A Prediction Model Uses the Sequence of Attempts and Hints to Better Predict Knowledge: Better to Attempt the Problem First, Rather Than Ask for A Hint

by Linglong Zhu

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

By

____

April 2014

APPROVED:

____

Professor Neil T. Heffernan, Thesis Advisor

____

Professor Joseph Beck, Thesis Reader

____

Professor Craig Wills, Department Head

Abstract

Intelligent Tutoring Systems (ITS) have been proven to be efficient in providing students assistance and assessing their performance when they do their homework. Many research projects have been done to analyze how students' knowledge grows and to predict their performance from within intelligent tutoring system. Most of them focus on using correctness of the previous question or the number of hints and attempts students need to predict their future performance, but ignore how they ask for hints and make attempts. In this research work, we build a Sequence of Actions (SOA) model taking advantage of the sequence of hints and attempts a student needed for previous question to predict students' performance. A two step modeling methodology is put forward in the work, which is a combination of Tabling method and the Logistic Regression. We used an ASSISTments dataset of 66 students answering a total of 34,973 problems generated from 5010 questions over the course of two years. The experimental results showed that the Sequence of Action model has reliable predictive accuracy than Knowledge Tracing and Assistance Model and its performance of prediction is improved after combining with Knowledge Tracing.

Acknowledge

I would like to thank my advisor, Professor Neil T. Heffernan, who introduced me to the field of Educational Data Mining. His creative idea inspired my innovation and his spirit of skeptics improves my scholar rigorousness. Thanks Professor Heffernan’s for his advice and guidance on academics and life. It was a great experience to work with such a nice person.

Thanks my thesis reader, Professor Joseph E. Beck, for his time and valuable suggestions. Also, I would like to thank Hien D. Duong and Yutao Wang, who gave me a lot of advices and support on my research work.

Thanks to lots of friends and lab mates who help me a lot while doing the development work of ASSISTments and the educational data mining research.

Contents

List of Figures

List of Tables

Chapter 1 Background

1.1 Intelligent Tutoring Systems

Computer systems with Artificial Intelligence techniques have been used for educational purposes since the early 1960s [1]. Intelligent Tutoring Systems (ITS) are such systems that integrate computer science, cognitive psychology and educational research. It is a computer based program that emulates a “human tutor” and provides individualized instruction and assignment based on students’ performance and progress [2]. This thesis investigates relationship between students’ performance and their actions using Intelligent Tutoring Systems (ITS), we will introduce ITS briefly in this section.

Although there are many different ITS and they have different structures, a typical ITS has four basic components: Domain Model, Student Model, Tutoring Model and User Interface, as shown in Figure 1-1. Domain Model is consists of concepts, facts, rules, and problem-solving strategies of the domain in context, while Student Models emphasizes cognitive and affective states of the student as they solve the domain problems 2. Tutoring Model, also known as tutor strategies, interacts with Domain Model and Student Model. It receives students’ actions from User Interface and sends it to Student Model. Student Model makes use of the information to generate students’ cognitive and effective state and sends back to Tutoring Model, which chooses individualized tutoring strategies based on problem-solving skills from Domain Model and present to students through User Interface.

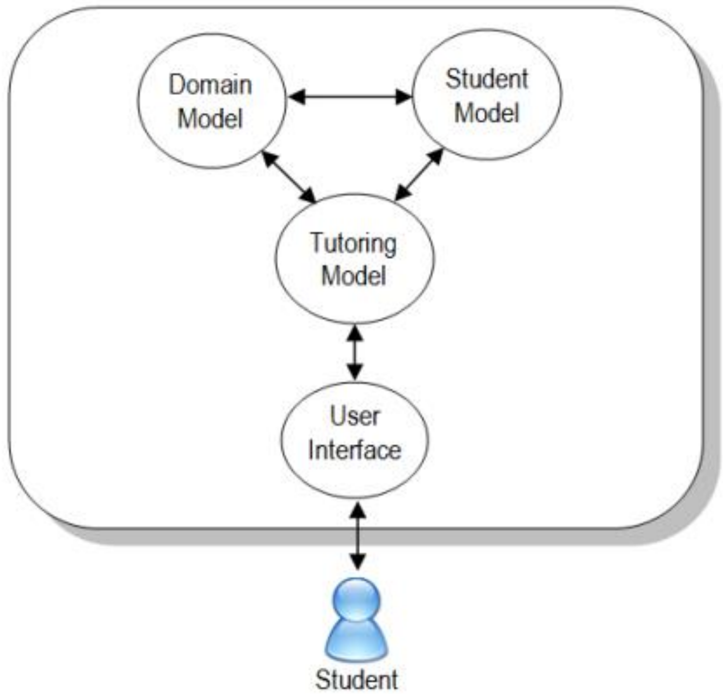

Figure 1-1Typical architecture of an Intelligent Tutoring Systems

Intelligent Tutoring Systems have been shown to be highly effective in helping students learn better. For example, Shute et al. [3] claimed that students using an ITS for economics could perform equally well as students taking a traditional economics course, but required half as much time to cover the materials [4]. Many ITSs, such as Cognitive Tutors created by Carnegie Mellon University [5], ASSISTments developed by Worcester Polytechnic Institute [6] and Wayang Outpost built by University of Massachusetts Amherst [7], have been built to improve students' learning rate. In the review of ITSs 2, Ahuja and Roohi describe the details of the ITS development history. From rule-based systems such as GUIDON [8] to buggy-based systems as BUGGY [9] and DEBUGGY [10]. From systems with natural language processing techniques SOPHIE [11] and new topics such as Pen-based tutoring systems [12] [13] and learning through games [14] [15].

1.2 State of The Art

1.2.1Student Modeling

Scholars in the society of Educational Data Mining and Intelligent Tutoring Systems have built many models to investigate the process of student learning and different aspects that affect this process. Some models based on students' performance, some models based on students' emotion, and some based on students' behavior while using the computer tutoring system.

One of the student modeling tasks is to trace the student's knowledge by using student's performance. Corbett and Anderson (1995) put forward the well-known Knowledge Tracing (KT) based on their observation that the students' knowledge is not fixed, but is assumed to be increasing [16]. KT model makes use of Bayesian network to model students' learning process and predicate their performance. KT model has been used in many different domains. For example, Jastrzembski and Gluck et al. use KT model to measure student reading proficiencies [17], and Kasurinen and Nikula use it to estimate students' learning process of programming and the three-phase measuring method they developed provides teacher a good way to assess the course contents and student performance [18].

A variety of extensions of KT model are put forward in recent years. Baker, Corbetta and Aleven builds a contextual guess and slip model based on KT that provides more accurate and reliable student modeling than KT [19]. Pardos and Heffernan extends KT four parameters model to support individualization and skill specific parameters and get better prediction of students' performance [20]. Also, they build Item Difficulty Effect Model (KT-IDEM), which incorporates item difficulty into KT by adding an item difficulty node pointing to each question node [21]. Their experimental results show that KT-IDEM model gives better performance prediction for students' skill mastery. Qiu and Qi et. al find that forgetting is more likely cognitive explanation for the over predicate of KT by considering the duration students finish their tasks [22].

Besides, many researches have been done to investigate the performance of KT. Riteer and Harris et al. reduce the parameter space of KT model by clustering which finds the smallest group of parameter sets that could model the data sufficiently well [23]. Pardos and Heffernan navigate the parameter space of KT model and find that initial parameter values leading to a degenerate state exist on a surface with predictable boundaries [24].

Some alternative methods to KT model are developed. For exmpale, in order to generate adaptive instructions for students, Pavlik Jr, Cen, and Koedinger put forward Performance Factor Analysis (PFA) Model [25] which is an adaptive version of Learning Factors Analysis (LFA) [26], and can make predictions for individual students with individual skills. Gong, Beck and Heffernan compare KT with PFA using multiple model fitting procedures [27].

1.2.2 Knowledge Prediction Model With Students' Action

In the educational data mining area, large amount of research have been done to help improve the student learning using Intelligent Tutoring Systems. Models shown in section 1.2.1Student Modeling are just a little part of it. However, not too much attention is paid to the interaction data generated when students interacts with computer tutors. In this section, we give a quick review of research related to this type of data. Shih, Koedinger and Scheines utilize Hidden Markov Model clustering to discover different strategies students used while working on a ITS and predict learning outcomes based on these strategies. Their work is based on data set consists of a series of transactions and each transaction is a <Student, Step, Action, Duration>

tuple. This model takes into account both students' action, attempt or help request, and action duration. The experimental result of their Stepwise-HMM-Cluster model shows that persistent attempts lead to better performance than hint-scaffolding strategy [28].

