Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Motivation

- Students have variance in *trait ability*, that is internalized problem-solving ability, with respect to certain skills. In a classroom this trait ability is normally distributed.

- Targetting any segment of the distribution of students provides that segment with better treatment at the expense of other segments. Targetting the middle neglects the wings (both high-performing and low-performing).

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Motivation

- In addition, students may have different levels of ability with respect to different skills, and may have different interests as well. Accommodating all of these with a classical approach would seem to be impossible.

- Ideally, we would like to personalize teaching; but the amount of time required to do so is prohibitive. Thus we introduce ITS (intelligent tutoring systems) to help us.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Objective

Any educational program can be thought of as a sequence of tuples

$$(\chi_i, t_i)$$

which is an item scheduled at a time. An item could be a question or an informative content item. When placed in sequence, these form a schedule:

$$X = \langle (\chi_1, t_1), (\chi_2, t_2), \ldots (\chi_n, t_n) \rangle.$$

The present work seeks to answer the question: knowing the properties of these items, and having per-student information about the responses to these items at any point i, how should the remainder of the items be sequenced (or scheduled)?

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Novel Contributions

- The graph data structure to represent an assessment, whose nodes are items, and whose edges represent dependencies;

- A modification to an existing assessment theory known as Item Response Theory, which now accounts for dependency relationships;

- A scheduler, or algorithm whose purpose was to determine what the questions should be given the item parameters and the students' response sets;

- An addendum to an existing theory of memory, forgetting, and practice, which could then be integrated into the scheduler to provide a fuller-featured system.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Bloom's Taxonomy

- **Knowledge**. Recalling factual information. *What is a for-loop?*

- **Comprehension**. Assigning meaning to information. *What does the example for-loop output? (Give example.)*

- **Application**. Applying a rule to a specific instance. *How can the update statement of the loop be changed to print only even numbers?*

- **Analysis**. Breaking information into parts and exploring relationships. *What would happen if the update statement decremented instead of incremented the counter?*

- **Evaluation**. Judging the use of knowledge or the validity of an argument. *Which is better for reading user input: a for-loop or a while-loop? Why?*

- **Synthesis**. Utilizing knowledge to create a new solution to satisfy a goal. *Write a for-loop to print only even numbers up to ten.*

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Classical Test Theory

- In Classical Test Theory (CTT), we give a test on a concept or group of concepts, then grade it and obtain a distribution of scores.
- The letter grades are our rough estimates of ability. They don't always map to what we anticipated (90-A, 80-B, etc.)
- In that event, then based on the distribution, we map scores to grades.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Classical Test Theory

- One of the problems with CTT is that it does not innately account for varying levels of difficulty in questions. (The usual way to fix this is to make more difficult problems worth more, but it isn't clear by how much.)

- Also, the probability of guessing plays a role. A grade of 75% on a true/false test is not indicative.

- Some questions may not be relevant (correlated with overall score), and it may not be apparent until after testing.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Item Response Theory

In Item Response Theory:

- α is the item discrimination, or how well the item can distinguish students of varying trait ability;
- β is the question difficulty,
- γ is the probability of guessing the answer correctly,
- and θ is the *trait ability* of the student, or the student's particular ability to answer that question correctly.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Item Response Theory

The Item Response Theory formula for calculating the probability that a student will answer a question given item parameters and student ability:

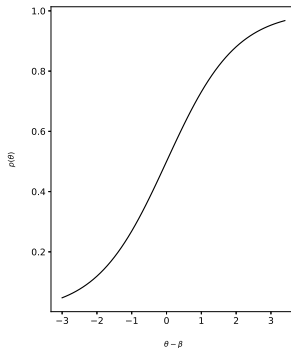$$p_i(\theta_s) = \gamma_i + \frac{1 - \gamma_i}{1 + e^{\alpha_i(\theta_s - \beta_i)}}$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Item Response Theory Curve

Figure: A probability curve in Item Response Theory.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Grading to Obtain Thetas

- We are interested in finding θ_s values for students, but not to grade (though we could).
- We need them primarily in order to determine questions to ask so as to raise trait ability.
- Nonetheless, knowing how θ_s might map to grades could make the process more intuitive.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Grade Mappings

