items accumulate that are not mastered. Figure 15 illustrates the same phase transition observed earlier: as the arrival rate increases, we first see an increase in the number of mastered items. However, as the arrival rate increases past the optimum, relatively fewer items are mastered and relatively more items get 'stuck' in deck 1. Intuitively, the user gets overwhelmed by incoming items so that fewer and fewer items get reviewed often enough to achieve mastery. Our findings in Figures 15 and 16 match the typical behavior of a queueing network: for arrival rates higher than the queue network capacity, the number of items in one queue (deck 1) blows up while the other queues (decks 2-5) remain stable.

5. CONCLUSIONS

Our work develops the first formal mathematical model for reasoning about spaced repetition systems (in particular, for the the popular Leitner system) that is tractable for optimizing, and validated by empirical data. Our formalization suggests the maximum speed of learning as a natural design metric; using queueing theoretic techniques we derive a tractable program for optimizing the speed of learning. Additionally, the queueing framework opens doors to leveraging an extensive body of work in this area to develop more sophisticated extensions. To inspire and facilitate further research in this direction, we release (1) all model and evaluation code, (2) framework code for carrying out user studies, and (3) the data collected in our Mechanical Turk study. The code for running the experiments in Sections 2, 3, and 4 and the entire the Mechanical Turk data are available online at http://siddharth.io/leitnerq.

6. ACKNOWLEDGEMENTS

We thank Peter Bienstman and the Mnemosyne project for making their data set publicly available. This research was funded in part through NSF Awards IIS-1247637, IIS-1217686, IIS-1513692, the Cornell Presidential Research Scholars Program, and a grant from the John Templeton Foundation provided through the Metaknowledge Network.

7. REFERENCES

[1] Amazon mechanical turk. https://www.mturk.com, 2005.

[2] The mnemosyne project. http://mnemosyne-proj.org, 2006.

[3] Duolingo. https://www.duolingo.com, 2011.

[4] Spaced repetition. http://www.gwern.net/Spaced%20repetition, 2016.

[5] N. J. Cepeda, H. Pashler, E. Vul, J. T. Wixted, and D. Rohrer. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological bulletin*, 132(3):354, 2006.

[6] X. Chao, M. Pinedo, and M. Miyazawa. *Queueing networks: Customers, signals, and product form solutions*. Wiley, 1999.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[8] F. N. Dempster. Spacing effects and their implications for theory and practice. *Educational Psychology Review*, 1(4):309–330, 1989.

[9] H. Ebbinghaus. *Memory: A contribution to experimental psychology*. Number 3. University Microfilms, 1913.

[10] D. Elmes. Anki. http://ankisrs.net, 2015.

[11] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[12] F. P. Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

[13] D. G. Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, pages 338–354, 1953.

[14] S. Leitner. *So lernt man lernen*. Herder, 1974.

[15] W. Linden and R. K. Hambleton. Handbook of modern item response theory. *New York*, 1997.

[16] R. V. Lindsey, J. D. Shroyer, H. Pashler, and M. C. Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.

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

[18] H. Pashler, N. Cepeda, R. V. Lindsey, E. Vul, and M. C. Mozer. Predicting the optimal spacing of study: A multiscale context model of memory. In *Advances in neural information processing systems*, pages 1321–1329, 2009.

[19] P. Pimsleur. A memory schedule. *Modern Language Journal*, pages 73–75, 1967.

[20] G. Rasch. *Probabilistic models for some intelligence and attainment tests*. ERIC, 1993.

[21] P. Wozniak and E. J. Gorzelanczyk. Optimization of repetition spacing in the practice of learning. *Acta neurobiologiae experimentalis*, 54:59–59, 1994.