Modeling Student Retention in an Environment with Delayed Testing

by

Shoujing Li

A Thesis

Submitted by Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

April 2013

APPROVED:

Professor Joseph E. Beck, Thesis Advisor

Professor Kathi Fisler, Thesis Reader

Professor Craig Wills, Department Head

Abstract

Over the last two decades, the field of educational data mining (EDM) has been focusing on predicting the correctness of the next student response to the question (e.g., [2, 6] and the 2010 KDD Cup), in other words, predicting student short-term performance. Student modeling has been widely used for making such inferences. Although performing well on the immediate next problem is an indicator of mastery, it is by far not the only criteria. For example, the Pittsburgh Science of Learning Center's theoretic framework focuses on robust learning (e.g., [7, 10]), which includes the ability to transfer knowledge to new contexts, preparation for future learning of related skills, and retention - the ability of students to remember the knowledge they learned over a long time period. Especially for a cumulative subject such as mathematics, robust learning, particularly retention, is more important than short-term indicators of mastery.

The Automatic Reassessment and Relearning System (ARRS) is a platform we developed and deployed on September 1st, 2012, which is mainly used by middle-school math teachers and their students. This system can help students better retain knowledge through automatically assigning tests to students, giving students opportunity to relearn the skill when necessary and generating reports to teachers.

After we deployed and tested the system for about seven months, we have collected 287,424 data points from 6,292 students. We have created several models that predict students' retention performance using a variety of features, and discovered which were important for predicting correctness on a delayed test. We found that the strongest predictor of retention was a student's initial speed of mastering the content. The most striking finding was that students who struggled to master the content (took over 8 practice attempts) showed very poor retention, only 55%

correct, after just one week. Our results will help us advance our understanding of learning and potentially improve ITS.

Acknowledgements

We want to acknowledge the funding on NSF grant DRL-1109483 as well as funding of ASSISTments. See here (http://www.webcitation.org/67MTL3EIs) for the funding sources for ASSISTments.

Most of all, I would like to thank my research advisor, Professor Joseph E. Beck, who introduced me to the field of Educational Data Mining, his advice and guidance on research. It was a great experience to work with him on research.

Also, I would like to thank my thesis reader, Professor Kathi Fisler, for her time and valuable suggestions. My thanks are also due to Professor Neil T. Heffernan, who introduced me to ASSISTments project and help me a lot on my research.

I have learned a great deal from Xiaolu Xiong, a PhD student at WPI, who gave me a lot of advices and support, and also worked together with me on developing and maintaining the ARRS system. Also thanks to all my co-workers on ASSISTments project.

Contents

List of Figures

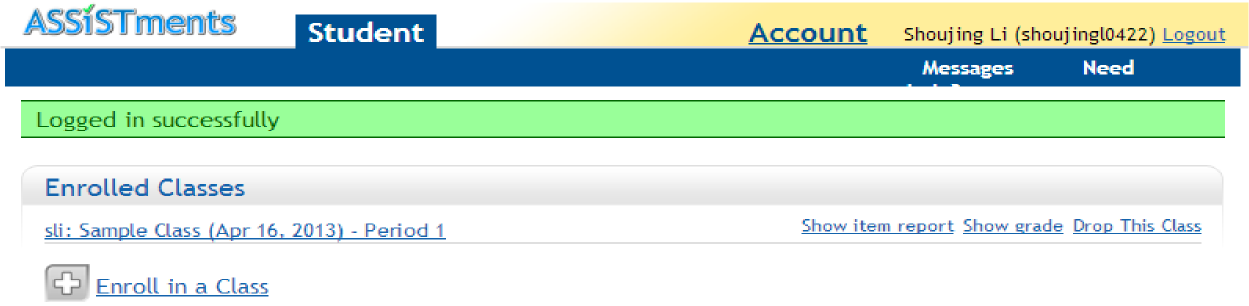
List of Tables

Chapter 1

Background

1.1 Automatic Reassessment and Relearning System (ARRS)

This thesis focuses on the design, development, and analysis of the ARRS system, which is an extension of the ASSISTments system [17]. The ASSISTments system is a well-known web-based tutoring system for 4th through 10th grade mathematics [9]. Students working with ASSISTments typically use the tutor one or more times per week for the entire school year. The word "Assistment" blends tutoring "*assist*ance" with "assess*ments*" given to teachers. Students who use ASSISTments can enroll into the online classes created by teachers (Figure 1.1), and finish the assignments that have been assigned by their teachers. When students are doing their assignments using ASSISTments, they will receive tutoring and immediate feedback as shown in Figure 1.2. In this problem, the student has made a mistake the ASSISTments is offering the student a webpage to help him learn the material.

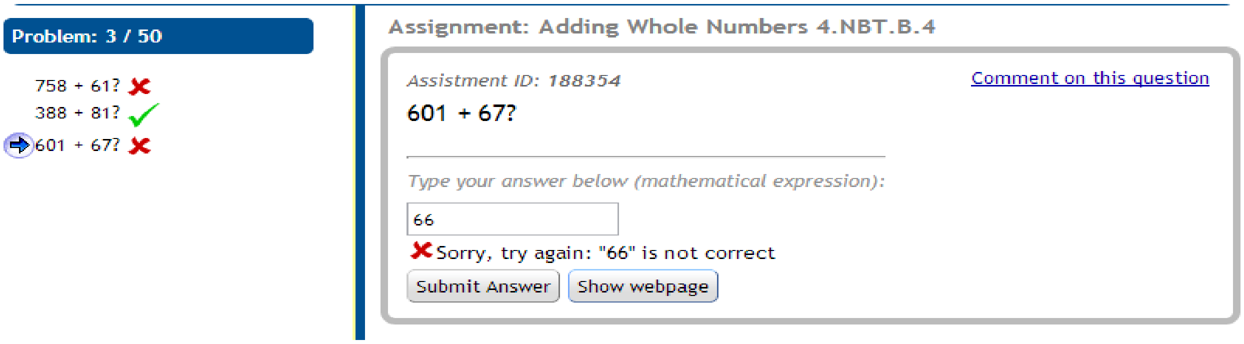


Logged in successfully

Enrolled Classes

sli: Sample Class (Apr 16, 2013) - Period 1 | Show item report | Show grade | Drop This Class

Figure 1.1: Student's class page



Problem: 3 / 50

758 + 61? ✗

388 + 81? ✓

601 + 67? ✗

Assignment: Adding Whole Numbers 4.NBT.B.4

Assistment ID: 188354

601 + 67?

Type your answer below (mathematical expression):

66

✗ Sorry, try again: "66" is not correct

Figure 1.2: Assignment interface

In Skill Building Problem sets, students are given problems to solve until they demonstrate mastery by getting three problems correct in a row. Students will receive tutoring if they get the problem wrong or, if they respond correctly, move to the next problem. The goal of the ARRS system is to ensure that students remember the skill after a long time period. Spaced practice, a learning technique that incorporates intervals of time between subsequent reviews of previously learned material, is known by teachers and researchers to improve student retention [1]. But not all teachers use spaced practice because of the difficulty for them to give such practice over a few months' time. Further, what if some students show they have not remembered the skill? Teachers cannot customize their instruction to each student, and do not want to be "educational accountants" and track students over long periods of time on each skill. The ARRS system is intended to help teachers give better teaching and help students achieve better learning.

I am the co-creator (along with Xiaolu Xiong) of the second-generation ARRS system which was deployed on September 1st, 2012 (a prototype version of ARRS was Sam Song's MQP project in 2010 [11]). The ARRS system can automatically give students spaced practice without teachers becoming educational accountants. Teachers are able, if they wish, to customize how the ARRS system will behave and not have to worry about the day to day details of how it behaves. Part of the ARRS system we developed provides reports for teachers so they can examine student progress. The system will test students' retention on different skills after a certain time period (e.g., [4, 8]), which defaults to one week, two weeks after that, one month after that, and finally to two months later. If a student fails the test, ARRS will provide him with the opportunity to relearn the skill. ARRS also provides summaries of student progress to teachers to help them understand common weaknesses in the class.

1.2 ARRS System Usage Summary

As described in the previous section, the motivating problem of ARRS is that if a student masters a problem set, such mastery is not necessarily an indication of long-term retention. Therefore, ARRS will present the student with a reassessment test on the same skill at expanding intervals. Figure 1.3 shows how ARRS system works to help students retain knowledge over a long time period.

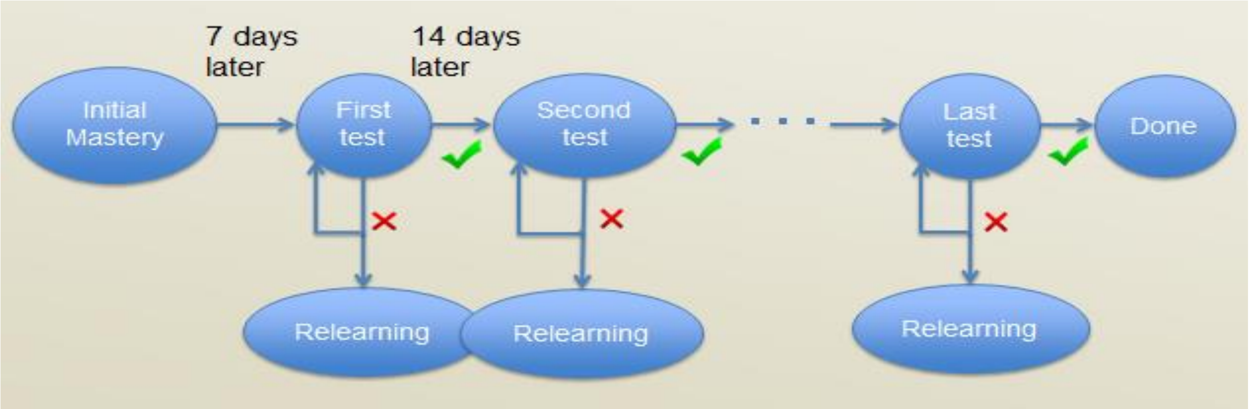
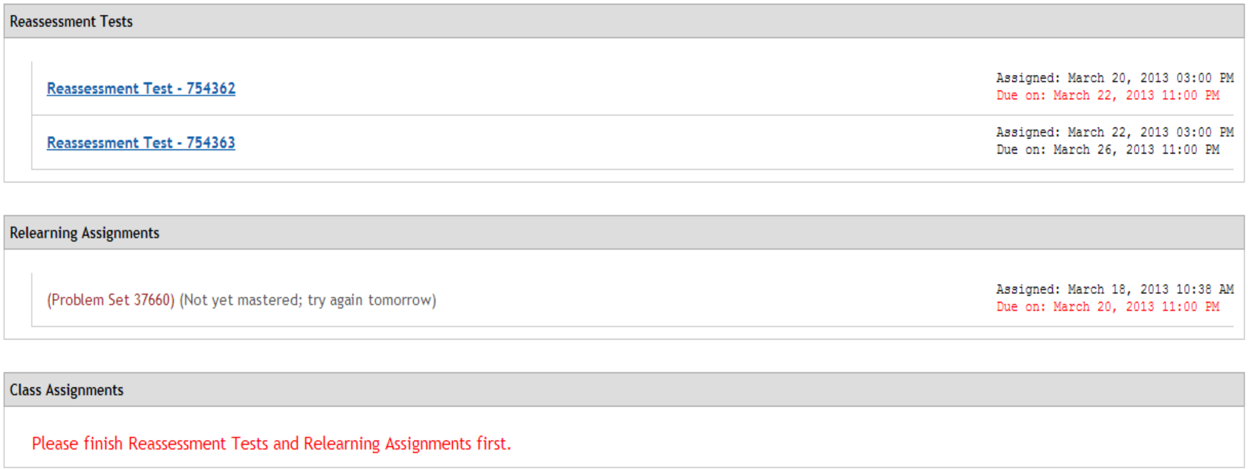