Letter	θ	Explanation
F	-3.0	unsatisfactory
D-	-2.5	
D	-2.0	minimal
D+	-1.5	
C-	-1.0	
C	-0.5	acceptable
C+	0.0	
B-	0.5	
B	1.0	good
B+	1.5	
A-	2.0	
A	2.5	distinguished
A+	3.0	

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Maximum Likelihood Estimate

To find trait ability given a response set, define

$$f_{si}(\theta_s) = \begin{cases} p_{si}(\theta_s) & \text{if } x_{si} = 1 \\ q_{si}(\theta_s) & \text{otherwise} \end{cases}$$

where

$$q_{si}(\theta_s) = 1 - p_{si}(\theta_s).$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Maximum Likelihood Estimate

- That is, f_{si} assumes the probability p_{si} if answered correctly and q_{si} if not answered correctly.
- The probablity of observing a total response set given a particular θ_s value is the product of the probabilities f_{si} for all i, $1 \leq i \leq n$, or

$$\prod_{i=1}^{n} f_{si}(\theta_s).$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Maximum Likelihood Estimate

- We suppose that there exists some θ_s which maximizes this product.

- The most likely value for the student's true trait ability θ_s is defined by:

$$\theta_s = \underset{\theta}{\operatorname{argmax}} \left[\prod_{i=1}^{n} f_{si}(\theta). \right]$$

- That is, that value of θ_s which maximizes the product which gives the probability of all the observations occuring together, given θ.

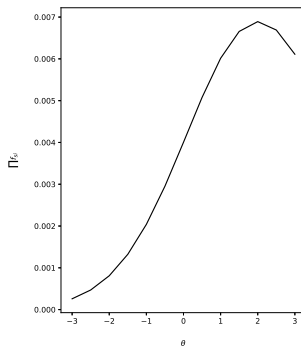- To obtain this, products for a range of possible θ values are calculated.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Item Response Theory MLE

Figure: A maximum likelihood estimation with Item Response Theory.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

The Database

- The database contains content items and assessments. Content items may be informative items or questions.

- Both have:
  - Bloom level
  - Difficulty
  - Concept
  - Dependencies

- Questions have:
  - Item discriminations
  - Probabilities of guessing
  - Solutions

- Assessments are collections of item sets, which may be represented as trees.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Introduction
Background
Previous Work

Difficulty and Bloom Level

- It is widely thought that Bloom level and difficulty are strictly equal [7, 8, 6, 5, 4].
- Previous work [2] has argued the contrary, producing empirical evidence for the hypothesis that Bloom level and difficulty are separate entities.
- It is sufficient to find pairs of problems to serve as counterexamples: one at a higher Bloom level and lower difficulty than another.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

There are Multiple Abilities

- A student may have differing levels of ability for different concepts: good at recursion, but with pointers.
- For that reason, trait ability should be separated per-concept.
- However, a student may have different levels of ability per Bloom level.
- For example:
  - A student may have exceptional comprehension of recursion. The student can follow a recursive procedure.
  - However, the same student may have low application ability; they cannot yet apply the rules of recursion to complete a fill-in-the-blank code.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Trait Ability Matrix

- Since trait ability is per-student and per-Bloom level, the most representative data structure is a matrix:

$$
\Theta_s = \begin{bmatrix}
\theta_{s11} & \cdots & \cdots & \cdots & \theta_{sn1} \\
\vdots & \ddots & & & \\
\vdots & & \theta_{sjk} & & \\
\vdots & & & \ddots & \\
\theta_{s1m} & & & & \theta_{snm}
\end{bmatrix}
$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

An Example Trait Ability Matrix

- A likely matrix may look like the following, which has a diagonal gradient.
- If the course follows a progression across Bloom levels and concepts, the student's ability per-concept and per-Bloom level should rise accordingly over time.

$$\Theta_1 = \begin{bmatrix} 3 & 2.5 & 1 & 0 & -1 & -2 \\ 2 & 1.5 & 0 & -.5 & -1.5 & -2.5 \\ 1 & .5 & 0 & -1 & -2 & -3 \\ 1 & 0 & -.5 & -1 & -2.5 & -3 \end{bmatrix}$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Question Dependencies

- Some questions have dependencies. Consider the following; one must know about integer division and the modulo operations before being able to combine them to solve problem (3).