Many papers have shown the value of using the raw number of attempts and hints. In fact, the National Educational Technology Plan cited Feng, Heffernan and Koedinger's work [29] and the User Modeling community gave it an award for best paper for showing that the raw number of hints and attempts is informative in predicting state test scores. Wang and Heffernan built an Assistance Model (AM) and generate a performance table based on students' behavior of doing the previous question [30]. This pure data driven result without any prediction shows that students who request more assistance have lower probability to know the knowledge. Hawkins et. al. extend AM by looking at students' behavior of previous two questions [31].

These educational data mining models that utilize the number of assistance students request and the number of attempts they make to predict students' performance including Feng, Heffernan and Keodinger [32] and AM model, have ignored the sequencing of students' interaction with ITS.

Chapter 2 Introduction

2.1 Motivation

Predicting student performance is an important part of the student modeling task in Intelligent Tutoring System [33]. This problem has attracted not only the ITS and the Educational Data Mining communities but also the Machine Learning and Data Mining communities. The objective of predicting student performance is to know how the students learn (e.g., generally or narrowly), how fast they adapt to new problems [34] or if it is possible to infer the knowledge requirements to solve the problems directly from student performance data [35]. Eventually, we would like to know whether the students perform the tests or exercises correctly or with some levels of certainty. As we mentioned in Section 1.2 State of The Art, many models and techniques have been used to model and investigate students' performance. However, not too much attention is paid to the temporally sequential actions of student when interacting with the tutoring system.

Statistic data extracted from action logs of an Intelligent Tutoring System, such as the number of hint request and the number of next question correct, are the basic data set for experiments. This extraction removed many intrinsic features of students' behaviors such as the interaction sequence. Consider a thought experiment. Suppose you know that Bob Smith asked for one of the three hints and makes one wrong answer before eventually getting the question correct. What if someone told you that Bob first made an attempt then had to ask for a hint compared to him first asked for a hint and then made a wrong attempt. Would this information add value to your ability to predict whether Bob will get the next question correct? We suspected that a student who first makes an attempt tends to learn by himself and has higher probability to master the knowledge and answer the next same question correct.

2.2 Problem Definition

ASSISTments is an online tutoring system for K12 students that gives immediate feedback to teachers, students and parents. The ASSISTments gives tutorial assistance if a student makes a wrong attempt or asks for help. Figure 1 shows an example of a hint, which is one type of assistance. Other types of assistance include scaffolding questions and context-sensitive feedback messages, known as “buggy messages”.

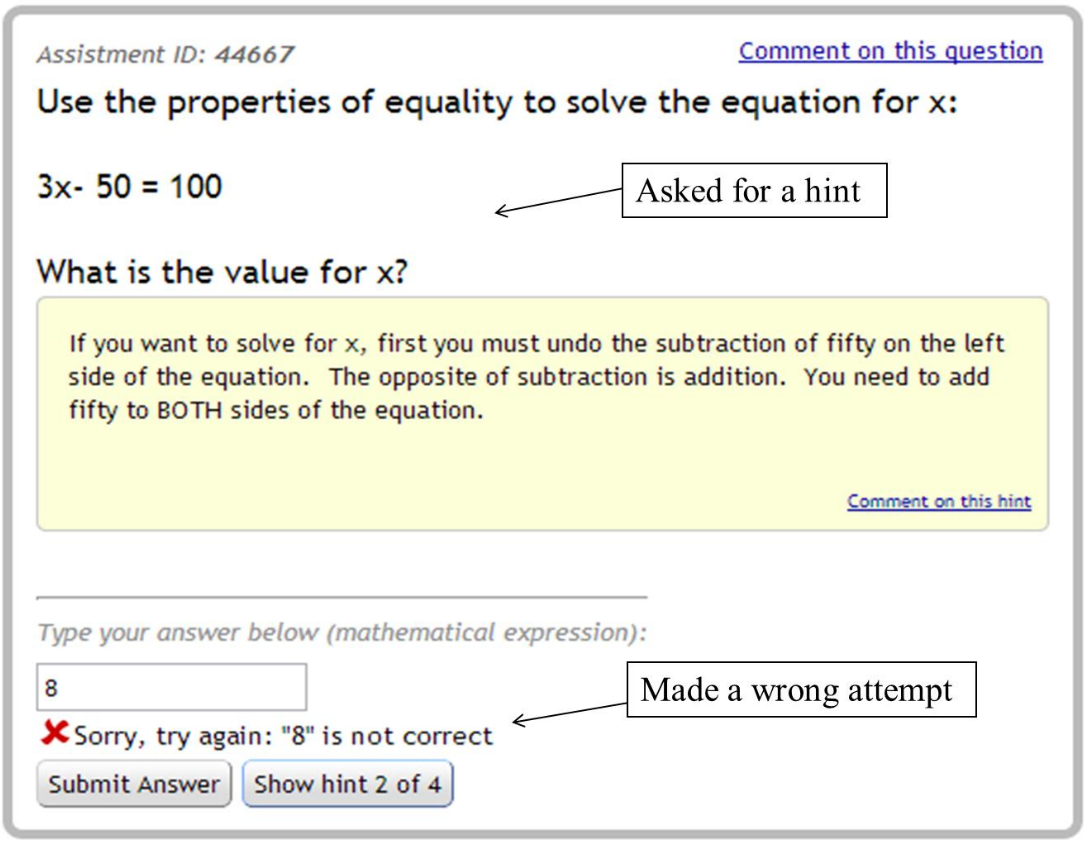

Figure 2-1Assistance in ASSISTments. Which is first: asking for a hint or make an attempt?

Figure 2-1 shows a student who asked for a hint (shown in yellow and also indicated by the button says “Show hint 2 of 4”), but it also show that the student typed in eight and got feedback that that was wrong. Though Figure 1 shows the number of hints and attempts, but interested, you cannot tell whether the student asked a hint first or made an attempt first. This papers argument is that information is very important and prediction of students learning.

ASSISTments records all of details about how a student does his or her homework and test, from which scientists can get valuable material to investigate students’ behavior and their learning process. These records include the start time and end time of a student does a problem, the time interval between a student makes an attempt and he or she asks for a hint, the number of attempts a student makes and the number of hints a student asks, as well as the answer and result for each attempt a student makes.

As mentioned in Section 2.1 Motivation, in the educational data mining society, only few attentions are paid to sequence of students’ action when interact with Intelligent Tutoring System, which we call sequence of action for simplification. In this thesis, we are going to investigate:

(1) What sequence of action might occur during students doing their homework and is there any pattern exist among all different kinds of action sequence.

(2) Whether the sequence of action has influence on students’ performance or if the sequence of action shows any information about students’ learning.

(3) How does sequence of student’s action perform in predicting students’ performance. i.e. next question correct percentage.

2.3 Goals Achieved

The goal of this research work is to prove sequential information students' action on Intelligent Tutoring System conveys information about students' learning and the sequence of action information can improve the accuracy of students' performance.

Students' actions can be very difference from each other, but we find there are some patterns existing. We divide students' sequence of action into five categories: only one attempt, all hints, all attempts, mix of hints and attempts but hint first, mix of hints and attempts but attempt first. Using a tabling method based on students' data without prediction, we found that students who make only one attempt have the highest next question correct percentage. This is reasonable since these students have already mastered these skills. But interestingly, students who make all attempts have better performance on next questions of the same skill than those who make all hints. Among actions consists of mixing of hints and attempts, students who make an attempt first have higher next percentage correct. This result proves that that students who make an attempt first before ask for a hint did better on next question with the same skill than those who ask for a hint first.

Except for the discovery that sequence of action does convey some information about students' learning, we create a logistic model SOA useing the sequence of action information to predict students' performance on the next question of the same skill. The experimental results on both student level and skill level shows that Sequence of Action (SOA) model has higher prediction accuracy than Knowledge Tracing model. Furthermore, we compared SOA with Wang and Heffernan's Assistance Model (AM) 30 which uses the number of hints students ask and the number of attempts students make. The experimental result shows that SOA predicts more accurate than AM, which indicts that the sequential information of action does convey more information about students' learning than the statistics information of actions students make.

2.4 Chapter Overview

This document is organized as follows:

In Chapter 1, we briefly introduced the background and infrastructure of Intelligent Tutoring System and the related work of student modeling and students' performance prediction based on sequence of action when students interact with ITS.