Figure 1.3: ARRS flow diagram

The most common setting for the ARRS system among our users is 4 reassessment tests in total, with the time interval between each test to be: first 7 days after the initial mastery, then 14 days after the prior test, than 28 days after the second test, and finally 56 days after the third test. Thus, the retention tests are spread over an interval of at least 105 (7+14+28+56) days.

Previously, the first reassessment test was assigned 7 days after the due date of the initial mastery. However, teachers expressed concern about students who had not mastered the skill being given retention tests on it. Therefore, we changed it to be 7 days after students demonstrated mastery on their initial mastery assignments. When students are doing retention tests, if they answer reassessment item correctly, then they will move to the next level retention test. If they give an incorrect response, the system will automatically assign a relearning assignment to the students and they will need to redo the test. Figure 1.4 shows the ARRS interface for students.



Reassessment Tests

Reassessment Test - 754362
Assigned: March 20, 2013 03:00 PM
Due on: March 22, 2013 11:00 PM

Reassessment Test - 754363
Assigned: March 22, 2013 03:00 PM
Due on: March 26, 2013 11:00 PM

Relearning Assignments

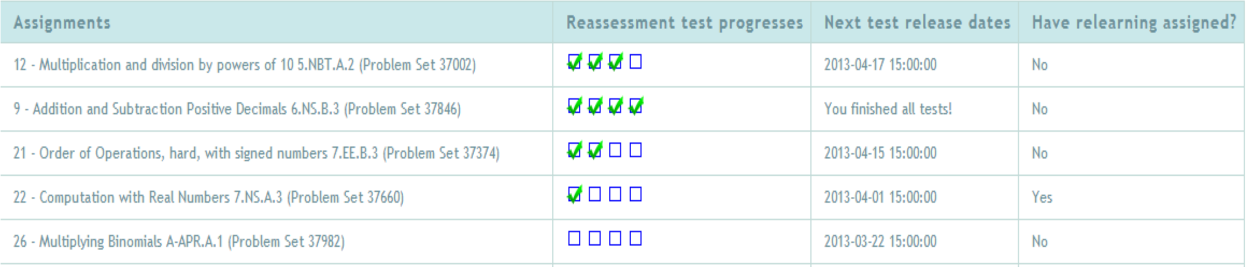
(Problem Set 37660) (Not yet mastered; try again tomorrow)
Assigned: March 18, 2013 10:38 AM
Due on: March 20, 2013 11:00 PM

Class Assignments

Please finish Reassessment Tests and Relearning Assignments first.

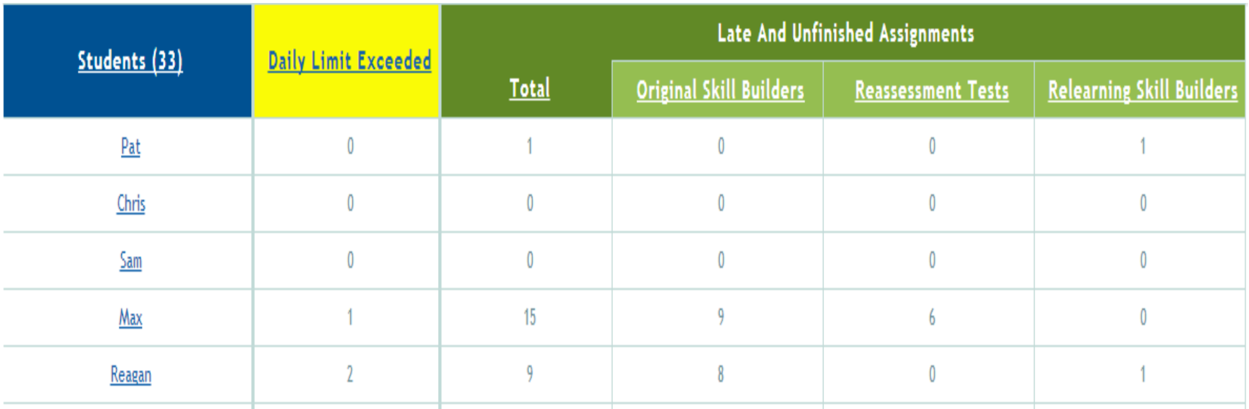
Figure 1.4: Student's interface

As we can see, this student has two reassessment tests on two different skills, one relearning assignment which indicates that he or she has done one previous reassessment test incorrectly, so our system gives the student an opportunity to relearn the skill. Note that the third section, "Class Assignments", in Figure 1.4, has nothing but a message in it. This is because the student is in *strict mode*, which means students in that class are forced to complete reassessment tests and relearning assignments prior to accessing other regular class assignments. Part of the ARRS system we developed provides reports for both students and teachers. Students can use reports to see their status in the ARRS circle (shown in Figure 1.3), and teachers can use reports to examine students' progress (shown in Figure 1.5 and 1.6).



Assignments	Reassessment test progresses	Next test release dates	Have relearning assigned?
12 - Multiplication and division by powers of 10 5.NBT.A.2 (Problem Set 37002)	☑ ☑ ☑ □	2013-04-17 15:00:00	No
9 - Addition and Subtraction Positive Decimals 6.NS.B.3 (Problem Set 37846)	☑ ☑ ☑ ☑	You finished all tests!	No
21 - Order of Operations, hard, with signed numbers 7.EE.B.3 (Problem Set 37374)	☑ ☑ □ □	2013-04-15 15:00:00	No
22 - Computation with Real Numbers 7.NS.A.3 (Problem Set 37660)	☑ □ □ □	2013-04-01 15:00:00	Yes
26 - Multiplying Binomials A-APR.A.1 (Problem Set 37982)	□ □ □ □	2013-03-22 15:00:00	No

Figure 1.5: Student's ARRS report



Students (33)	Daily Limit Exceeded	Late And Unfinished Assignments			
		Total	Original Skill Builders	Reassessment Tests	Relearning Skill Builders
Pat	0	1	0	0	1
Chris	0	0	0	0	0
Sam	0	0	0	0	0
Max	1	15	9	6	0
Reagan	2	9	8	0	1

Figure 1.6: Teacher's ARRS report

After deployed the system on September 1st, 2012, 219 classes from 63 schools are using this system. As a result, we have 6,292 students who finished 287,424 reassessment test problems and 311,146 relearning questions tagged with 111 different skills.

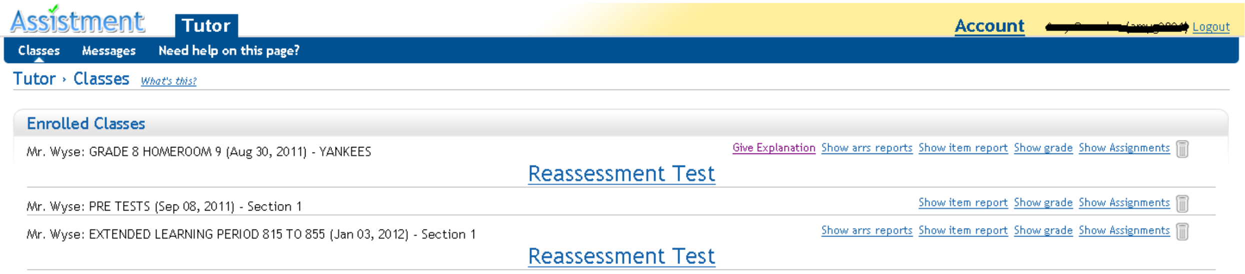
1.3 Comparison with Prototype Version of ARRS

The first-generation ARRS system was Sam Song's MQP project [11] implemented inside ASSISTments. It implemented the basic functionality to give students spaced practice and automatically generated reports for teachers. But after a while, we found some limitations and bugs within it, and also received some complaints and feature requests, which were the main reasons we decided to implement a new generation of the ARRS system. The following are some detailed comparison and explanation about why we moved to the new ARRS:

1. Student interface, reassessment tests and relearning assignments

In the prototype version of ARRS, reassessment items were combined into one big test. Figure 1.7 shows a student would have one link for all of the reassessment items for a particular class he is enrolled in. This interface required students to finish all the test items in order for the system to record the results. If students just did part of the test, the system will not record the students' answers. In our new system, questions are separated into small tests based on the day the original assignment was due.

The other issue is about the relearning assignments. As we can see from Figure 1.8, the assignment, for example "9-(9) Area Circle – THE SKILL BUILDING SET (Problem Set 10762)", represents both the original assignment and the relearning assignment, which makes students confused and when students have 20 or 30 or even more original assignments, they need to scroll down the page to find out which assignments they need to relearn. Both teachers and students complained about this. So in our new ARRS system, we have three separate sections for reassessment tests, relearning assignments, and regular class assignments (Figure 1.4), and only list skills which require the students to answer questions. This approach makes it easier for students to get an idea about what they need to do.