1. *What is (5 % 2)?*
2. *What is (5 / 2)?*
3. *What is (5 % (5 / 2))?*

- It is reasonable to assume that the student's ability to handle a a dependency (or a dependee) influences the ability to handle the question which depends on the dependencies (the depender).

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Identifying Potential Dependencies

- If dependencies of the form $c \rightarrow a$ and $c \rightarrow b$ exist (that is, c depends on a and b), they may be indicated by a high simple matching coefficient:

$$\frac{n_{00} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

where the values for these various n are:

	$y = 0$	$y = 1$	total
$x = 0$	n_{00}	n_{01}	$n_{0\bullet}$
$x = 1$	n_{10}	n_{11}	$n_{1\bullet}$
total	$n_{\bullet 0}$	$n_{\bullet 1}$	n

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Identifying Dependencies

- Alternatively, the phi coefficient, or mean square contingency coefficient, can be used to identify the degree of association between two questions. It is defined as:

$$\phi = \frac{n_{11} n_{00} - n_{01} n_{10}}{\sqrt{n_{\bullet 0} n_{\bullet 1} n_{0 \bullet} n_{1 \bullet}}}$$

- This is the binary analogue to the Pearson correlation coefficient; but its range is only $[-1, 1]$ if there is a fifty-fifty split.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Assessments are Forests
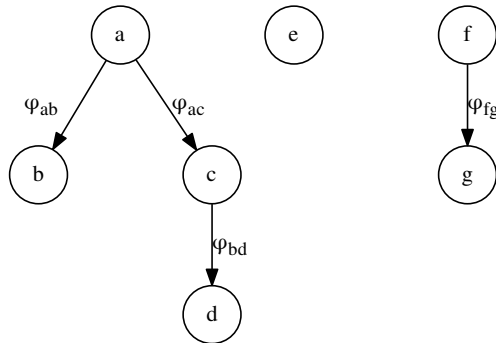
Figure: A forest obtained from an item set, where each tree is a subset of the item set which has dependency relationships

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Some "Dependencies" are Not Dependencies
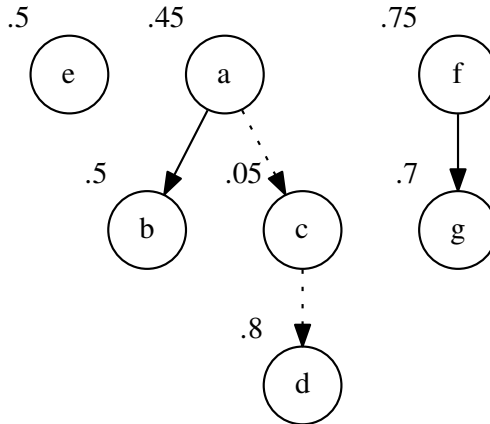
Figure: The severance of dependency relationships based upon low α values

Preliminaries
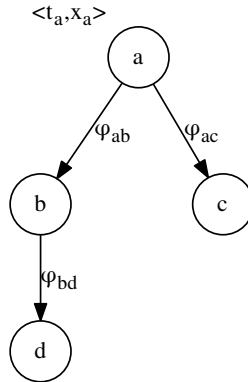**Models and Data Structures**
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models
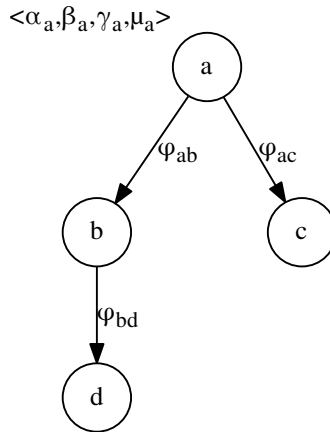
The Base Graph

Figure: The base item set graph, which includes item-specific paramters

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Student Graph

Figure: The student-specific item set graph, which includes the list of student responses and timestamps for each response

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Linear Regression

- We might consider the use of linear regression to determine whether or not a student will answer a question correctly (Y) given responses for X_1, X_2, etc.

- Consider the linear equation in which a binary dependent variable Y depends on some response X:

$$Y = a + bX + \epsilon$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

We Shouldn't Use Linear Regression

- Unfortunately, our particular scenario violates a few of the assumptions required for linear regression.