In Chapter 2, we descripted the motivation of this research work and the identified tasks of this thesis following by the goals we have achieved.

In Chapter 3, we present the data set we use for this research work and the Sequence of Action (SOA) model including the five categories of action sequencing, discovery based on a tabling method, a logistic regression model and the ensemble of SOA and Knowledge Tracing (KT) model. In the end, we describe the experiments of SOA model on predicting students' performance and compare it with KT and ensemble of SOA and KT model.

In Chapter 4, we compare performance of SOA model with Wang and Heffernan's Assistance Model (AM) using the same data set as experiments in chapter 3.

In Chapter 5, before we give an outlook in this area and suggest some work for the future, we summarize the models and methods we put forward in this thesis work.

Chapter 3 Sequence of Action Model

3.1 Students' Actions in ASSISTments

The data we used in the analysis presented in this paper came from ASSISTments, a free public service of WPI funded by federal and foundation grants. ASSISTments is an online tutoring system for K-12 students that gives immediate feedback to teachers, students, school administrators, and parents. ASSISTments gives tutorial assistance if a student makes a wrong attempt or asks for help. For some questions, when a student types a wrong answer, a set of scaffolding questions come out, which are based on the steps required solve the original question.

In Figure 3-1 of calculate angle A given supplementary angle of BCA, the student typed a wrong answer which triggers problems in Figure 3-2 and Figure 3-3. The student correctly answered the first scaffolding question, degree of angle BCA, but still have no idea about how to solve the second scaffolding question about the angle of A, then he asked for three hints before type the correct answer.