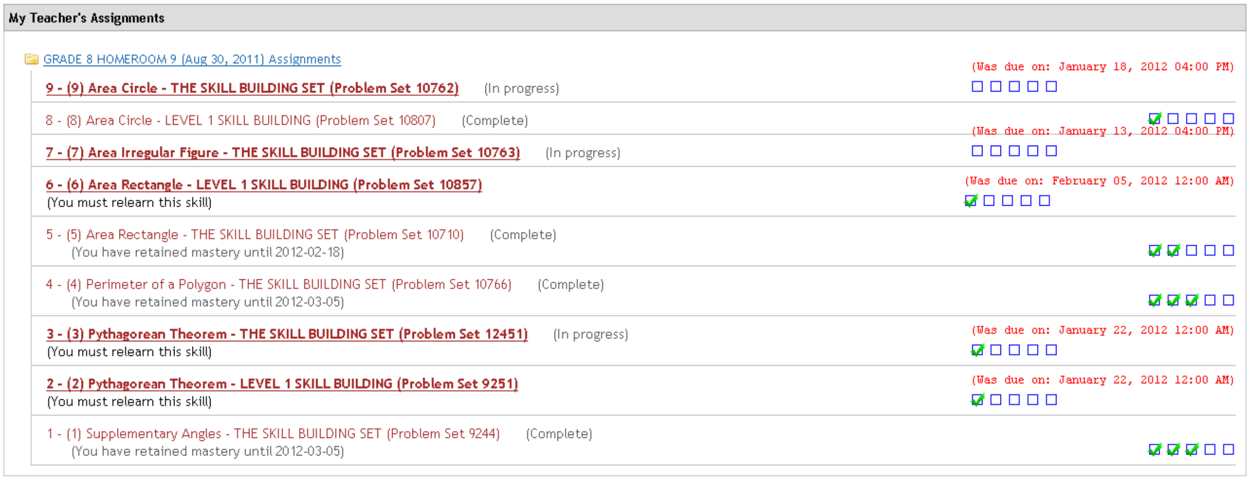
Enrolled Classes

Mr. Wyse: GRADE 8 HOMEROOM 9 (Aug 30, 2011) - YANKEES — Give Explanation · Show arrs reports · Show item report · Show grade · Show Assignments
Reassessment Test

Mr. Wyse: PRE TESTS (Sep 08, 2011) - Section 1 — Show item report · Show grade · Show Assignments

Mr. Wyse: EXTENDED LEARNING PERIOD 815 TO 855 (Jan 03, 2012) - Section 1 — Show arrs reports · Show item report · Show grade · Show Assignments
Reassessment Test

Figure 1.7: Student's interface for reassessment test



My Teacher's Assignments

GRADE 8 HOMEROOM 9 (Aug 30, 2011) Assignments

9 - (9) Area Circle - THE SKILL BUILDING SET (Problem Set 10762) (In progress) (Was due on: January 18, 2012 04:00 PM) □ □ □ □ □

8 - (8) Area Circle - LEVEL 1 SKILL BUILDING (Problem Set 10807) (Complete) ☑ □ □ □ □

7 - (7) Area Irregular Figure - THE SKILL BUILDING SET (Problem Set 10763) (In progress) (Was due on: January 13, 2012 04:00 PM) □ □ □ □ □

6 - (6) Area Rectangle - LEVEL 1 SKILL BUILDING (Problem Set 10857) (Was due on: February 05, 2012 12:00 AM)
(You must relearn this skill) ☑ □ □ □ □

5 - (5) Area Rectangle - THE SKILL BUILDING SET (Problem Set 10710) (Complete)
(You have retained mastery until 2012-02-18) ☑ ☑ □ □ □

4 - (4) Perimeter of a Polygon - THE SKILL BUILDING SET (Problem Set 10766) (Complete)
(You have retained mastery until 2012-03-05) ☑ ☑ ☑ □ □

3 - (3) Pythagorean Theorem - THE SKILL BUILDING SET (Problem Set 12451) (In progress) (Was due on: January 22, 2012 12:00 AM)
(You must relearn this skill) ☑ □ □ □ □

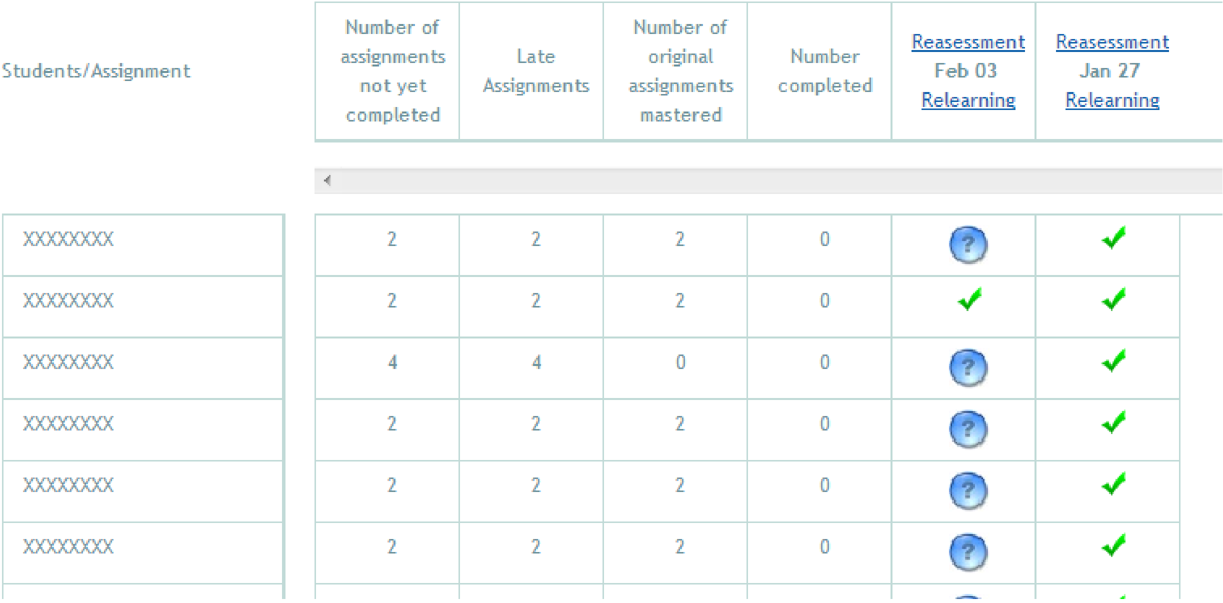
2 - (2) Pythagorean Theorem - LEVEL 1 SKILL BUILDING (Problem Set 9251) (Was due on: January 22, 2012 12:00 AM)
(You must relearn this skill) ☑ □ □ □ □

1 - (1) Supplementary Angles - THE SKILL BUILDING SET (Problem Set 9244) (Complete)
(You have retained mastery until 2012-03-05) ☑ ☑ ☑ □ □

Figure 1.8: Student's interface for relearning assignments

2. Teacher reports

The other important reason we redesigned the ARRS system is that the previous version of teacher reports didn't give enough information about students' progress and it was hard to interpret (Figure 1.9). Teacher cannot easily tell which students are far behind the other students and which skill most students are having trouble with. We designed our new reports (Figure 1.6 in the previous section) on the basis of teacher requests for how the interface should work. We provide teachers information about the number of each student's late assignments on original skill builders, reassessment tests, and relearning assignments. Teacher can look into the detail or sort by the total number of late assignments to see which students are far behind. Also our new reports tell the teacher on how many assignments each student has exceeded ASSISTments' *daily limit.* The daily limit is set to 10 problems, before which the student must achieve three correct responses in a row. This statistic is important as it means the student is really stuck on that skill and the teacher should probably talk about the skill during class if many students exceeded daily limit.



Students/Assignment	Number of assignments not yet completed	Late Assignments	Number of original assignments mastered	Number completed	Reassessment Feb 03 Relearning	Reassessment Jan 27 Relearning
XXXXXXXX	2	2	2	0	?	✓
XXXXXXXX	2	2	2	0	✓	✓
XXXXXXXX	4	4	0	0	?	✓
XXXXXXXX	2	2	2	0	?	✓
XXXXXXXX	2	2	2	0	?	✓
XXXXXXXX	2	2	2	0	?	✓

Figure 1.9: Old version of teacher's report

3. Functionality

In our second-generation ARRS system, we provide more functionality to make the system work more intelligently. For example, we combine tests on the same skill into one test, so students won't be tested on the same skill several times on the same day. We also provide more settings for teachers to make ARRS system behave as they want. For example teachers can set the total number of reassessment tests, the interval between each test, the release time of the tests and the due date. One other new feature we implemented is called "strict mode", which forces students to finish reassessment tests and relearning assignments before they can access their regular class assignments.

Chapter 2

Introduction

2.1 Motivation

Researchers of intelligent tutoring systems (ITS) and educational data mining (EDM) have been giving increasing attention on predicting student long-term retention performance and trying to find good methods to help improve student knowledge retention [8]. Wang and Beck proposed a system (Figure 2.1) which allows ITS to aim for student long-term mastery learning [4]. Inspired by the design of the enhanced ITS mastery cycle, we developed and deployed the ARRS system which makes decisions about when to review each skill the student has learned.