- There is no linear relationship since our dependent variable is categorical (0 or 1).

- Every linear combination of components should have a univariate normal distribution, but every component only assumes one of two possible values.

- Linear regression requires homoscedasticity. The error terms along the regression should be equal, but in the above situation, the variance of the error is dependent on the probability. In particular $\mathrm{var}(\epsilon) = p(1 - p)$.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Assumptions of Logistic Regression

- The dependent variable is binary,
- P(Y=1) is the probability that the event Y occurs,
- The model is fitted correctly, which means that there are no extraneous variables used in the regression, but that all variables are meaningful,
- That error terms should be independent; each observation should be independent, and little to no multicollinearity should exist.
- Indepedent variables should be linearly related to the log odds of the event.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Possible Violations

- The user can eliminate extraneous variables by examining α, and specifies the graph from the outset, so there *shouldn't* be extraneous variables.

- Some multicollinearity will probably exist for most dependency sets due to the nature of the dependee-depender relationships.

- An extraction technique like PCA or factor analysis could be done to avoid multicollinearity.
  - ...But then the graphs would not be as legible.
  - Plus, we would need to do PCA/FA every time the question is answered, which can be expensive.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Logistic Regression: The Formulae

Logistic regression is a regression of what are known as log odds. The odds are defined as:

$$\text{odds} = \frac{p}{1-p}$$

And the log odds, or logit, is defined as:

$$\text{logit}(y) = \ln(\text{odds}) = \ln\left(\frac{p}{1-p}\right) = a + bX$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Logistic Curve and Item Response Theory

Correspondingly, odds may be defined as follows:

$$\frac{p}{1-p} = e^{a+bx}$$

Solving this equation for p yields

$$p = \frac{e^{a+bx}}{1 + e^{a+bx}}$$

which can be reduced to

$$p = \frac{1}{1 + e^{-(a+bx)}}$$

which is called the logistic curve, which is one in the family of sigmoid curves. It is identical to the type of curve used in Item Response Theory.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Inclusion of Dependency Information

- To account for item dependencies, a modified form of logistic regression may be sought.
- It is desirable to keep in account the item discrimination as well as the trait ability of the student.
- Here, X_i is the correctness of the ith dependee response, and Y is the log odds that the depender question is answered correctly:

$$Y = b_1 X_1 + b_2 X_2 + \ldots + b_{n-1} X_{n-1} + b_n \alpha_i (\theta_s - \beta_i)$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models
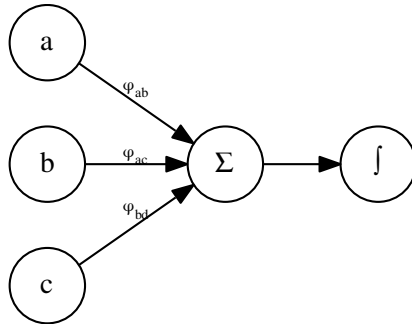
A Neural Network

Figure: A perceptron; a single-layer neural network, where the inputs a, b, and c are multiplied by weights, summed, and applied to a sigmoid squash function

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Dependency Graph with Weights
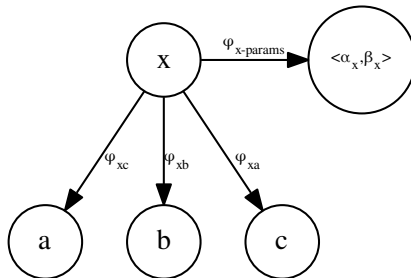
Figure: A view of the dependency graph and weights used in the logistic regression, including the item parameters

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Law of Forgetting

- Ebbinghaus is credited with a theory of memory and forgetting which has withstood empirical study for over a century [3], known as the power law of forgetting.

