SCIENCE HUB REVIEW
2026
Volume 1
Issue 5
www.sciencehubreview.uz
ILMIY ELEKTRON JURNAL

# ADAPTIVE SPACED REPETITION: GENERATING INDIVIDUAL REVIEW SCHEDULES FROM STUDENT STUDY ACTIVITY

***Olimjonov Qamariddin Otabekovich***

***Student at OXUS University, Tashkent, 100204, Uzbekistan***

***Samandar Yusupov Anvarovich***

***Main teacher at OXUS University, Tashkent, 100208, Uzbekistan***

***Annotation***

*Spaced repetition systems such as SM-2 and FSRS schedule reviews effectively, but they rely largely on fixed rules or on the learner's own difficulty ratings, which can be subjective and inconsistent. This article proposes an adaptive spaced repetition framework that builds an individual review schedule for each student from study-activity data: response accuracy, response time, time since the last review and review history. For every student–item pair, the model estimates memory strength and item difficulty, updates both after each review, predicts the probability of recall with a logistic function, and sets the next review for the moment when this probability falls to a target level (for example, 0.90). The article describes the data, preprocessing steps, update rules and scheduling algorithm, and outlines a plan for evaluating the method against fixed-interval and SM-2 baselines. The method has not yet been tested on real student data, so the article presents a proposal rather than experimental results.*

***Keywords***

*spaced repetition, personalized learning, adaptive scheduling, forgetting curve, learning analytics, SM-2, FSRS, educational software.*

## Introduction

Retaining what has been learned is a basic problem in education. Human memory fades with time, a pattern first described by Ebbinghaus (1885) and confirmed in a later replication (Murre & Dros, 2015). Without review, newly learned material is lost quickly at first and then more slowly. Common study habits such as cramming or rereading make poor use of study time, and they ignore differences between learners, their prior knowledge and the difficulty of the material.

Spaced repetition addresses this problem by increasing the interval between successive reviews. Research on the spacing effect shows that study distributed over time is generally remembered better than the same study packed into one session (Cepeda et al., 2006), and that retrieving material from memory strengthens it more than rereading (Roediger & Karpicke, 2006). Many spaced repetition programs exist, but most use fixed rules or ask the learner to rate the difficulty of each item. Such ratings are subjective and inconsistent, and fixed rules cannot follow how an individual student actually learns.

This article proposes a way to generate individual review schedules from data that students produce anyway while studying: response accuracy, response time and the timing of their reviews. For each student and each item, the model estimates how well the item is remembered, predicts the probability of recall, and plans the next review for the moment when this probability falls to a chosen level. The article presents the theoretical basis, the data and features, and the scheduling algorithm, and then outlines how the method should be evaluated. It does not report experimental results.

**Literature review**

The theoretical basis of spaced repetition is the forgetting curve of Ebbinghaus (1885). The Leitner system (Leitner, 1972) was one of the first practical schemes: flashcards move between boxes that are reviewed at fixed intervals, and a wrong answer sends a card back to the first box. The SM-2 algorithm of the SuperMemo program (see Wozniak & Gorzelanczyk, 1994) sets intervals with an ease factor that is updated from the learner's own 0 to 5 rating of each answer. Model-based schedulers go further. Half-life regression, used at Duolingo, estimates the half-life of a memory from review history and item features (Settles & Meeder, 2016), and FSRS, a scheduler used in flashcard software, models memory stability and difficulty and builds on the work of Ye et al. (2022). These methods improve on fixed rules, but they mostly use recall outcomes and explicit ratings, and they rarely use finer behavioral signals such as response time.

Learning analytics offers such signals. It deals with the collection and analysis of data about learners and their environments in order to understand and improve learning (Siemens, 2013). Logs of time on task, error patterns and interaction sequences give a more objective picture of study behavior than self-ratings alone (Baker & Siemens, 2014).

A related line of work is knowledge tracing, which estimates the probability that a student has mastered a skill from the sequence of responses (Corbett & Anderson, 1995). It models mastery rather than the timing of review, so it complements scheduling methods but does not replace them.

The gap addressed here is the combination of the two lines of work. Forgetting differs between learners and between items, yet widely used schedulers rely on generalized rules or on subjective ratings. A scheduler that estimates an individual forgetting curve from observed behavior could place reviews more precisely. This article proposes one such design.

**Research methodology**

This section describes the proposed algorithm: the data it needs, how the data are prepared, and the model that turns them into review dates.

**Data sources and preprocessing**

The algorithm uses three kinds of data about student–item interactions:

1. **Interaction logs.** Timestamped records of each time a student works with an item: user ID, item ID, interaction type (first exposure, review, practice) and duration.

2. **Performance records.** A correctness indicator (correct or incorrect) for questions with a checkable answer, or a self-rating on a 1 to 5 scale for flashcard-style items.

3. **Item metadata.** Static properties of each item, such as an initial difficulty rating, a content category and the item format.