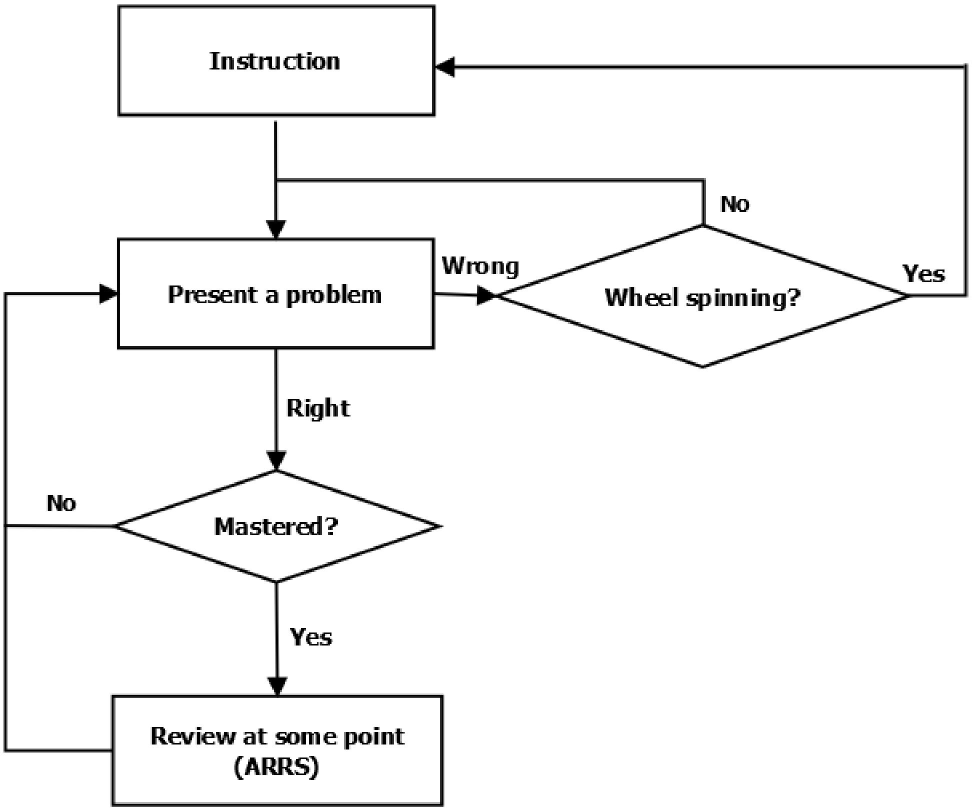


Figure 2.1: Enhanced ITS

Previously, researchers in the field of educational data mining (EDM) focused mostly on predicting the correctness of the next student response, in other words, predicting student short-term performance [13]. But for a cumulative subject such as mathematics, we are more concerned long-term retention than with short-term indicators of mastery. The ARRS system enables us to get data from a system that is intended to measure students' retention performance. Using student modeling methods and data mining techniques, we can get a better understanding of a student's learning process to better improve that student's retention. Although reviews and tests can be used by teachers to help prevent forgetting (e.g., [18, 20], no one wants to be an "educational accountant," and it will be a lot of work for teachers to track individual student's performance. So a system that can do all those work automatically and a reliable model in predicting student retention performance may reduce teachers' work. Furthermore, it would be important to better helping and predicting student's retention performance from an educational data mining point of view.

2.2 Problem Definition

As we discussed in the previous chapter, we've developed and deployed the ARRS system inspired by the enhanced ITS mastery circle. In order to make the system better at helping students retain their knowledge, we proposed the following research questions:

***Research Question 1*:** How long should we wait to give students a retention test?

1a: How can we apply data mining techniques to the student data to determine the best intervals?

1b: Does the best interval vary by student?

1c: Does the best time interval vary by skill?

***Research Question 2*:** Can we automatically detect interference[1] between different skills?

2.3 Related Work

Wang and Beck have created several models to predict students' retention performance and discovered several helpful features. We have done some studies that extend Wang and Beck's models and get some interesting results.

Wang and Beck [4] investigate predicting student performance after a delay of 5 to 10 days, to determine whether, and when, the student will retain the material seen. Two aspects that

[1] Interference occurs in learning when there is an interaction between the new material and transfer effects of past learned behavior, memories or thoughts that have a negative influence in comprehending the new material.

make their work interesting are: first, the factors influencing retention are different than those influencing short-term performance. Specifically, they found that the number of student correct and incorrect responses were not reliable predictors of long-term performance. Their result is in contrast to most student-modeling efforts on predicting performance on the next response. Second, they argue that answering the question of whether a student will retain a skill is more useful for guiding decision making of intelligent tutoring systems (ITS) than predicting correctness of next response. They introduce an architecture that identifies two research topics (which are "is student retention predictable?" and "does forgetting vary by student?") that are meaningful for ITS decision making. Their experiments found one feature in particular that was relevant for student retention: the number of distinct days in which a student practiced a skill. This result provides additional evidence for the spaced practice effect, and suggests our models need to be aware of features known to impact retention.

2.4 Goals Achieved

Our goal for this research work is to help students get a better retention on skills they have learned and to better predict students' retention performance. To summarize:

- We've developed and deployed a second-generation ARRS system, which decides when to review a skill that a student has learned and automatically assign reassessment tests, relearning assignments, and generate reports about students' progress to teachers.
- We showed that class level features (Chapter 3) are worth modeling when predicting student's retention performance. Most of the previous work focused on student's individual information instead of this kind of high level information. In this work, we

also found that mastery speed is an important feature that affects students' retention performance.

- We also presented several models on predicting students' retention performance using different feature sets and found that mastery speed is the most useful feature (Chapter 4). We also showed that mastery speed not only affects students' first retention tests, but also has a long term effect on students' retention performance even after 105 days. In our further studies (Chapter 5), we also presented that mastery speed is affected by both the students themselves and the skills.

2.5 Chapter Overview

The document is organized as follows:

- In Chapter 1 we introduced the Automatic Reassessment and Relearning System (ARRS), and showed the current usage summary after deployed the system on September 1st, 2012. We also compared the new ARRS with the prototype version of ARRS.
- In Chapter 2 we proposed the importance of predicting student's retention performance and providing a tool to release retention tests, also defined the tasks of this research work and described the goals we have achieved.
- Chapter 3 presented our analysis on using class level features to predict student's retention performance. We found that some class features are worth modeling since we get better prediction when using these features. This work was published in part of the following two papers:

  - Xiong, X., Li, S., Beck, J. (Accepted) Will you get it right next week: Predict delayed performance in enhanced ITS mastery cycle. The 26th International FLAIRS Conference
  - Xiong, X., Beck, J., Li, S. (Accepted as a short paper) Class distinctions: Leveraging class-level features to predict student retention performance. 16th International Conference on Artificial Intelligence in Education

- Chapter 4 proposed several models that predicted student's retention performance using different feature sets and tested the real effect of each feature in several separate models. This work is submitted to the 6th International Conference on Educational Data Mining, 2013.
  - Li, S., Xiong, X., Beck, J. (Accepted) Modeling Student Retention in an Environment with Delayed Testing. 6th International Conference on Educational Data Mining, 2013

- Chapter 5 summarized our research results and presented some further analysis to answer our research questions.

- Chapter 6 put all the proposed methods into context for conclusion. We also gave an outlook in this area and suggested some work for the future.

Chapter 3

Leveraging Class-level Features to Predict Student Retention Performance

As we discussed in the previous chapter, most educational data mining researchers have investigated the prediction of student behavior on the immediate next question, in other words, student short-term performance. But we are more concerned about robust learning, particularly retention, than with short-term indicators of mastery. More specifically, this part of our studies focuses on the problem of estimating student retention, and discovering factors likely to influence it, especially class level features.

3.1 Intuition and Approach

In general, student modeling uses data about a student's performance in order to assess his degree of knowledge. However, consider a situation where all of a student's classmates respond incorrectly to a particular item. When this student encounters the item, we would not expect him to respond correctly based on his peers' performance. Strangely, most student modeling approaches would not take advantage of this information, even though it is presumably relevant to understanding this student's knowledge. Rather, student modeling typically uses data from this student, except for estimating model parameters. So in our studies, we will use class level features along with other potential helpful features to predict students' retention performance.

Our data about students' retention performance came from ARRS system which was first deployed on September 1st 2012. At the time we run this study, 182 classes from 50 schools are using this system. As a result, we have 3422 students who finished 83,159 reassessment test problems as well relearning problems when they respond incorrectly. One of the important characteristics of these data is that they represent the students' long-term performance on different delayed time periods. By employing this dataset, we believe that we can build models to predict if students will remember a skill after a certain period of time and help to solve ITS decision-making process. For instance, if our models can tell that a student appears likely to retain a skill, it is probably not necessary to keep presenting that item of that skill. Conversely, if a student is predicted to be unlikely to retain the skill, the tutor can give additional items, or ask the student to do review problems perhaps as soon as a few minutes later. However, it is unclear how to apply a short-term model such as KT or PFA for such a decision-making task, as they simply predict the probability of getting the next item correct.

3.2 Modeling Retention

The first step toward developing a model to predict student retention performance is to create an operational definition of performance that can be used as a training label. As we employed data from the ARRS system, we operationalized retention performance as the reassessment test performance one week after the due date of the initial mastery assignment (i.e., the first reassessment test). We selected these data as they are the most numerous, and investigating later reassessment tests requires finding a way to represent the student performance on prior reassessment tests and relearning, if any. Thus, for simplicity, we start by modeling performance on the first test.

Given this definition of retention, we built such models by a data driven approach: rather than attempting to predict every next student performance, instead we focus on student performances that occur on his first attempt at the first retention test for each skill. In this way, even though we are not explicitly modeling the forgetting and retention, our straightforward modeling approach captures aspects of performance that relate to student long-term retention of the skill. Wang and Beck [4] already suggested that features like number of student correct and incorrect responses were not reliable predictors of long-term performance, which is in contrast to most student-modeling efforts on predicting performance on the next response. So naturally, we interest in what factors and features influence the prediction of long-term performance in our ARRS data.

In the study of ARRS data, we first noticed that the number of attempted problems before students achieve mastery has great influence on the one-week delayed performance [19]. At a minimum, students require 3 attempts to master a skill since ASSISTments uses 3 correct in a row as its mastery criterion. If a student gets the first item wrong, he could master the skill in 4 attempts. We refer to this number of problems required as the mastery speed. Mastery speed represents a combination of how well the student knew this skill initially, and how quickly he can learn the skill. The question is whether mastery speed is a useful measure of students' retention performance?

Table 3.1 shows the relation between student mastery speed and performance on delayed retention tests. We binned the mastery speed into four categories which were "3-4 attempts", "5-8 attempts", ">8 attempts" and "failed at mastering." We observed that, in general, the slower the mastery speed, the lower the probability that the student can answer the retention item correctly. Students who mastered a skill in 3 or 4 problems had an 82% chance of responding correctly on the first retention test, while students who took over 8 attempts to master only had a 59% chance of responding correctly on the first retention test. Why this result is of interest is that most ITS simply label a student as having mastered the skill or not. This result shows that there are wide variations in performance based on *how* a student mastered a skill. Thus, not all mastery is alike, with an approximate 10% decrease in percentage correctness between each level of mastery speed. It is surprising that such stark differences should appear given our method of disaggregating mastery was straightforward. Finally, there is a group of students who tried but failed to master the skill and, predictably, did the worst.