- According to it, the strength of a memory after a time t falls off exponentially:

$$S(t) = ae^{-bt}$$

- In this model, a is the initial strength of the memory, and b^{-1} is a decay rate. If $a = 1$, the function may be interpreted as a probability function:

$$p_{recall}(t) = e^{-bt}$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Curve of Forgetting
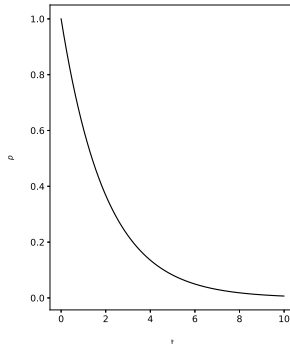
Figure: The Ebbinghaus curve of forgetting, which features an exponential dropoff of memory strength over time

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Modified Curve of Forgetting
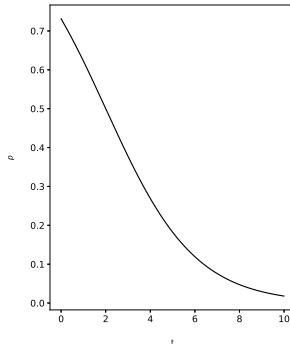
Figure: The modified curve of forgetting, which resembles the reverse sigmoid function

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

ACT-R

- John Anderson is credited with having developed Adaptive Control of Thought-Rational (ACT-R), a process-based model which simulates the solving of problems.

- In ACT-R, there are goals, akin to problem statements; and rules, or processes used to solve problems; and finally facts, or knowledge utilized in the course of applying rules.

- In addition to this, however, Anderson added models for memory and forgetting to support realistic recall probabilities and latencies, some of which inspire the present work.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Anderson's Activation via Related Concepts

- According to Anderson's model, a chunk of memory i is re-activated (or additionally activated) to the extent that other chunks of information (related concepts, words, ideas, etc.) which have some association to i are attended to.

- This notion is captured in the following equation [1]:

$$a_i = b_i + \sum_{i=1}^{n} w_j s_{ji}$$

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Effect of Practice

- Practice has the effect of causing the base strength of the memory to increase, and delays cause the strength of the memory to drop off [1]:

$$b_i = \ln\left(\sum_{j=1}^{n} t_j^{-d} \right)$$

- Here, t_j is the time since the jth practice of an item, and d is a decay rate.
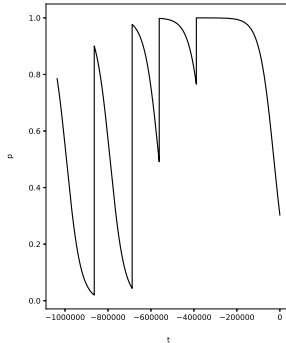
Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Forgetting with Re-Activation

Figure: Forgetting with re-activation; each spike in the graph is an additional trial where the student is exposed to the item again

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

A Modification to the Memory Models

- A slight modification to this theory accounts for short-term memory and short-term memorization.
- These allow for a small time window for the student to enjoy a high probability of recollection before dropping off sharply, as in the original curve:

$$p_{recall}(t) = \frac{1}{1 + e^{m(t-\lambda)}}$$

- In this equation, λ is the lifespan of the memory; or, the amount of time that passes until there remains only a .5 probability that the student recalls the information.
- The value m is a parameter which controls the rate of dropoff, much like the decay rate in Ebbinghaus' model.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Re-Activation

- To account for re-activation, a simple model for the extension of half-life may be used:

$$\lambda_n = \rho_s \lambda_{n-1}$$

- Here, n refers to exposure or trial number n. In the ITS, this is the nth time that the student has seen the problem.

- λ_{n-1} is the former lifespan of the memory. ρ_s is a learning rate, which is a parameter particular to the student; its domain is $(1, \infty]$.

- The intuition captured by this formula is that with an increased number of trials, the lifespan of the memory increases.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Forgettability

- In addition, there is a difference in problems in the ease with which they are learned. An addendum to this can be used to account for individual differences in problems:

$$\lambda_n = \mu_i \rho_s \lambda_{n-1}$$

- Here, μ_i represents the memorability of the problem, or the ease with which the problem solution can be committed to memory.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Spacing Effect

- The spacing effect is the effect that the amount of time in between trials has on the memorization of a chunk of memory.
- In the above model, memorization is interpreted as an increase in the lifespan of a memory. If only a short amount of time passes between the last trial, the effect will not be as great as if a longer time has passed.
- One consequence of this is that, according to the spacing effect hypothesis, cramming is ineffective (where cramming is namely repeating trials in short bursts).
- The spacing effect can be accommodated in the memory model used by the intelligent tutoring system. We define a function for the dropoff:

$$\sigma_t = (1 - e^{-at})$$
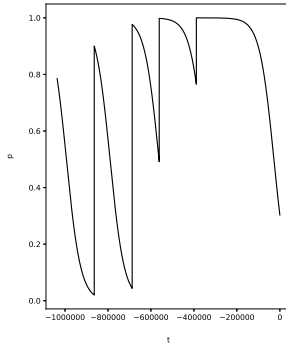
Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Forgetting with Re-Activation