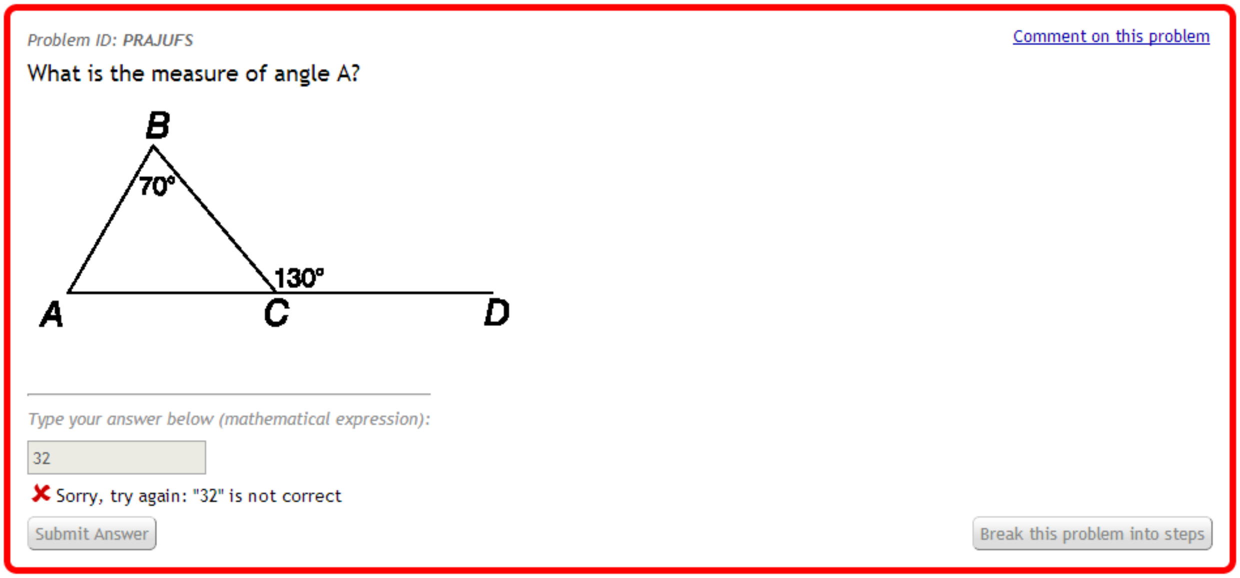

Figure 3-1 Example of ASSISTment Problem: make a wrong attempt

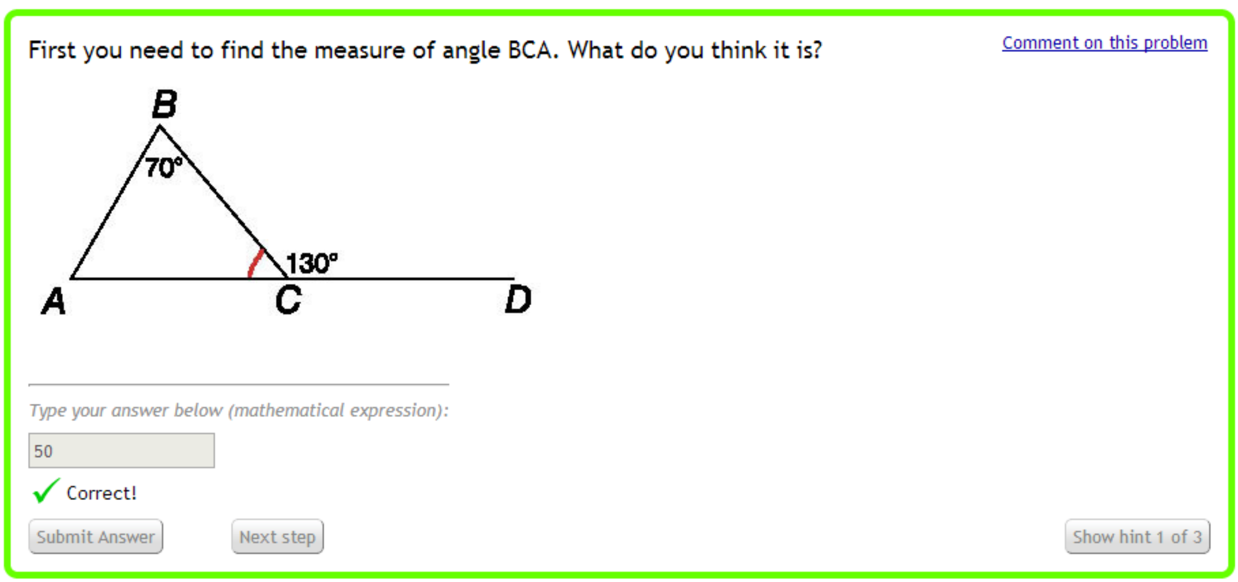

Figure 3-2 Example of ASSISTment Problem: make a correct attempt

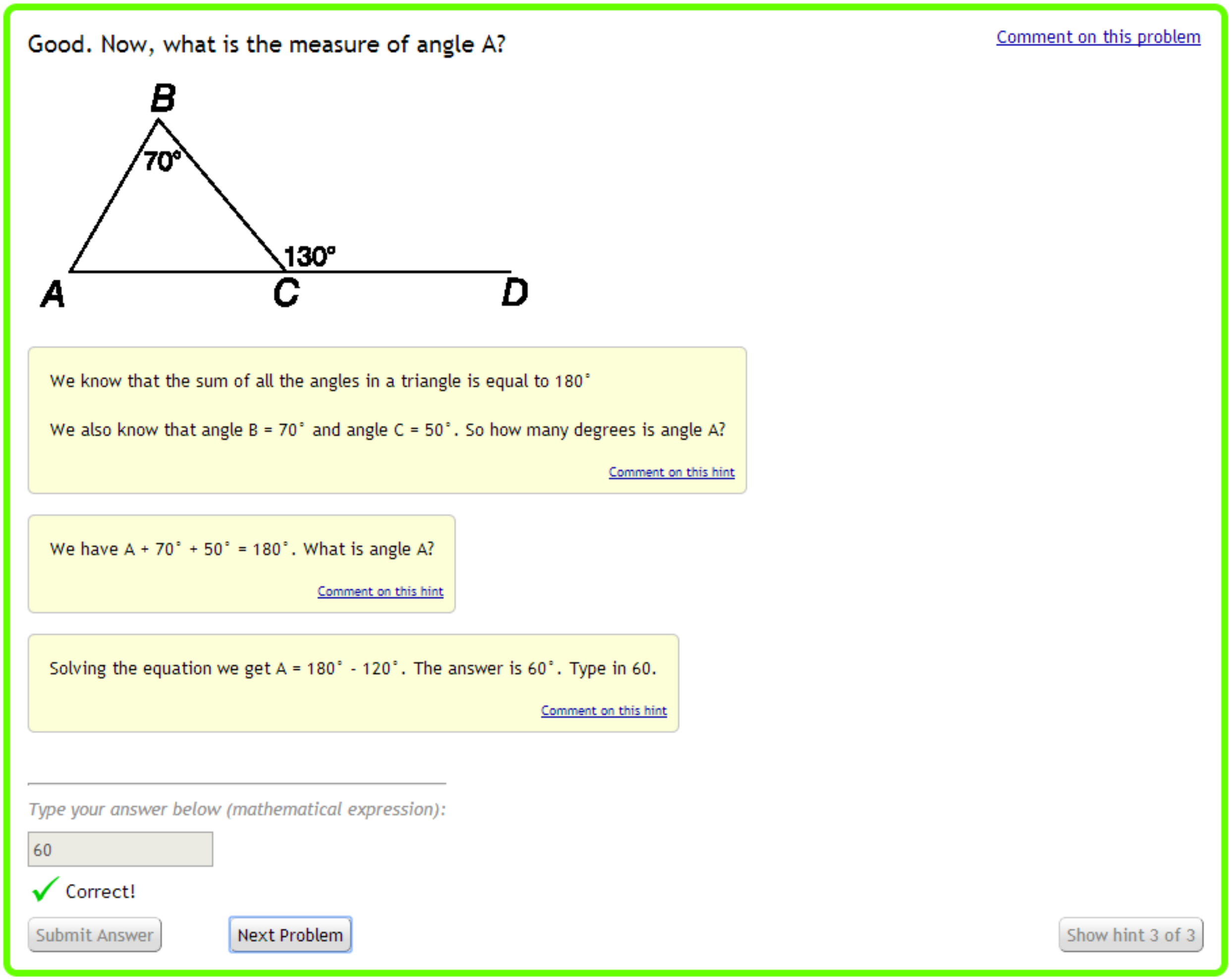

Figure 3-3 Example of ASSISTment Problem: ask for three hints and type the answer

ASSISTments records all of details about how a student does his or her homework and test, from which scientists can get valuable material to investigate students' behavior and their learning process. These records include the start time and end time of a student does a problem, the time interval between a student makes an attempt and he or she asks for a hint, when and what attempts a student makes and the time that a student asks for a hint, as well as the answer and result for each attempt a student makes. Especially, some example records of students' action are shown in Figure 3-4 Students' action records in ASSISTments.

Assignment: 17 - Adding and Subtracting Proper Fractions 5.NF.A.1

Time	Action	Object ID / Input text
Wed Feb 29 13:16:00 -0500 2012	Started a problem	PRAQZ8J
0 mins 4 secs	Asked for a hint	
0 mins 3 secs	Asked for a hint	
0 mins 1 secs	Asked for a hint	
Wed Feb 29 13:22:02 -0500 2012	Resumed a problem	PRAQZ8J
0 mins 12 secs	Answered	5/6
Wed Feb 29 13:22:17 -0500 2012	Started a problem	PRAQZ6Y
1 mins 19 secs	Answered	1/2
Wed Feb 29 13:28:34 -0500 2012	Resumed a problem	PRAQZ6Y
0 mins 32 secs	Asked for a hint	
2 mins 0 secs	Answered	63
0 mins 6 secs	Asked for a hint	
0 mins 4 secs	Asked for a hint	
0 mins 13 secs	Answered	4/63
Wed Feb 29 13:33:21 -0500 2012	Started a problem	PRAQZ4V
1 mins 39 secs	Answered	4/10]
0 mins 4 secs	Answered	4/10
2 mins 25 secs	Answered	4/10
0 mins 7 secs	Answered	2
0 mins 1 secs	Answered	2
0 mins 20 secs	Answered	4 2/10
0 mins 1 secs	Answered	4 2/10
0 mins 11 secs	Asked for a hint	
0 mins 5 secs	Asked for a hint	
0 mins 6 secs	Asked for a hint	
0 mins 9 secs	Answered	2/5
Wed Feb 29 13:38:31 -0500 2012	Started a problem	PRAQ2AR
0 mins 39 secs	Answered	1/2

Figure 3-4 Students' action records in ASSISTments

In student star report, shown in Figure 3-4 Students' action records in ASSISTments, the row in blue means that the answer is correct, the row in red means that the answer is wrong, and the orange row means the student asked for a hint. Figure 3-4 Students' action records in ASSISTments is an example of student star report, in which we can see that this student Tom (fake name) asked three hints continuously for the first problem PRAQZ8J, and stopped doing the homework, and resumed 6 minutes later, its sequence of action is 'hhha'. For the second problem PRAQZ6Y, he alternatively made attempts and asked for hints, its sequence of action is 'ahahha'. For the third question PRAQZ4V, he made 7 attempts before asked for hints and its action sequencing is 'aaaaaaahhha'. One interesting observation is that the time he spent on the first and third attempts is much longer than others. We guess he spent some time thinking about the answer, but for the second wrong answer, he was kind of gaming the system by submitting the same answer twice even though they are wrong. Tom spent some time thinking about how to solve the problem Fortunately, he answer the last question PRAQ2AR correctly with only one attempt and its action sequencing is 'a'.

3.2 DataSet

We used data from one Mastery Learning classes. Mastery Learning is a strategy that requires students to continually work on a problem set until they have achieved a preset criterion (typically three consecutive correct answers). Questions in each problem set are generated randomly from several templates and there is no problem-selection algorithm used to choose the next question.

Sixty-six 12-14 year-old, 8th grade students participated in these classes and generated 34,973 problem logs during a two year period, from 09/2010 to 07/2012. The correctness of each answer was logged, as well as the sequence of students' action, number of hints required and the number of attempts made to answer each question. We only used data from a problem set for a given student if they had reached the mastery criterion. This data was collected in a suburban middle school in central Massachusetts. Students worked on these problems in a special "math lab" period, which was held in addition to their normal math class.

In the data, all of problems were in tutor mode, which means that the students can get assistance. If the students are in the test mode, there is no feedback or assistance at all. If a problem only has one hint, the hint is the answer of the problem and is called the bottom hint. After a student asks for a bottom hint, any other attempt is meaningless because he or she already knows the answer. In the experiment, we only consider the problem logs that have at least two hints. And the answer will be marked as incorrect if students ask for a hint or the first attempt is incorrect. Moreover, we excluded such problem logs like: 1) students quit the system immediately after they saw the question and the action logs were blank or 2) after they requested hints, but did not making any attempts and no answer was recorded.

Here we only consider the question pair that have the same skill and skills having only one question were removed because they do not help in predicting. Questions of the same skills were sorted by start time in ASSISTments. We split equally 66 students into six groups, 11 students in each, to run 6-fold cross validation. We trained the SOA model and the KT model on the data from five of the groups and then did the prediction accuracy on the sixth group. We did this for all six groups.

3.2 Sequence of Action Model (SOA)

In this section, we put forward Sequence of Action (SOA) model, which takes advantage of the order information about how students make attempts and ask for hints. Section 3.1 shows a naïve model built based on the tabling method (Wang, Pardos and Heffernan2011). Section 3.2 describes the logistic model for SOA.

Different students have different sequence of actions. Some students answered correctly only after one attempt and some students kept trying many times. Some students asked for hints and made attempts alternatively, which we believe that they were learning by themselves. In the data, there are 217 different sequences of actions. Intuitively, students' actions reflect their study attitude, which decides their performance. Based on the assumption that students who make more attempts are tend to master knowledge better than students who ask for more hints, we divided them into five categories or bins: (1) One Attempt: the student correctly answered the question after one attempt; (2) All Attempts: the student made many attempts before finally get the question correct; (3) All Hints: the student only asked for hints without any attempts at all; (4) Alternative, Attempt First: the students asked for hints and made attempts alternatively and made an attempt at first; (5) Alternative, Hint First: the students asked for hint and made attempts alternatively and asked for a hint first. Table 1 shows the division and some examples of the action sequences in each category.

Table 3-1 Sequence of Action Category and Examples

Sequence of Action Category/ Bin Name	Examples
One Attempt/Bin 'a'	a
All Attempts/Bin 'a+'	aa, aaa, …, aaaaaaaaaaaa
All Hints/Bin 'h+'	ha, hha,…, hhhhhhha

Alternative, Attempt First/Bin ‘a-mix’	aha, aahaaha,…, aahhhhaaa
Alternative, Hint First/Bin ‘h-mix’	haa, haha,…, hhhhaha

Notice that each sequence ends with an attempt because in ASSISTments, a student cannot continue to next question unless he or she fills in the right answer of the current problem. In Table 1, ‘a’ stands for answer and ‘h’ stands for hint. An action sequence “ahha” means that a student makes an attempt and then asks for two hints before he or she types the correct answer and move on to the next question.