Table 3.1: Relation between speed of mastery and one-week retention

mastery speed (# of problems needed to master the skill)	% correctness on one week delayed retention tests
3-4 attempts	82%
5-8 attempts	70%
> 8 attempts	59%
failed at mastering	44%

3.3 Modeling Class-level Effects

We employed the mastery speed, as well as three other basic features to establish a baseline for our modeling work. These features focused on item and skill information, including:

- on_grade: whether this skill is typically taught in the same grade-level of the student. Perhaps skills from prior grades are better understood and thus better-retained; for example, it is unlikely 12-year old students have forgotten how to add whole numbers. Alternately, students might not remember skills from prior grade levels if it is an infrequently used skill, such as finding the volume of a solid.
- grade_diff: the binned value of grade difference. We compute the grade difference by student current grade - skill grade. This feature is potentially helpful since students might have different performance on skills that are far below the students' grade, skills that are on the same grade level of the students, and the skills which might be above the students'

grade level. For the grade level of the student, we used the value supplied by the teacher. For the grade level of the skill, we used a mapping created by Cristina Heffernan, a subject-matter expert, who works on the ASSISTments project.

- item_easiness: this number is represented by using the percentage of correctness for this problem across all answers and all students and across whole ASSISTments system.

We fit this base model using multinomial logistic regression; we got an R^2 of 0.183. The p-values of the model show that master speed, grade_diff, and item_easiness are all reliable predictors of students' retention performance. Since this model fit is not that strong, and is on our training data, we continued to work to improve the model.

When building this model, we assumed that student performance is relevant to features of the student (the mastery speed, on_grade, and grade_diff information) and a global item easiness feature. It is worth to notice that these two aspects of individualization and population averaging features are also similar to PFA models [5], as such models track the number of correct and incorrect responses, but differ more from Bayesian Knowledge Tracing. We have the intuition that there might be some other factors between an individual student and the whole population can help improve our prediction. In both ASSISTments (and perhaps most ITS) and in traditional school teaching environments, students in the same class typically get the same assignments. Based on this observation, we considered the following couple scenarios: To predict the one-week delayed reassessment test performance for a certain student, if we knew that all of his classmates took the test for the same skill before him and they answered a particular item incorrectly, how would we predict the correctness of the student's answer? Similarly, how would knowing that all of his classmates answered the item correctly affect our prediction? Therefore, we made a hypothesis that the class performance and student individual performance are not independent and, to enhance our model, added features that represent the class' performance.

To test our hypothesis, the next step was to gather a set of class-level features and identify which features can be used as predictors. Towards this end, we selected the following three features to capture different class level information:

- class_id: classes were created by teachers who using the ASSISTments, and represent each distinct class a teacher has (so a teacher with multiple classes of math will have distinct class_id values). We used class_id as a fixed effect in the model. Although modeling class_id as a random effect may be theoretically warranted, our statistics software did not support this option. By modeling class_id as a factor, we are estimating an overall effect of the classroom, including data from this skill, from other skills, and even from the future. Thus, this factor is not realistic, but provides an estimate of how well we could hope to do with a covariate.
- class_prior_performance: measures the class' performance on prior reassessment tests on same skill. For each reassessment test, the performance is represented by using the percentage of correctness of tests that have been answered in the same class, on the same skill, and have been answered before the student attempt this retention item. Thus, this variable is realistic for use in an actual system.
- class_other_skill_performance: measures the class' performance on all reassessment tests on all other skills (i.e., ignores the skill this reassessment item is about). This feature is permitted to use data from the future, and is thus not realistic in an actual system, but provides an upper bound for how well such information could work.

3.4 Model Results

To train our model, we used 42,332 instances of a student using the ARRS system and attempting the first retention test for each skill one week after mastery. We separate these data into 33,866 instances for the training set and 8,466 for the testing set. The training set was selected by randomly selecting 20% of the dataset. Thus, this model does not test our ability to generalize to new students.

To investigate how our class-level features could impact our predictions on student retention test performance, we started from our base model, described in the first half of section 3.3, and added to it a representation of the class' performance. We experimented with using the class_id as a factor, prior performance on this skill's retention test, and all performance on all retention tests that did not involve this skill. Table 3.2 provides the results for each of these models. We provide both the classic R^2 metric, as well as the Nagelkerke (pseudo) R^2 for comparison purposes as other logistic regression results reported have used Nagelkerke [e.g., 12]. We can see that the additional features provided some improvement. However, on the test set, the class' performance on other skills was not predictive of retention test performance, with R^2 unchanged at 0.142 in both cases. However, student performance on the skill increased R^2 by 0.01 to 0.155, a modest increase for improving a student model's accuracy. class_id appears to generalize well, given performance on the test and training set, but this is deceptive as instances from the same class are in the test and the training set. Although this procedure overstates the results, creating a training set without instances of a particular class would leave the model unable to generalize at all. For those rare instances when a class only appeared in the training data, we imputed the value of the various class parameters by using the mean value observed in the training data.

Table 3.2: Class level model performance

Model	R^2 on training set	R^2 on test set
Base model + *class_id*	0.158 (Nagelkerke: 0.215)	0.159
Base model + *class_prior_performance*	0.155 (Nagelkerke 0.204)	0.153
Base model + *class_other_skill_performance*	0.145 (Nagelkerke 0.185)	0.142
Base model	0.143 (Nagelkerke 0.183)	0.142

From the above results, we can see that new model with class_id and class_prior_performance performed slightly better than the base model. The importance of class_id in the prediction may suggest that there seems to be an overall class effect that differs from average performance on other skills, which is modeled by class_other_skill_performance.

The results from using class_prior_performance clearly indicate that classmates' performance on similar content is helpful for improving predictions. These results confirmed our intuition about using class level performance as predictor of student retention test performance. One question is whether combining the two features would be fruitful in improving accuracy? Somewhat surprisingly, a model using both class_id and class_prior_performance achieved an R2 value of 0.165. Thus, whatever class_id represents, it is relatively distinct from class_prior_performance as the R^2 increases noticeably when both are modeled.

From an educational data mining point of view, class-level information can help better predicting students' retention performance and also help make ITS work more intelligently. If students in one class have some trouble on a certain skill, shortening the time interval between tests may be necessary and may also improve students' performance and knowledge retain. Also class-level poor performance on certain skills may indicate that teachers probably should speed more time on instructions on those skills during regular classroom time.

3.5 Contributions

This analysis makes three contributions. First, we identify speed of mastery as a useful construct. Bayesian Knowledge Tracing [6] and its extensions (e.g., [13, 15]) have what is known as the Markov assumption that past history is unimportant, as the required knowledge can be represented with the current state. For example, if a student has an estimated knowledge of 0.8, it does not matter whether he achieved that level with 3 problems or with 30. We have found that such longer-term trends are important, and thus researchers should either use a modeling approach that does not have the Markov assumption, or enrich their state representation to better account for longer-term performance. Speed of mastery is unlikely to be considered in the course of more normal uses of student modeling, as predicting student performance post-mastery is an atypical problem, as frequently systems aren't as concerned with topics the student has already mastered. Thus, this work has discovered a new feature relevant to robust learning.

Second, this work explored and identified class-level effects as being worth modeling. Most prior works have focused on features at individual student level and population averages, such as item difficulty in PFA [5], or prior knowledge as being associated with all students on a particular skill in Bayesian knowledge tracing [6]. We suspect this bias is from the KDD Cup 2010 on Educational Data Mining, as that dataset did not include indicators about which class the student was in. Our analysis adopted class level features in order to account for influences that will affect all members of the class, such as quality of classroom instruction, amount of time devoted to the topic, and time spent on prerequisites.

The third contribution of this work is by employing class id in our prediction; we adopted a generic approach for intuitively “clustering” students. There is prior work on clustering students into similar groups in order to improve predictive accuracy [14]. Our approach can be thought of as a straightforward approach to clustering students by their math class. Such an approach misses potential deeper connections; perhaps two math classes at completely different schools could be joined since they are both advanced calculus courses? However, our approach of clustering requires little additional information, no complex processing, and it is easy to understand our clusters and the semantics behind them.

Chapter 4

Modeling Student Retention Using Different Feature Sets

4.1 Data Set and Feature Selection

For this study, we used data from ARRS system, specifically students' 7-day test performance and other features about their previous knowledge on that particular skill. The features were collected consisted of the student's identity, the correctness on the retention item, mastery speed, and the skill needed to solve the problem. When we performed this analysis, we had 48,873 questions answered by 4054 students, from 91 different skills. Then we used data from January 2012 until the time point before students' first reassessment test (1,236,252 problem attempts) to calculate the following features which were used in our regression models:

- mastery_speed: the number of problems needed to master a certain skill. We binned this feature into 6 categories ('<3 attempts', '3-4 attempts', '5-8 attempts', '>8 attempts', 'not mastered', 'skipped initial mastery'). Students could master a skill in less than 3 attempts if their teachers overrode ASSISTments' mastery criterion.

- n_correct (n_incorrect): the number of students' prior correct (incorrect) responses on that skill before the retention test.

- n_day_seen: the number of distinct days that the students have practiced this skill.

- g_mean_performance: the exponential decayed mean of students' correctness on those problems finished before the reassessment test. By using exponential mean, the most recent performance will have the largest impact on the feature. We used same formula as in Wang and Beck's previous work [4]:

  *g_mean_performance(opp)=g_mean_performance(opp-1)*0.7+correctness(opp)*0.3*

  using opp to represent the opportunity count and a decay of 0.7.

- g_mean_time: the exponential mean of students' response time on that skill before the reassessment test. The formula is: *g_mean_time(opp)=g_mean_time(opp-1)*0.7+response_time(opp)*0.3*.

- problem_easiness: percentage correct for this problem across all students.

4.2 Models and Results

4.2.1 Overall Model Comparisons

For the first model, we applied multinomial logistic regression using SPSS with all features we created to the data set. We used user_id, skill_id and mastery_speed as factors in this model, and other features as covariates. This model has an R^2 of 0.382 which indicated a reasonable model fit and also showed that students' retention performance is predictable.

Table 4.1 shows the Beta-coefficient and p-values for each covariate; a positive B value means the larger the covariate the more likely the student will get the retention item correct. The mastery_speed feature we created, along with user_id and skill_id all had p-values below which SPSS could represent, so we report 0.