Figure: Forgetting with re-activation; each spike in the graph is an additional trial where the student is exposed to the item again

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

The Lifespans of Memories

- This function indicates the extent to which the spacing from the time the item was last seen influences the increase in the lifespan of the memory:

$$\lambda_n = (1 + \sigma_t \mu_i \rho_s)\lambda_{n-1}$$

- The utility of this model is in assessing the probability with which a student answers a question; not only based on trait ability and dependency relationships, but also on the inherent tendency to forget information with the passage of time.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models
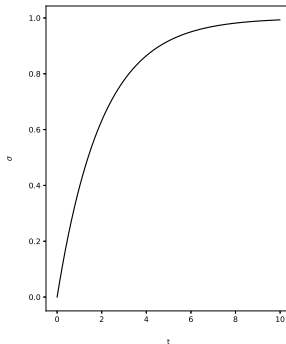
Learning Curve

Figure: The learning curve, which indicates the extent of memory lifespan increase given a time between trials

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Remember or Re-Solve

- It will be assumed that, if a student has been exposed to a item before, then the probability of being able to answer the item correctly may assume one of two values.

- The first is based upon recollection; it is the probability of recalling the facts, processes, and so forth required to produce the solution for an item.

- The second is based upon derivation of the solution from known facts, processes, and so forth in the dependencies, as if the student were answering the question for the first time.

Preliminaries
Models and Data Structures
Algorithms for Scheduling

Representing Trait Ability
Representing Dependencies
Logistic Regression
Memory Models

Remember or Re-Solve

- That is, if a student does not recall the process for solving a problem, the probability defaults to the probability based upon item parameters, trait ability and dependency relationships:

$$p = \begin{cases} p_{recall}(t) & \text{if } p_{recall} > p(\theta_s, x_1, \ldots) \\ p(\theta_s, x_1, \ldots) & \text{otherwise} \end{cases}$$

Initialization

- First, at the very beginning of the program, the trait ability matrix for the student is initialized:

$$\Theta \leftarrow -3$$

Preliminary Testing

- The first iteration schedules a preliminary test consisting of a
  diagonal block of the trait ability matrix, for example:

$$
\Theta_s = \begin{bmatrix} \theta_{s11} & \theta_{s21} & \theta_{s31} & \dots \\ \theta_{s12} & \theta_{s22} & & \\ \theta_{s13} & & \ddots & \\ \vdots & & & \end{bmatrix}
$$

- The test should be small enough to be feasible while having
  enough questions to support an MLE. At least two questions
  are needed per (Bloom $\times$ concept) for an MLE.

Question Pruning

- With respect to α_i, it is desirable to discard any question i for which $\alpha_i \leq 0$. It should be severed from the dependency tree.
- If α_i is close to -1, then the question may have predictive power.
- With the response set, Θ_s can be constructed and the proximal zone of development can be identified.

How to Select Categories

- Categories with $\theta_{sjk} = 0$ are areas where the student has a roughly .5 probability of answering a question of difficulty $\beta = 0$ correctly.
- Knowing what the trait ability matrix looks like, what categories should be selected?

Too Easy: Don't Bother

- The higher θ_{sjk} values should be left alone, particularly those nearing 3, since this demonstrates exceptional mastery of that category.

- In particular, if $\theta_{sjk} = 3$, there is no reason to ask questions from that category, since trait ability is capped at 3.

- As will be discussed later, there may be precedent to ask such a question if asking it will lead to an increase in the probability of answering other questions.

Too Hard: The Potential for Cruelty

- It would be unfair (perhaps even cruel), to ask questions for which $\theta_{sjk} - \beta > 1$; that is, those questions for which the student has a less than half chance of answering correctly.

- Asking such questions consistently could have psychological ramifications!

- In the course of ordinary non-formative instruction, such questions may be asked, but in this case the schedule may be personalized and the probabilities of success are known.