3.3 Tabling Method

After divide all of sequence of actions into five categories, we use a Tabling method, which gets the next percent correct directly from the data without any assumption or prediction. In the six-fold experiment, 66 students are divided into 6 groups. In each fold, five groups of students are used as training group and the other group is used as test group. During the six-fold experiment, each group is used as test group once. For each fold, one table is generated by the tabling method by counting the number of total appearance and the number of next correct of each bin. After counting, a next correct percent is calculated by dividing *Next Correct Count* by *Total Count of Bin*.

Table 3-2 Next correct percent table of training group of first fold

Sequence of Action Category/Bin Name	Total Count of Bin	Next Correct Count	Next Correct Percent
One Attempt/Bin ‘a’	22964	19157	0.834219
All Attempts/Bin ‘a+’	3538	2690	0.760317

All Hints/Bin ‘h+’	335	172	0.513433
Alternative, Attempt First/Bin ‘a-mix’	2030	1318	0.649261
Alternative, Hint First/Bin ‘h-mix’	72	37	0.513889

Table 3-3 Next correct percent table of training group of second fold

Sequence of Action Category/Bin Name	**Total Count of Bin**	**Next Correct Count**	**Next Correct Percent**
One Attempt/Bin ‘a’	22995	19167	0.833529
All Attempts/Bin ‘a+’	3589	2741	0.763722
All Hints/Bin ‘h+’	360	167	0.463889
Alternative, Attempt First/Bin ‘a-mix’	2040	1285	0.629902
Alternative, Hint First/Bin ‘h-mix’	70	30	0.428571

Table 3-4 Next correct percent table of training group of third fold

Sequence of Action Category/Bin Name	**Total Count of Bin**	**Next Correct Count**	**Next Correct Percent**
One Attempt/Bin ‘a’	22918	19042	0.830875
All Attempts/Bin ‘a+’	3565	2740	0.768583
All Hints/Bin ‘h+’	376	179	0.476064
Alternative, Attempt First/Bin ‘a-mix’	2101	1329	0.632556

Alternative, Hint First/Bin 'h-mix'	80	37	0.4625

Table 3-5 Next correct percent table of training group of fourth fold

Sequence of Action Category/Bin Name	Total Count of Bin	Next Correct Count	Next Correct Percent
One Attempt/Bin 'a'	22933	19113	0.833428
All Attempts/Bin 'a+'	3474	2660	0.765688
All Hints/Bin 'h+'	407	183	0.449631
Alternative, Attempt First/Bin 'a-mix'	2166	1364	0.629732
Alternative, Hint First/Bin 'h-mix'	88	39	0.443182

Table 3-6 Next correct percent table of training group of fifth fold

Sequence of Action Category/Bin Name	Total Count of Bin	Next Correct Count	Next Correct Percent
One Attempt/Bin 'a'	23138	19427	0.839614
All Attempts/Bin 'a+'	3392	2629	0.775059
All Hints/Bin 'h+'	396	179	0.45202
Alternative, Attempt First/Bin 'a-mix'	2063	1306	0.633059
Alternative, Hint First/Bin 'h-mix'	70	28	0.4

Table 3-7 Next correct percent table of training group of sixth fold

Sequence of Action Category/Bin Name	Total Count of Bin	Next Correct Count	Next Correct Percent
One Attempt/Bin 'a'	23412	19459	0.831155
All Attempts/Bin 'a+'	3612	2750	0.761351
All Hints/Bin 'h+'	381	185	0.485564
Alternative, Attempt First/Bin 'a-mix'	2225	1408	0.632809
Alternative, Hint First/Bin 'h-mix'	75	39	0.52

In order to see the trend clearly, we generate another table from all of these 66 students' data. From Table 3-8, we can see that the percent of next-question-correct is highest among students only using one attempt since they master the skill the best. They can correctly answer the next question with the same skill. For students in All Attempts category, they are more self-learning oriented, they try to learn the skill by making attempts over and over again. So they get the second highest next-question-correct percent. But for students in the All Hints category, they do the homework only relying on the hints. It is reasonable that they don't master the skill well or they don't even want to learn, so their next-question-correct percent is very low.

Table 3-8 Results of Tabling method for SOA model

Sequence of Action Category/Bin Name	Total Count of Bin	Next Correct Count	Next Correct Percent
One Attempt/Bin 'a'	23060	19227.5	0.833803394
All Attempts/Bin 'a+'	3528.33	2701.666667	0.765786747

All Hints/Bin ‘h+’	375.83	177.5	0.473433585
Alternative, Attempt First/Bin ‘a-mix’	2104.17	1335	0.634553139
Alternative, Hint First/Bin ‘h-mix’	75.83	35	0.461357023

The alternative sequence of action reflects students’ learning process. Intuitively, these students have positive attitude for study. They want to get some information from the hint based on which they try to solve the problem. But the results for the two alternative categories are very interesting. Though students in these two categories alternatively ask for hints and make attempts, the first action somewhat decided their learning altitude and final results. For students who make an attempt first, if they get the question wrong, they try to learn it by asking for hints. But for students who ask for a hint first, they seem to have less confidence in their knowledge. Although they also make some attempts, from the statistics of action sequence, they tend to ask for more hints than making attempts. The shortage of knowledge or the negative study attitude make their performance as bad as the students asking exclusively for hints first.

3.4 Logistic Regression Model

The Sequence of Action (SOA) model evolved out of a simple intuition that if in the same skills, students attempt to answer the question before requesting a hint might have better understanding about the problem and they might have greater chance to answer the next question correctly. That is, given a past particular sequence of actions, can we predict future performance of other students given the action sequence in the same bin? In this section, we are going to introduce the second part of Sequence of Action (SOA) model which makes use of a logistic regression model and information we get from the first part of SOA, i.e. tabling method.

Even though the next correct percentage we get from the tabling method indicates that the action of sequence can reflect the trend of next correct percentage, the table is very rough and is not intelligent to be used to predict students' performance. However, we can use that bit of information as one part of the input for our prediction model, i.e. the logistic regression prediction model.

The specific form of logistic regression model and its logit transformation, as follow:

$$\pi(\mathrm{x}) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2}}$$

$$\mathrm{g(x)} = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

The dependent variable ***Next Correct***, which has two states correct (listed in the table as one) and incorrect (a zero). The independent variables inputted into SOA logistic regression model are ***Skill_ID*** (x_1) and ***Credit*** value(x_2) . ***Skill_ID*** was treated as a categorical factor, while Credit was treated as a continuous factor. There are totally 51 skills of the data. As mentioned in 3.3 Tabling Method, there are six fold and each fold has their own next correct percentage table. We will call the next correct percentage gotten from tabling method ***Credit*** in the following description in order to distinguish it from the next correct prediction of test group. Each fold has their own input data to train the SOA logistic regression model.

Because students' performance is either correct or incorrect, we use Binary Logistic Regression in SPSS to train the logistic regression model and the following table is a sample input for the first fold. Each student is identified by a unique user id from ASSISTments. In Next Correct column, one means the student gets next question correct, and zero means wrong. Each skill is also identified by a skill id generated by ASSISTments. The Actions Sequence column shows the actual actions students made when answer questions and the bin and bin number right next to Action Sequence shows which bin the action belongs to. Credit is the next correct percentage generated by the tabling method shown Table 3-2 to Table 3-7. The last column Control indicates if corresponding row is in test group. One means the current row will be used as training data and two means it will be used as test data. In SPSS, Dependent is ***Next Correct*** column and Covariates are ***Skill_id***(Categorical) and ***Credit***. ***Control*** column is used as Selection Variable and the Rule is equals to 1.