Table 4.1: Parameter table of covariates in Model 1

Covariate	**B**	**p-value**
n_correct	0.009	0.000
n_incorrect	-0.009	0.005
n_day_seen	0.020	0.116
g_mean_performance	0.710	0.000
g_mean_time	0.000	0.294
problem_easiness	6.011	0.000

From the table, we can see that problem_easiness and g_mean_performance are reliable predictors, which is the same conclusion from prior work on modeling retention [3]. Features such as n_day_seen and g_mean_time are not reliable predictors. Our result that n_correct and n_incorrect are reliable predictors contradicts prior work on retention [4]. However, a multivariate model like this one does not tell us which of those factors are related to retention, just how useful they are in the context of the other (unfortunately, correlated) “independents”. Our next step is to test each feature one at a time to see their real influence.

4.2.2 Separate Model with each Feature

We first tested a base model with just three features: user_id, skill_id, and problem_easiness, which showed as reliable predictors in our first model. This model provided an R^2 of 0.373. The next step we took was to test each feature one at a time added to the base model with just three features. Table 4.2 shows the Beta coefficient, p-values and R^2 gain for each regression model.

Each row in the table represents one regression model, with the feature listed and other three features in the basic model. The last column, R^2 gain, shows the increase in R^2 from adding that feature to the base model. Given even the modest (by EDM standards) data set we have for this study, circa 50,000 rows, even trivially small effects can show up as statistically “significant.”

Therefore, we compute how much improvement in accuracy the feature actually provides us. From the table, it's clear that mastery_speed is the most powerful predictor for students' retention performance. And also the students' previous performance on that skill (g_mean_performance) has a clear influence on prediction. The other variables have a trivial impact on performance. Note that even the best two features have a small impact on retention.

Compared with prior work [4], we found that n_day_seen did not replicate as being a useful feature. Strangely, a student's number of correct responses has little impact on retention. One possible explanation is that students who get a large number of items correct probably took more practice opportunities to master the skill, which indicates a larger number of *incorrect* responses as well. Thus, this feature does not contribute much to prediction.

Table 4.2: Parameters table for separate Models

Feature	**R^2**	**B**	**p-value**	**R^2 gain**
mastery_speed	0.379	---	0.000	0.006
n_correct	0.374	0.010	0.000	0.001
n_incorrect	0.373	-0.007	0.004	0.000
n_day_seen	0.373	0.026	0.002	0.000
g_mean_performance	0.378	1.130	0.000	0.005
g_mean_time	0.373	0.000	0.649	0.000

4.2.3 Impact of Mastery Speed

From the previous models we presented, we found that mastery speed has a clear influence on students' 7-day reassessment tests. However, what about the 14 day test, 28 day test, and even the 56 day tests? We collected all student performances on all four reassessment tests. As shown in Figure 4.1, we calculated the percentage of correctness on each retention test, disaggregated by initial mastery speed.

To interpret the graph, the first point to be aware of is that students must get the previous level's retention test correct in order to get to the next level. An incorrect response necessitates repeating the retention test with the same interval. Therefore, there is a selection bias since the weaker students are screened out as the retention test interval increases. This bias makes it problematic to compare students across different test intervals. Therefore, we focus our analysis within a single retention test. A second issue is that students may progress to the reassessment tests without even attempting their initial mastery exercises, so we got one mastery speed category which is "skipped initial mastery" (as noted previously, this behavior has since been removed from ARRS).

As Figure 4.1 shows, students' performance gets better as they move to the later retention tests. This outcome is expected since students must get the previous tests correct, so the ARRS system effectively screens out weaker students from progressing to the 56-day test. Weaker students are systematically oversampled on the left side of the graph. On the 7-day retention test, students who mastered a skill quickly with 3 or 4 attempts (blue continuous line) have a 24% higher chance of responding correctly than those students who required more than 8 attempts to master a skill (purple dash dot line). Such a difference is perhaps not surprising. More interesting is the persistence of this differential performance: the 56 day level tests, the group who mastered quickly are still performing about 15% better than the students who mastered slowly. This difference persists in spite of weaker students being screened out on earlier retention tests. This result tells us that the initial mastery speed does matter and affects students' retention performance even after about 105 days.

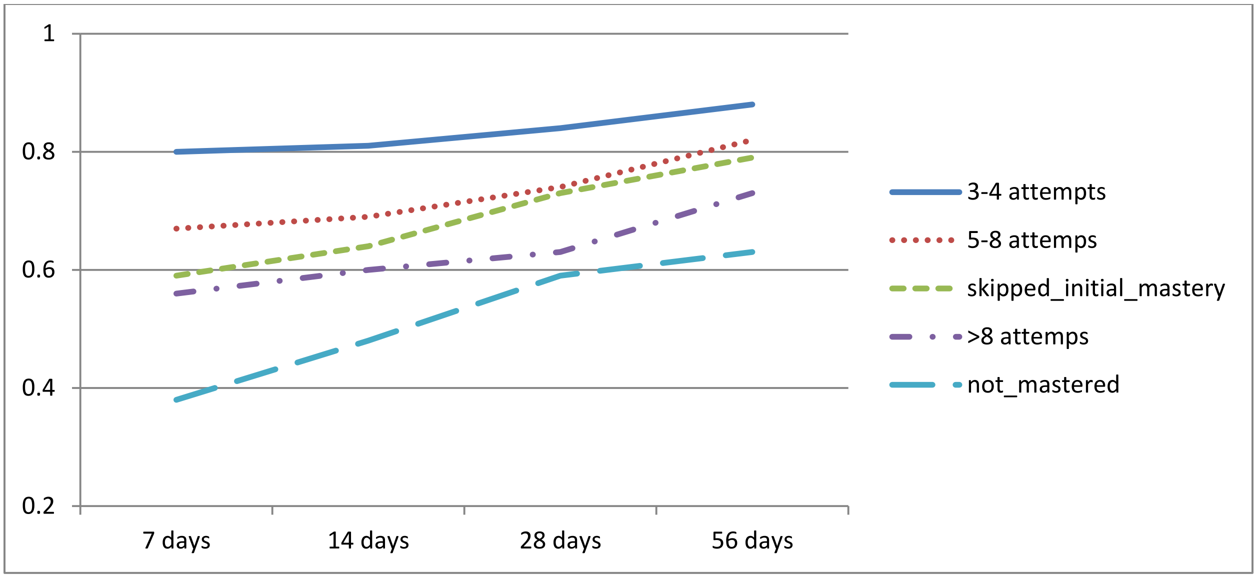


Figure 4.1: Impact of mastery speed on retention tests

4.3 Contributions

This analysis examined and compared several different models with different feature set using data from ARRS system which is intended to measure students' retention performance. So we are able to extend prior work on retention [4] and examined the implications of extending PFA [5, 12] to modeling retention.

For extending Wang and Beck's [4] study, we examined each individual feature to test their actual power of prediction. Now we are able to say, the students' previous performance before the test (g_mean_performance) and the prolem_easiness are useful features to know, while other features such as g_mean_time are relatively useless in predicting retention performance—at least in our models. The mastery speed feature we introduced turns out to be the most important feature influencing both short term and long term retention performance. And we also further confirmed their conclusion that retention performance is predictable since we also get a reasonable model fit with a R^2 of 0.382. Another improvement we did was that we used better dataset since our data came from ARRS system introduced at the beginning which was targeted as providing students retention tests and measure their performance.

The second contribution is extending the Performance Factors Analysis (PFA) proposed by Pavlik, et al. in 2009 [5]. We found that number of students' previous correct and incorrect responses, while being reliable predictors of retention, were not useful in improving model accuracy noticeably. However, we did find that initial mastery speed, which indicated how many problems students needed on initial mastery, was a useful predictor on the 7-day retention test, and remained a useful feature across a 3-month interval.

Chapter 5

Research Question Analysis

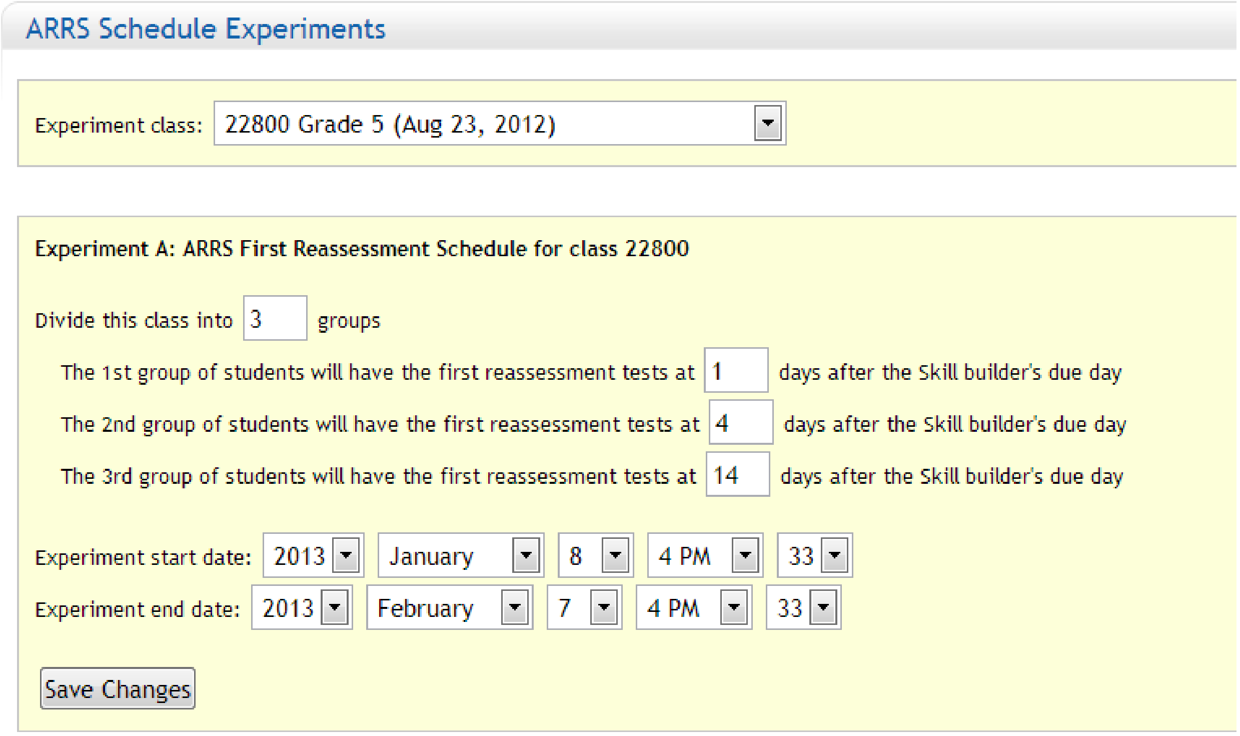
In this chapter, we summarize our results from the studies discussed in the previous chapters to answer each of our research questions, and also present some further analysis and infrastructure implementation related to some research questions.