Just Right: Where p .5

- The difficulty matrix for the student may be

$$B_s = \Theta_s - \delta$$

where δ is some (small) number. The smaller the number, the harder the questions relative to the student's ability; but also the fewer that the student must answer to raise their trait ability estimate.

- Ultimately, selecting the neighborhood of difficulties to include questions from is a personal decision.

Initial Calculations

- When a student is assigned an assessment, first the probabilities that the student will be able to answer each question are recursively calculated.
- They must be calculated from the bottom-up, since internal nodes require probability estimates from their children.
- The leaves are the only nodes that do not require logistic regression to obtain a probability estimate, since the leaves have no dependencies.
- Probability estimates for the children are obtained using Item Response Theory.

Initial Recalls

- For each node which has been answered, the probability of recall can be calculated using the equation for recall.
- In the presence of a probability of recall, the final probability of answering the question can be calculated using the combined formula.

Finding a Question

- Then, a recursive traversal beings at the root of each tree. If the probability of answering the question at the root is within the neighborhood of $[.5, .5+\delta]$, it is asked.

- Recall that low probabilities correspond to higher differences in $\beta - \theta$; thus if a probability is very low, yet the student answers the question correctly, it will indicate a corresponding rise in trait ability.

- Such items have higher impact on the MLE estimate of trait ability.

- Yet if the probability is below .5, there is no reasonable assurance the student will be able to answer it to begin with.

p Too High, p Too Low

- If the probability to answer the question correctly is in the neighborhood of $(.5+\delta, 1]$, the question is skipped. The algorithm may recursively descend into its dependencies to find lower-p items.
- If the probability is less than .5, then the system will seek to raise the probability by targetting the dependees, in order of the highest-impact dependees to the lowest-impact.

Findin the Highest-Impact Dependency

- The dependee which controls the highest amount of variance in the parent probability is that dependee which has the highest coefficient in the logistic regression model.
- However the student may already have a high probability of answering that dependee, such that there is little room for increases in the probability in the parent node due to answering the child node question correctly.
- Supposing p_i is the probability of answering the ith dependee correctly, then the amount which can be gained in the dependee is:

$$\Delta_i = 1 - p_i$$

Implications of 1-p

- If Δ_i assumes a value close to zero, it implies two things:
    - First, the student's trait ability for the dependee is high enough not to bother testing it,
    - And second, raising it to 1 would likely have little effect on the probability estimate of the parent.

- Thus the highest impact item is determined not only by the coefficient, but also by the gain in probability for the dependee.

The Odds Ratio

Supposing that the odds estimate for the parent j were given by

$$logit_j = e^{b_1 p_1 + b_2 p_2 + ... b_i p_i ... b_n p_n}$$

Then the odds estimate for the parent j if the probability were to rise to 1 would be

$$logit_j = e^{b_1 p_1 + b_2 p_2 + ... b_i (1) ... b_n p_n}$$

The Odds Ratio

Then the odds ratio would be equal to

$$\frac{e^{b_1 p_1 + b_2 p_2 + \ldots + b_i(1) + \ldots b_n p_n}}{e^{b_1 p_1 + b_2 p_2 + \ldots + b_i p_i + \ldots b_n p_n}}$$

which when reduced is simply

$$e^{b_i \Delta_i}$$

The Highest Increase in Odds Ratio

Thus that dependency which has the highest increase in the odds ratio is the item i such that

$$\operatorname*{argmax}_{i}\left[\left(e^{b_i}\right)^{\Delta_i}\right]$$

If that dependency has a probability within the window $[.5, .5+\delta]$, it is asked; otherwise if it has probability $[0, .5)$, the algorithm is recursively applied to the dependency.

Bibliography

📄 John R Anderson and C Schunn.
Implications of the act-r learning theory: No magic bullets.
Advances in instructional psychology, Educational design and cognitive science, pages 1–33, 2000.

📄 Dennis Castleberry and Steven R Brandt.
The effect of question ordering using bloom's taxonomy in an e-learning environment.
In *International Conference on Computer Science Education Innovation & Technology (CSEIT). Proceedings*, page 22.
Global Science and Technology Forum, 2016.

📄 Hermann Ebbinghaus.
Über das gedächtiß.
Passavia Universitätsverlag, 1880.