Table 3-9 Input for SOA logistic regression model

Student User Id	Next Correct	Skill ID	Actions Sequence	Bin	Bin #	Credit	Control
98071	0	17	ahhha	a-mix	4	0.6331	1
98071	1	17	aahhhha	a-mix	4	0.6331	1
98071	0	17	a	a	1	0.8396	1
98071	0	17	ahhha	a-mix	4	0.6331	1
98071	1	17	ahhhha	a-mix	4	0.6331	1
98071	0	17	a	a	1	0.8396	1
98071	0	17	hhha	h+	3	0.452	1
98071	1	17	aaaa	a+	2	0.7751	1
98071	1	25	a	a	1	0.8396	1
98071	1	25	a	a	1	0.8396	1
…	…	…	…	…	…	…	…
110137	1	94	a	a	1	0.8396	2
110137	1	94	a	a	1	0.8396	2
110137	1	94	a	a	1	0.8396	2
110137	1	94	a	a	1	0.8396	2
110137	1	94	a	a	1	0.8396	2
110137	1	94	a	a	1	0.8396	2
110137	1	94	a	a	1	0.8396	2
110137	0	96	a	a	1	0.8396	2
110137	1	96	hha	h+	3	0.452	2
110137	1	96	a	a	1	0.8396	2

110137	1	96	a	a	1	0.8396	2
…	…	…	…	…	…	…	…

Logistic coefficients β0, β1 and β2 are fitted through Expectation Maximization of at most 20 steps. Parts of coefficients of the first fold are shown in Table 3-10. All of coefficients of first fold are shown in 6.2 Experimental Result for Logistic Regression.

Table 3-10 Coefficients of logistic regression model of the first fold

Parameters	Value
Intercept in the model: $\beta 0$	-1.679
$\beta_{1,0}$ (skill_id, 16)	0.322
$\beta_{1,1}$(skill_id 17)	-0.007
$\beta_{1,2}$(skill_id 24)	20.168
$\beta_{1,3}$ (skill_id 25)	3.098
$\beta_{1,4}$ (skill_id 26)	2.086
$\beta_{1,5}$ (skill_id 34)	-0.137
…….	……
$\beta_{1,48}$ (skill_id362)	0.642
$\beta_{1,49}$ (skill_id 368)	-0.117
$\beta_{1,50}$ (skill_id 371)	0.470
Parameter for the credit factor: $\beta 2$	3.286

3.5 Ensemble of Knowledge Tracing and Sequence of Action Model

3.5.1 Knowledge Tracing Model

Knowledge Tracing (KT) is one of the most common methods that are used to model the process of student's knowledge gaining and to predict students' performance 16. The KT models is an Hidden Markov Model (HMM) [36] with a hidden node (student knowledge node) and an observed node (student performance node). It assumes that skill has 4 parameters; two knowledge parameters and two performance parameters as shown in Figure 1.2. The two knowledge parameters are: prior and learn. The prior knowledge parameter is the probability that a particular skill was known by the student before interacting with the tutor. The learn rate is the probability that a student transits from the unlearned state to the learned state after each learning opportunity, i.e. after see a question. The two performance parameters are: guess and slip. Guess is the probability that a student will guess the answer correctly even if the skill associated with the question is in the unlearned state. Slip is the probability that a student will answer incorrectly even if he or she has mastered the skill for that question.

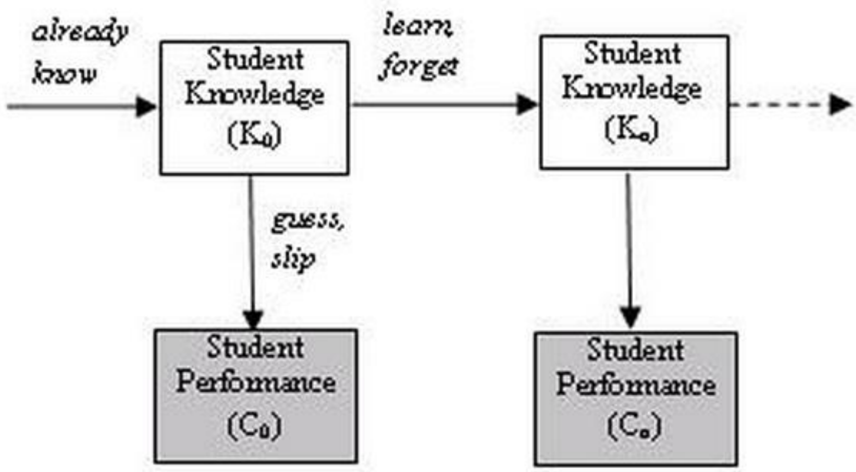

Figure 3-5 Bayesian Knowledge Tracing Model

According to Corbett and Anderson (1995) 16, the following equation is used in knowledge tracing to update the estimate of the student's knowledge state:

$$p(L_n) = p(L_{n-1}|evidence) + \left(1 - p(L_{n-1}|evidence)\right) * p(T) \tag{1.1}$$

The probability of a skill in the learned state following the nth opportunity to apply the skill, $p(L_n)$, is the sum of two probabilities: (1) the posterior probability that the skill is already in the learned state contingent on the evidence (whether or not the nth action is correct) and (2) the probability that a skill make transition to the learned state if it is not already there. Bayesian inference scheme is used here to estimate the posterior probability $p(L_{n-1}|evidence)$. Following Atkinson (1972) [37], $p(T)$ the probability of the transition from unlearned to learned state during procedural practice, which is independent of whether the student applied the skill correctly or incorrectly.

The goal of KT is to estimate the student knowledge from his or her observed actions. At each successive opportunity to apply a skill, KT updates its estimated probability that the student knows the skill, based on the skill-specific learning and performance parameters and the observed student performance (evidence). It is able to capture the temporal nature of data produced where student knowledge is changing over time. KT provides both the ability to predict future student response values, as well as providing the different states of student knowledge. For this reason, KT provides insight that makes it useful beyond the scope of simple response prediction.

The original KT does not consider multiple skills. In the experiments, we trained one KT model for each skill and the result is average of all of them.

3.5.2 Ensemble of KT and SOA

Since our SOA model is so different from the Knowledge Tracing approach, there would be a potential improvement from combing our results from SOA model with the predictions from KT model. The SOA model is based on the overall sequence of actions, and only takes the previous sequence of actions into account when predicting behavior. Whereas Knowledge

Tracing can use a longer sequence of responses and models the student's probability of knowledge while also making predictions, but doesn't look at the sequence of prior actions. We hoped that by combining two models we could produce a more accurate prediction of behavior. We took the simple average of the SOA model's prediction and KT's prediction for each response.

3.6 Experimental results

In order to evaluate the Sequence of Action model and its performance compared with KT. We ran six-fold cross validation experiments on the dataset described in Section 3.2 DataSet. Sixty-six students are randomly divided into six groups, 11 students in each group. We trained the SOA model and the KT model on the data from five of the groups and did prediction on the last group. Each of the six groups was used as test data once. Section 3.6.2 Student Level Results across the 66 students will focus on how our model is generalized across students, while Section 3.6.3 Skill Level Results across 51 Skills will report on the generalizability across skills.

To evaluate how well each of the individual models fit the data, we used three metrics to examine the predictive performance on the unseen test set: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Area Under ROC Curve (AUC). Lower values for MAE and RMSE indicate better model fit while higher values for AUC reflect a better fit.

3.6.1 Performance Prediction Results Across Six Folds

Table 3-11 shows the result of comparison for the three metrics. The values are calculated by averaging corresponding numbers obtained in the 6-fold cross validation. The prediction accuracy of each fold is shown in Section Experimental Result for Six Folds. The raw data and experiment result is available at this website: http://users.wpi.edu/~lzhu/SOA/SOA_New_Experiments.rar.

Table 3-11 Prediction Accuracy of KT, SOA and Ensemble(Avg_KT_SOA) across six folds

	MAE	RMSE	AUC
SOA	0.2900	0.3814	0.6839
KT	0.2967	0.3836	0.6768
Ensemble(KT, SOA)	0.2934	0.3812	0.6867

Although most numbers seem very close, both SOA and Ensemble (KT, SOA) outperformed KT in all three metrics. To examine whether the difference were statistically reliable, for every two models, we did a 2-tailed paired t-test based on the result from the cross validation. If the p value of t-test is less than 0.05, the result would be considered statistically significant and the null hypothesis would be rejected. The remaining degrees of freedom for the t-test is 5 in all cases. Table 3-12 below shows that the differences are significant in all three metrics.

Table 3-12 Reliability of difference of KT, SOA and Ensemble

	MAE	**RMSE**	**AUC**
KT vs SOA	**0.0000**	0.0635	0.1210
KT vs Ensemble	**0.0000**	**0.0022**	**0.0021**
SOA vs Ensemble	**0.0000**	0.5870	0.2768

The statistical test results suggest there are no significant differences between each pair of models in MAE. Also, KT and the Ensemble of KT and SOA are significantly different in all of three matrices. The p value is relatively higher in RMSE and AUC. The p-value results are not as good as expected because the t-test was conducted on accuracy of six fold, i.e. the p-value is only based on six rows of data (shown in Section Experimental Result for Six Folds).