5.1 RQ1: How long should we wait to give students a retention test?

The ARRS system uses 7 days, 14 days, 28 days, and finally 56 days as the default time intervals for spacing practice. The default setting is the most popular setting among our users (89% of classes used the default setting). Our goal for this question is to find the optimal interval setting for the ARRS system. By optimal interval setting, we mean a time interval which does not quiz students too frequently, but does not delay so long that the student forgets the skill. We suspect the speed of forgetting depends on several factors, which might include the difficulty of the skill and may vary among individual students.

5.1.1 RQ 1a: How can we apply data mining techniques to the student data to determine the best intervals?

After we deployed the ARRS system for about 4 months, we created the infrastructure in the ARRS system to run experiments using different intervals (Figure 5.1 shows the interface). The first question we want to answer is: what is the best delay for the first retention test. Currently, students get first retention test 7 days after initial mastery in the default setting, which is also the most popular setting. But we don’t have enough evidence to say that this interval is optimal, or even a good choice. We set up our very first experiment with two classes from one of our most familiar teachers. We ran this pilot experiment first to make sure our infrastructure works well, the data are recorded correctly, and teacher won’t have too much problems with variable intervals.



ARRS Schedule Experiments

Experiment class: 22800 Grade 5 (Aug 23, 2012)

Experiment A: ARRS First Reassessment Schedule for class 22800

Divide this class into 3 groups

The 1st group of students will have the first reassessment tests at 1 days after the Skill builder's due day

The 2nd group of students will have the first reassessment tests at 4 days after the Skill builder's due day

The 3rd group of students will have the first reassessment tests at 14 days after the Skill builder's due day

Experiment start date: 2013 January 8 4 PM 33

Experiment end date: 2013 February 7 4 PM 33

Figure 5.1: ARRS schedule experiments interface

Table 5.1 below shows the experiment settings for the two different classes.

Table 5.1: ARRS schedule experiment settings

Class	Setting
Class 1	1, 4, 14 days
Class 2	4, 10 days

As we can see in Table 5.2, Class 1 has a setting of “1, 4, 14 days”, which means one third of the students will get their first retention test 1 day after the initial mastery was due, one third of the students will be tested after 4 days, and the remaining one third of the students will receive first retention tests after 14 days. Table 5.3 shows Class 2, which has settings of “4, 10 days” which indicates that half of the students will get the first reassessment test after 4 days, and half will receive the test after 10 days. Since we already had considerable data on first retention test 7 days after initial mastery was due, we did not test this interval. As a result of these experiments, we will have data for first retention tests after 1, 4, 7, 10, 14 days. We expect the interval to fall within this range.

The pilot experiment started on Jan 8th, 2013 and ended on Feb 7th, 2013 for both the two classes. The results for the two classes are shown in Table 5.2 and 5.3. The results are not reliable due to the small sample size, but our real goal is to make sure that our infrastructure works well, the data are correctly recorded, and teachers don’t have too much trouble with the experiment.

Table 5.2: Results from Class 1

Class 1			
Interval	Number Correct	Number Incorrect	Percentage Correct
1 day	56	25	0.69
4 days	70	23	0.75
14 days	48	29	0.62

Table 5.3: Results from Class 2

Class 2			
Interval	Number Correct	Number Incorrect	Percentage Correct
4 days	295	78	0.79
10 days	227	62	0.79

Although this pilot study was not large, we are able to say that we have constructed a reliable infrastructure to schedule the ARRS experiments, and correctly record the results. This experimental mechanism is the first attempt at having an explicit infrastructure for running experiments within ASSISTments. The other reason why we cannot get sufficient data is that it’s hard to run longitudinal experiments since the complete ARRS cycle takes a long time compared with many other studies running in ASSISTments system, many of which only take one school day. Also we need certain usage patterns for this ARRS schedule experiment in order to get meaningful data. We noticed that some students use ARRS once a week, which means giving them the first retention test one or four days after the initial mastery makes no sense, because they won’t actually do it until one week later.

Currently we are running an experiment with 498 students in 8 classes. The experiment started on April 1st, 2013, and will end on August 1st, 2013. By the end of the experiment, we will have sufficient data to obtain reliable results.

5.1.2 RQ 1b: Does the best interval vary by student?

As described in Chapter 3, our analysis about using class level features (class_id, class_prior_performance, and class_other_skill_performance) to predict student retention performance shows that class_id and class_prior_performance are worth modeling to get better prediction about students’ retention performance. The new model with class_id and *class_prior_performance* performed slightly better than the base model. The model using both *class_id* and *class_prior_performance* achieved an R^2 value of 0.165. This result means that class level features, which affect all members of the class, such as quality of classroom instruction, amount of time devoted to the topic, and time spent on prerequisites, will result in different retention performance on the class level, which suggest different interval for different group of students. The semantics behind using class level features in prediction are clustering students by their math class, and we think that students in one class have something similar, for example, some of the classes maybe advanced calculus courses. Students in a class have certain similarities, and classes being different from each other, strongly suggests that students vary.

The study results presented in both Chapter 3 and Chapter 4 show that mastery_speed is an important feature in prediction. Chapter 4 even shows that mastery_speed was the best feature, outside of our base model, for predicting student's retention performance and has a long term effect on students' retention performance even after 105 days. According to these results, we can say that students with different mastery_speeds have different retention patterns. This result is clear evidence that customizing the retention interval can have a payoff. Students with fast mastery speed can have longer time interval between reassessment tests, compared with those students with slower mastery speeds. After this change, we may observe a smaller difference in performance on the first retention test for each group (Figure 4.1).

5.1.3 RQ 1c: Does the best time interval vary by skill?

Our intuition for this problem is that skills which are hard to learn could result in better retention performance if we shorten the time interval. Conversely, skills which are easy can be reassessed less frequently. We looked into our data, and found that the time students spent on different skills and the percentage of correctness for different skills differed a lot. Skills like "Substitution" and "Multiplication and Division of Integers" take around 20 seconds on average for students to answer and have an average percentage of correct around 65%. Skills like "Prime Factor" and "Translations" take around 80 seconds and only around 25% are answered correctly. Also we thought that skill information (since skills vary) and student identity (since students vary) may either be the key factor for the mastery speed feature which is the most important feature in our model. So we did the following further analysis:

We collected data from the ARRS system, specifically students' performance data on their 7-day retention tests. We have 101,541 data records answered by 4031 students on 111 different skills. Each data record contains features include student_id, skill_id, mastery_speed, and num_required_to_mastery (the actual attempts students made to reach mastery).

We first screened out the data records in which the students skipped initial mastery and moved forward to the reassessment tests directly, resulting in 87,294 data records remaining with num_required_to_master greater or equal to 3. We then calculated a target column simplified_mastery_speed with just two values (fast and slow), "fast" means the students mastered on the skill builder in the range of 3 to 10 attempts, while "slow" means the students mastered on the skill with greater or equal to 10 attempts.

We constructed three different multinomial logistic regression models for this analysis, all of which used simplified_mastery_speed as the dependent variable. The first model used student_id as the factor, the second one used skill_id, and the third one used both the student_id and skill_id as the factors to do prediction. Table 5.4 shows the regression results.

Table 5.4: Regression model results

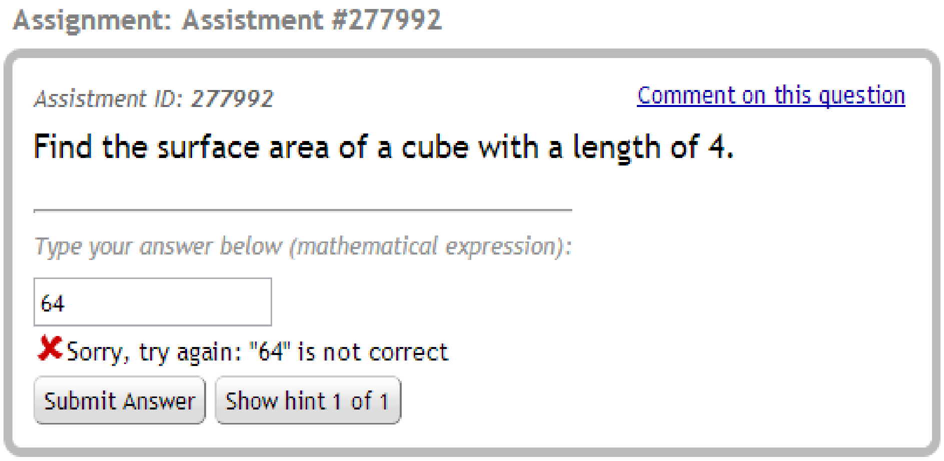
Model	**R^2**
student_id	0.283
skill_id	0.156
student_id and skill_id	0.428

We can see from the results that student_id has more effect on mastery speed compared with skill_id since the R^2 is higher. And for the last model, when we used student_id and skill_id together to do the prediction, we got an R^2 of 0.428. This value is almost the same as the sum of the R^2 from the two previous models combined (0.283 + 0.156 = 0.439). Since 0.428 is nearly the same of 0.439, this means that student_id and skill_id are largely independent predictors, and the mastery speed is a function of both the student_id and the skill_id. Therefore, our results indicate that the answer to RQ1b is that the retention interval should be a function of both the individual student and the content.

5.2 RQ2: Can we automatically detect interference between different skills?

Interference is defined as when one skill being learned impairs the recall of another. For example, learning a foreign language can be more difficult if a word is similar to an unrelated word (such as “ambulance” and “ambulatory”). Knowledge of one fact can impair the ability to retrieve other information. In the learning process, interference can impair students’ memory of prior material and the learning of new concepts. The intuition for this problem comes from one observation from an early pilot of ARRS that once students learned surface area, they performed very poorly on an ARRS retention test involving volume. Computer tutors would seem to be a strong research vehicle for better understanding such effects in an authentic learning context, and over longer time than typical psychology lab studies. If we could automatically detect such interference effects in student problem solving, that would advance our understanding of learning and potentially improve ITS.