Before use, timestamps are converted to a common time base and then to the time elapsed since the previous interaction. Missing performance data are either replaced by a neutral value or the interaction is left out. Derived features are computed for each student–item pair: the number of earlier successful reviews, the average response time and the time since the item was last shown.

**Adaptive scheduling algorithm**

For each student u and item i the algorithm keeps two quantities. Memory strength $M_{u,i}$ is the current estimate of how well student u remembers item i; it starts at a baseline value for new items. Item difficulty $D_i$ describes how hard the item is, and it starts from the metadata rating and is refined with the results of all students. After each review the

student’s result is recorded as y, which equals 1 for a correct answer and 0 for an incorrect one. A graded value between 0 and 1 may be used instead, for example a lower value for a correct but slow answer.

The probability that student u recalls item i after an interval Δt since the last review is modeled with a logistic function:

$$P_{u,i} = 1 / (1 + \exp(-(k_1 M_{u,i} - k_2 D_i - k_3 \ln \Delta t))) \quad (1)$$

where $k_1$, $k_2$ and $k_3$ are positive coefficients that have to be estimated from data. Recall becomes more likely when memory strength is high, and less likely when the item is difficult or when a long time has passed.

After every review the two quantities are updated by a gradient step, as in online logistic regression:

$$M_{u,i} \leftarrow M_{u,i} + \alpha (y - P_{u,i}) \quad (2)$$

$$D_i \leftarrow D_i + \beta (P_{u,i} - y) \quad (3)$$

where α and β are learning rates. A correct answer that was unlikely raises memory strength and lowers the difficulty estimate, and an unexpected failure does the opposite.

The next review is set for the moment when the predicted probability of recall falls to a target level p* (for example, 0.90). Solving Equation (1) for Δt gives:

$$\Delta t_{next} = \exp((k_1 M_{u,i} - k_2 D_i - \ln(p^* / (1 - p^*))) / k_3) \quad (4)$$

The interval is rounded to whole days and kept between a minimum of one day and a maximum set by the course. In this way each item returns when it is predicted to be close to forgotten, and the schedule follows the individual forgetting curve of the student instead of a fixed pattern.

**Planned evaluation**

The method should be tested on real student data before it is used in practice. The proposed design compares it with two baselines, a fixed-interval schedule of the Leitner type and SM-2. Students would study the same material under each schedule, with participants assigned to schedules at random, and the schedules would be compared on three measures: retention in a test given after a delay, such as 30 days; the total number of reviews needed; and adherence, meaning the share of scheduled reviews that students

actually complete. The coefficients of the model would be estimated on one part of the data and checked on another.

The expectations for such a test are hypotheses, not findings: that retention under the adaptive schedule is at least as high as under SM-2, that it needs fewer reviews to reach the same retention, and that adherence is not lower.

**Conclusions and recommendations**

This article proposed an adaptive spaced repetition method that builds an individual review schedule from study-activity data. For every student–item pair it estimates memory strength and difficulty, predicts the probability of recall and sets the next review when that probability reaches a target level. The method uses signals that most learning platforms already record and does not depend only on self-ratings.

The main limitation is that the method has not been tested. Its assumptions, in particular the logistic form of the recall model and the update rules, need to be checked on real data. A new student has no history, so the first reviews would have to rely on item metadata and group averages until enough data are collected.

As a next step we recommend a pilot study with students of one course, following the plan above. Later work could compare the method with knowledge tracing and with deep learning models and add further signals, such as the time of day of study. In any implementation, learners should be told why an item is scheduled on a given day, so that the system is not seen as a black box. Only the data needed for scheduling should be collected, they should be anonymized where possible, and students should know how the data are used.

**References**

1. Baker, R. S., & Siemens, G. (2014). Educational data mining and learning analytics. In R. K. Sawyer (Ed.), The Cambridge Handbook of the Learning Sciences (2nd ed., pp. 253–274). Cambridge University Press.

2. Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. Psychological Bulletin, 132(3), 354–380.

3. Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4), 253–278.

4. Ebbinghaus, H. (1885). Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie. Leipzig: Duncker & Humblot.

5. Leitner, S. (1972). So lernt man lernen: Angewandte Lernpsychologie – ein Weg zum Erfolg. Freiburg: Herder.

6. Murre, J. M. J., & Dros, J. (2015). Replication and analysis of Ebbinghaus' forgetting curve. PLoS ONE, 10(7), e0120644.

7. Roediger, H. L., & Karpicke, J. D. (2006). Test-enhanced learning: Taking memory tests improves long-term retention. Psychological Science, 17(3), 249–255.

8. Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (pp. 1848–1858).

9. Siemens, G. (2013). Learning analytics: The emergence of a discipline. American Behavioral Scientist, 57(10), 1380–1400.

10. Wozniak, P. A., & Gorzelanczyk, E. J. (1994). Optimization of repetition spacing in the practice of learning. Acta Neurobiologiae Experimentalis, 54, 59–62.

11. Ye, J., Su, J., & Cao, Y. (2022). A stochastic shortest path algorithm for optimizing spaced repetition scheduling. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 4381–4390).