3.6.2 Student Level Results across the 66 students

In order to better investigate the prediction performance of our model, we compute the MAE, RMSE and AUC on student level to account for the non-independence of their actions. For each model, we computed the MAE, RMSE and AUC using the actual performance of each student and their predicted performance. Predictions of students' performance are from the experimental results of all of models generated by six-fold experiments. For each student, a MAE is calculated based on all data available for that student. Then an average value for MAE is computed based on MAE of all students. The prediction accuracy shown in Table 3-13 shows that the SOA model outperforms KT model in all of three matrices. However, Ensemble of SOA and KT has more accurate prediction than SOA according to all of matrices. As a result, the students' sequence of action provides some information that KT does not have, which makes the prediction much more accurate.

Table 3-13 Prediction accuracy of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT	0.2939	0.3790	0.6738
SOA	0.2871	0.3767	0.6786
Ensemble(KT, SOA)	0.2905	0.3765	0.6811

Also, we calculate t-test p value for each pair of these three models where the remaining degrees of freedom on all the tests is 65. From the results in Table 3-14, we can see that the differences in MAE and RMSE of the different models in Table 3-13 are significantly different from each other. One exception is the p value of RMSE of SOA versus Ensemble is higher than 0.05. That is maybe due to the fact that SOA already outperforms KT so Ensemble of SOA and KT should be statistically similar.

Table 3-14 Reliability of difference of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT vs SOA	**0.0000**	**0.0000**	0.0551
KT vs Ensemble	**0.0000**	**0.0000**	**0.0000**
SOA vs Ensemble	**0.0000**	0.0698	0.0698

Note that there is no significant difference of AUC between KT and SOA. We interpret these results by pointing out that RMSE and AUC are metrics that are optimized for measuring different things, and so this is quite possible. The statistic difference among these models at student level are calculated based on prediction of all of 66 students, which means that it is much more reliable than the reliability of difference gotten from six-fold experiments directly.

3.6.3 Skill Level Results across 51 Skills

Traditional KT model is based on one skill, so we trained a KT model for each skill in the experiments. The prediction accuracy in this paper is based on the results of 51 skills in the dataset. The full table of results for the 51 skills is available at this website (http://users.wpi.edu/~lzhu/SOA/SOA_New_Experiments.rar) while Table 3-15 reports the average statistical result of three models across the three metrics. From the statistics of prediction accuracy of all of three models, we can see that SOA and Ensemble outperform KT in all of three metrics. The combination of SOA and KT is slightly better than SOA in RMSE, but is slightly worse than SOA in MAE. The AUC of all of them are not bigger than 0.5, which indicates these models do not make a good classification at skill level.

Table 3-15 Prediction accuracy of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT	0.3064	0.3762	0.4675
SOA	0.2942	0.3713	0.4769
Ensemble(KT, SOA)	0.3003	0.3710	0.492

In order to determine whether the difference between two models is statistically significant, we computed each evaluation metrics value for each skill and compared each pair of these three models using a two tailed paired t-test. The remaining degrees of freedom on all the tests are 50. The values in Table 3-16 show the statistically significant differences between corresponding pairs of models across three metrics. As shown in the table, the differences in MAE and RMSE of the different models in Table 3-15 are significantly different from each other except RMSE of SOA and Ensemble. The experimental results indicate that the sequence of action could accurately predict students' performance than KT.

Table 3-16 Reliability of difference of KT, SOA and Ensemble

	MAE	**RMSE**	**AUC**
KT vs SOA	**0.0000**	**0.0136**	0.3492
KT vs Ensemble	**0.0000**	**0.0002**	**0.0003**
SOA vs Ensemble	**0.0000**	0.3982	**0.0059**

3.7 Conclusion

In this chapter, we introduce the sequence of action model which makes use of the clicking sequence of making attempts and asking for hints when students do their homework using an Intelligent Tutoring System. Their sequence of actions are divided into five categories based on our intuition of their study enthusiasm. The data we used in the work is from 66 students doing their homework on ASSISTments. The two-steps modeling methodology is another highlight of the work. A tabling method is first used to find the correct percentage of next questions based on students' current action bin. Surprisingly, we found that students who made more attempts have higher next question correct percent than students who asked for more hints by a tabling method. Then, we built a logistic regression model to predict students' performance on next question based on their current action sequence. According to our six-fold cross validation experiments and paired two tail t-test, both on student level and skill level, our Sequence Of Action (SOA) model has reliable higher prediction accuracy than KT, especially on measure criteria MAE and RMSE. Also, we combine SOA and Knowledge Tracing (KT) using average of their prediction, and the ensemble model's prediction performance is much better than both SOA and KT. The sequence of action of students' making attempts and asking for hints reflects the level students mastering knowledge. Students who are more prone to learn a skill make more attempts spontaneously and students who ask for more hints or ask hints at first prepare to ask for help have lower inclination to learn the knowledge by themselves, which leads to the low next correct percentage.

Chapter 4 Comparing to Other Methods Using Students' Action

4.1 Assistance Model

Motivated by the intuition that students who need more assistance have lower probability possessing the knowledge, Wang and Heffernan 30 built a pure data driven "Assistance" model to disclose the relationship between assistance information and students' knowledge. A parameter table is built in which row indices represent the number of attempts a student required in the previous question and column indices represent the number of hints the student asked. Each cell contains the probability that the student will answer the current question correctly. In order to distinguish different assistance requirements, the attempts are separated into three bins: one attempt, small amount of attempts (2-5 times), large amount of attempts (more than 5 attempts) and hints are separated into four bins: no hint, small amount of hints (1, 50%], large amount of hints [50%, 100%), students ask all hints.

Table 4-1 Assistance Model parameter table, average across six folds

	attempt = 1	**0 < attempt < 6**	**attempt >=6**
hint_percent = 0	0.8410	0.7963	0.7808
0<hint_percent <=.5	0.6286	0.6933	0.6741
.5<hint_percent<1	0.4494	0.6290	0.6522
hint_percent = 1	0.4293	0.6147	0.6218

In this work, we reproduced AM parameter table and predict students' performance using data described in Section 3.2 DataSet. The experiment for computing AM parameter table is the same as that for SOA. There are 66 students evenly divided into six fold. For each fold, a parameter table is computed based on students' data in that fold. The six AM parameter tables are shown in Table 4-2 to Table 4-7.

As with Wang and Heffernan's experimental results, the average of parameters shown in Table 4-2 confirms that students requiring more assistance to solve a problem probably have less corresponding knowledge. Each cell in Table 4-2 is the average value of corresponding value in the following six parameter tables.

Table 4-2 Assistance Model parameter table of first fold

	attempt = 1	**0 < attempt < 6**	**attempt >=6**
hint_percent = 0	0.8413	0.7948	0.7692
0<hint_percent <=.5	0.6739	0.6901	0.6571
.5<hint_percent<1	0.4889	0.6474	0.7586
hint_percent = 1	0.4598	0.6346	0.6440

Table 4-3 Assistance Model parameter table of second fold

	attempt = 1	**0 < attempt < 6**	**attempt >=6**
hint_percent = 0	0.8404	0.7958	0.7831
0<hint_percent <=.5	0.625	0.6858	0.7143
.5<hint_percent<1	0.475	0.6154	0.6176
hint_percent = 1	0.4066	0.6092	0.6316

Table 4-4 Assistance Model parameter table of third fold

	attempt = 1	0 < attempt < 6	attempt >=6
hint_percent = 0	0.8376	0.7965	0.775
0<hint_percent <=.5	0.6522	0.6987	0.6190
.5<hint_percent<1	0.4333	0.6316	0.6
hint_percent = 1	0.4362	0.6143	0.6071

Table 4-5 Assistance Model parameter table of fourth fold

	attempt = 1	0 < attempt < 6	attempt >=6
hint_percent = 0	0.84097	0.7956	0.7823
0<hint_percent <=.5	0.5849	0.6890	0.6809
.5<hint_percent<1	0.4098	0.6105	0.6765
hint_percent = 1	0.4189	0.6085	0.6105

Table 4-6 Assistance Model parameter table of fifth fold

	attempt = 1	0 < attempt < 6	attempt >=6
hint_percent = 0	0.8472	0.8028	0.8039
0<hint_percent <=.5	0.62	0.7020	0.7222
.5<hint_percent<1	0.4531	0.6264	0.5938
hint_percent = 1	0.4078	0.6088	0.6198

Table 4-7 Assistance Model parameter table of sixth fold

	attempt = 1	0 < attempt < 6	attempt >=6
hint_percent = 0	0.8385	0.7925	0.7712
0<hint_percent <=.5	0.6154	0.6940	0.6512
.5<hint_percent<1	0.4364	0.6425	0.6667

hint_percent = 1	0.4463	0.6125	0.6179

4.2 Linear Regression of Assistance Model and Knowledge Tracing

In Assistance Model (AM), the prediction of students' performance on next question is the percentage value in parameter table corresponding to their current number of attempts and number of hints. According to this, we compute the prediction of AM for all of six folds based on parameter table generated for each fold. As been proved by Wang and Heffernan, AM itself does not have very high prediction accuracy compared with KT, but it does improve the prediction accuracy of KT after combined with KT. In Chapter 3, experimental results show that SOA has higher prediction accuracy than KT. In order to compare with AM model, we will compare SOA with combination of KT and AM instead of AM itself.