We first asked teachers to give us some possible interfered skills according to their experience. The top three pairs of skills are: surface area and volume, mean and median, and area circle and circumference. Our first goal was to look into our data and see student’s performance on the skill pairs, especially the one that assigned later whose performance would be potentially influenced. If students tend to have bad performance on the later assigned skill, we could be able to automatically detect this interference. But to our surprise, we found that almost all of the teachers using our system assigned the skill pairs at the same time, which result in the lack of data and we cannot use student's performance to detect interference. Teachers may follow this procedure since learning both related skills at the same time helps students better understand the difference between the related skills. But at least we know that teachers have the same idea with us that interference between skills is worth to pay attention to. The next step we took was to look into the data and see if interference happened and how often it would show up. For the skill pair surface area and volume, we found that for surface area questions, volume was the most common wrong answer. For volume questions, students answered using surface area answer when the figure was not too complicated. Figure 5.2 shows the common cases for this pair of skills. I checked 50 problems for skill surface area, and find out that among all wrong answers, around 26% are correct if we change the question to ask about volume. The median and mean skill pair didn't have too much interference exists in our data. For area circle and circumference, students usually messed up with circle area questions, and answered using circumference. Figure 5.3 shows the common case for this pair. I followed the same procedure as for the surface area and volume skill pair, and found that out of 50 problems, among all incorrect answers, around 25% are answered using circumference.



Assignment: Assistment #277992

Assistment ID: 277992

Comment on this question

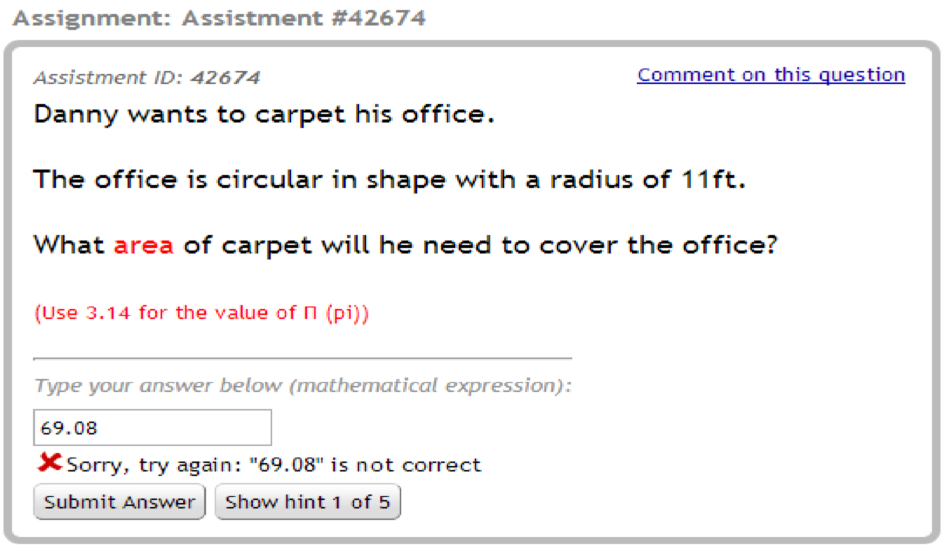
Find the surface area of a cube with a length of 4.

Type your answer below (mathematical expression):

64

✗ Sorry, try again: "64" is not correct

Figure 5.2: Common wrong answer for skill pair surface area and volume


Assignment: Assistment #42674

Assistment ID: 42674

Danny wants to carpet his office.

The office is circular in shape with a radius of 11ft.

What area of carpet will he need to cover the office?

(Use 3.14 for the value of π (pi))

Type your answer below (mathematical expression):

69.08

✗ Sorry, try again: "69.08" is not correct

Figure 5.3: Common wrong answer for skill pair area circle and circumference

From the previous results, we know that interference does exist in the data. However, it is hard to automatically detect, since the interfering pairs are usually assigned together and in the same order. So maybe a randomized control trial would be useful to gather enough data and get some useful results. Automatically detecting interfering skill pairs is certainly helpful since it would advance our understanding of learning and potentially improve ITS, one simple example is that ITS can give students hints telling them they messed up with two similar skills when interference happens.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

As described in Chapter 1, we've developed a second-generation ARRS system including the front end interface and the back end algorithms designs. We deployed the system on September 1st, 2012 and have a considerable usage by both teachers and students. The new ARRS system addressed some shortcomings with the prototype version of ARRS, specifically (Section 1.3):

1. Unclear student interface for reassessment tests and relearning
2. Teacher reports that were difficult to act on
3. Limited ability to customize ARRS for each class.

In the second part of this thesis, which includes Chapter 3, 4, and 5, I presented series of analyses on predicting student's retention performance. We've find that the class level information (class_id and class_prior_performance) are useful predictors for student's retention performance. And these two features are somehow independent as the R^2 increases when both are modeled. This work confirmed our intuition about using class level performance as a predictor of student retention-test performance. Clustering students by their math class seems useful, and are an intuitive way to cluster similar students together.

In Chapter 4, we described our study which examined and compared several different models for measuring student retention with different feature sets using data from the ARRS system. We extended Wang and Beck's study by examining each individual feature to test their real power of prediction. And we found that, the students' previous performance before the test (g_mean_performance) and the prolem_easiness are useful features to know and some features like g_mean_time is apparently useless in predicting retention performance. The mastery speed feature we introduced turns out to be the most important feature influencing both short term and long term retention performance. Our study showed that mastery speed is a helpful feature to predict students' 7-day test performance, and also has a long-term effect on students' later retention test even 105 days later.

Finally in Chapter 5, we showed our work on building an experimental infrastructure for future studies, and answered to our research questions about the best settings for the ARRS system. Basically we found that different students in different groups may benefit from certain interval settings, and for different skills, students may also need different intervals between reassessment tests. Our work on detecting interference between different skills showed that this phenomenon does exist and also impairs student's performance. The challenge in this study is that most teachers assigned the skill pair together which result in lack of data, so some further studies, like a randomized control trial, might be useful to gather enough data and run the study to detect the interference.

6.2 Future work

Currently, we already got some results on optimal ARRS settings. But we still need to run more studies to get more reliable and useful results. This task is harder than what we expected since ARRS studies take a long time to run and it would take way too long to optimize the results from randomized control trials.

We are also working on finding a way to show that ARRS is really helping students' with their knowledge retention. We are planning a "summer melt" study in the future to give us an idea about that question. In the summer melt study, we will choose a set of skills taught during April and May, students will have 50% possibility to get ARRS tests and 50% possibility not to get ARRS tests. After the summer vacation time, we will present students with a test in September to compare how well they retain the knowledge based on whether they received ARRS tests or not.

Furthermore, some other interesting questions can also be future directions of ARRS study. For example, we already know slow mastery speed is bad, but would a stronger mastery criterion, such as 4 or 5 questions right in a row, be helpful? We can certainly say that students who get 5 questions right in a row will do better than those who only got 3. But it's not easy to tell how much of the difference is learning versus just screening out stronger students.

Bibliography

[1] Anderson, J.R., Rules of the Mind. Lawrence Erlbaum (1993).

[2] Beck, J.E., et al. Predicting student help-request behavior in an intelligent tutor for reading. Ninth International Conference on User Modeling, (2003), Johnstown, PA.

[3] Gong, Y., Beck, J.E. (2011). Items, Skills, and Transfer Models: Which Really Matters for Student Modeling? In Proceedings of the 4th International Conference on Educational Data Mining, pp. 81–90

[4] Wang, Y., & Beck, J. E. (2012). Using Student Modeling to Estimate Student Knowledge Retention. In Proceedings of the 5th International Conference on Educational Data Mining, 176-179.

[5] Pavlik, P. I., Cen, H. & Koedinger, K. (2009) Performance Factors Analysis - A New Alternative to Knowledge. Proceedings of the 14th International Conference on Artificial Intelligence in Education, Brighton, UK, pp. 531-538.

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

[7] Wilson, C. Are we asking the right questions? Understanding which tasks lead to the robust learning of the English article system. In R. Luckin, K.R. Koedinger, &amp; J. Greer (Eds.), Artificial Intelligence in Education - Building Technology Rich Learning Contexts that Work (Vol 158). 2007. Pages 709-710.

[8] Xiong, X., Li, S., Beck, J. (Accepted) Will you get it right next week: Predict delayed performance in enhanced ITS mastery cycle. The 26th International FLAIRS Conference.

[9] Xiong, X., Beck, J., Li, S. (Accepted as a short paper) Class distinctions: Leveraging class-level features to predict student retention performance. 16th International Conference on Artificial Intelligence in Education

[10] Baker, R., Gowda, S., Corbett, A., & Ocumpaugh, J. (2012). Towards automatically detecting whether student learning is shallow. In Intelligent Tutoring Systems, 444-453. Springer Berlin/Heidelberg.

[11] Song, S, Heffernan, N. Automatic Reassessment and Relearning System, http://www.wpi.edu/Pubs/E-project/Available/E-project-042710-150339/.

[12] Gong, Y., Beck, J. & Heffernan, N. (2010). Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting. Proceedings of the 10th International Conference on Intelligent Tutoring Systems, 35-44.

[13] Pardos, Z. A., & Heffernan, N. T. (2010). Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. Journal of Machine Learning Research W & CP.

[14] Pardos, Z., Trivedi, S., Heffernan, N., & Sárkozy, G. (2012). Clustered knowledge tracing. In Intelligent Tutoring Systems, 405-410. Springer Berlin/Heidelberg.

[15] Wang, Y., Heffernan, N.T. 2012. The Student Skill Model. In Proceedings of the 11th International Conference on Intelligent Tutoring Systems, pp. 399-404, 2012.

[16] Beck, J.E. & Mostow, J. (2008) How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different types of students. 9th International Conference on Intelligent Tutoring Systems, Montreal, 353-362.

[17] Wang, Y. & Heffernan, N. (2011) Towards Modeling Forgetting and Relearning in ITS: Preliminary Analysis of ARRS Data. Proceedings of the 4th International Conference on Educational Data Mining. Pages 351-352.

[18] Perruchet, P. (1989), The effect of spaced practice on explicit and implicit memory. British Journal of Psychology, 80: 113–130. doi: 10.1111/j.2044-8295.1989.tb02306.x.

[19] Li, S., Xiong, X., Beck, J. (Accepted). Modeling Student Retention in an Environment with Delayed Testing. In Proceedings of the 6th International Conference on Educational Data Mining.

[20] Cain, L.F. and Willey, R.D.V, The effect of spaced learning on the curve of retention. Journal of Experimental Psychology, Vol 25(2), (Aug 1939), 209-214. doi: 10.1037/h0054640.