We use SPSS to train a linear regression of KT and AM, in which students' next correct is dependent variable and the prediction results of KT and AM in six-fold experiments are two independent variables. The regression function is:

-0.322+0.639*AM_prediction+0.769*KT_prediction;

4.3 Linear Regression of Sequence Of Action and Knowledge Tracing

In Chapter 3, we combined Sequence of Action (SOA) with Knowledge Tracing (KT) using average of prediction results of both models. But using averaging to combine the predictions of different models makes the assumption that the different models' predictions should have the same weight, which may not necessarily be the case. Inspired by Wang and Heffernan's paper 30, we also constructed a linear regression model with student performance as the dependent variable and prediction results of SOA and KT as independent variables, in order to find the best weights for the models we intend to combine. We will not combine AM and SOA model, because both of them use information about hints and attempts, which we think that the combination of them will not make a big difference. From the linear regression function we got from SPSS, we can see that SOA weights heavier than KT, which means that SOA is more useful than KT in making a prediction.

-0. 004+0. 687*SOA_prediction+0. 321*KT_prediction;

4.4 Experiments

In this section, we compare Sequence of Action (SOA) model with Assistance Model (AM) and the linear regression model of AM and Knowledge Tracing (KT) model, called as LG(AM_KT), and the linear regression model of SOA and KT model, called as LG(SOA_KT), and ensemble of SOA and KT using average, called as AVG(SOA_KT). We ran six-fold cross validation experiments on the dataset described in Section 3.1 . Sixty-six students are randomly divided into six groups, 11 students in each group. Models are trained on the data from five of the groups and prediction is done on the last group. Each of the six groups was used as test data once. Here we used three metrics to examine the predictive performance on the unseen test set: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Area Under ROC Curve (AUC). Lower values for MAE and RMSE indicate better model fit while higher values for AUC reflect a better fit.

Table 4-8 shows values of three metrics from six-fold across validation, which are calculated by averaging corresponding numbers obtained from each validation. As with Wang's results (Wang & Heffernan 2011), the performance of linear regression combination of AM and KT, LR(AM, KT) is better than AM itself, which indicates information about the number of hints and attempts improves the prediction of KT model. Overall, the combinations of any two models have higher prediction accuracy and especially, the average ensemble of SOA and KT, AVG(SOA, KT), has better accuracy than the other two combinations. Also, the linear regression of AM and KT has better prediction accuracy than linear regression combination of AM and KT. However, from the two tailed paired t-test results shown in Table 4-9, the statistically difference between any two pair of model combinations are not significant. One exception is that the linear regression of KT and SOA is statistically different from the averaging combination of KT and SOA in RMSE and AUC. This is because given values of two independent variables, the coefficient of linear regression decide the value of the dependent variable which makes a big difference in the final result.

Table 4-8 Prediction accuracy of KT, SOA, AM and Ensemble

	MAE	**RMSE**	**AUC**
AM	0.3007	0.3844	0.5795
SOA	0.2871	0.3767	0.6786
KT	0.2939	0.3790	0.6735
LR(AM, KT)	**0.2874**	0.3759	0.6824
LR(SOA, KT)	0.2878	0.3762	0.6813
AVG(SOA, KT)	0.2876	**0.3757**	**0.6836**

To examine whether there is significant difference between these models, we did a 2-tailed paired t-test based on the result from the cross validation. The p value higher than 0.5 supports the null hypothesis of no difference between prediction of two models. Table 4-9 below shows that the differences are significant between AM and any other model, KT and any other model, SOA and any other model except for SOA and linear regression of AM and KT in MAE and AUC. Both SOA and AM use the information about students' actions of hints and attempts. There might be a chance that SOA and LG(AM, KT) have some prediction overlap.

Table 4-9 Reliability of difference of KT, SOA, AM and Ensemble

	MAE	**RMSE**	**AUC**
AM vs SOA	**0.000**	**0.000**	**0.000**
AM vs KT	**0.000**	**0.000**	**0.000**
AM vs LG(AM, KT)	**0.000**	**0.000**	**0.000**
AM vs LR(SOA, KT)	**0.000**	**0.000**	**0.000**
AM vs AVG(SOA, KT)	**0.000**	**0.000**	**0.000**
SOA vs KT	**0.000**	**0.000**	**0.0367**
SOA vs LG(AM, KT)	0.2983	**0.0299**	0.0830
SOA vs LR(SOA, KT)	**0.0000**	**0.0016**	**0.0059**
SOA vs AVG(SOA, KT)	**0.0199**	**0.0000**	**0.0030**
KT vs LG(AM, KT)	**0.0000**	**0.0000**	**0.0000**
KT vs LR(SOA, KT)	**0.0000**	**0.0000**	**0.0000**
KT vs AVG(SOA, KT)	**0.0000**	**0.0000**	**0.0000**
LG(AM, KT) vs LR(SOA, KT)	0.2648	0.2961	0.4689
LG(AM, KT) vs AVG(SOA, KT)	0.2714	0.1380	0.0789
LR(SOA, KT)vs AVG(SOA, KT)	0.2584	**0.0012**	**0.0104**

4.5 Conclusion

In this chapter, we reproduce Wang and Heffernan's Assistance Model (AM) using the same data we used in Chapter 3. Also, we use the prediction of AM, SOA and KT model from six-fold cross validation to fit linear regression of AM and KT and linear regression of SOA and KT. The experimental results of these six models show that SOA has reliable higher prediction accuracy than KT and AM. However, SOA is not powerful than the combination of any two models. Among these three combinations, the average of SOA and KT has reliably highest prediction accuracy in RMSE and AUC. In sum, information of students' behavior SOA provides is different from KT and the information of sequence of action that students make attempts and ask for hints improve the prediction accuracy of KT model.

Chapter 5 Conclusion and Future Work

5.1 Conclusion

In this thesis, we put forward a Sequence Of Action (SOA) model which makes use of the clicking sequence of students making attempts and asking for hints. SOA model consists of two parts. In the first part, the sequence of students' actions are divided into five categories: only one attempt, all attempts, all hints, alternative attempts with attempt first and alternative attempts with hints first. According to the result of tabling method, students in all attempts bin have a lot the second highest next question correct percentage following students in only on attempt bins, and students in all hints have the lowest next question correct percentage. Students who make more attempts are trying to figure problems out by themselves and it is an efficient way to master knowledge than they are told the steps to answer these questions by asking for hints.

In the second part, in order to better predict students' performance, we build a logistic regression model with next question correct percentage as dependent variable and skill_id, credits of sequence of action bins as independent variables. Also, we reproduce Assistance Model (AM) and linear regression combination of AM and Knowledge Tracing (KT) using the same data with that for tabling method. We conducted six-fold cross validation experiments. The experimental results shows that SOA has reliably higher prediction accuracy than Knowledge Tracing model and Assistance Model. The average combination of the SOA and KT has the highest the prediction accuracy than other combinations. In sum, sequence of students' action provides important information about students' learning process.

5.2 Future Work

The experimental results of this work shows that sequence of students' asking for hints and making attempts reflects that students' inclination to learn and the way they. The experiment data we used is from ASSISTments, does SOA model still makes a big difference if use data from other intelligent tutor systems? Does students in high school or college have the same learning features as middle school students? If they also have this feature, how big the difference is? Also, the experiment result of this work can be used in intelligent tutor system to give students more chances to try by themselves and help them master the knowledge more efficiently. For example, a new feature called redo is designed to provide students more chance to practice without hints or feedback messages. For each problem, several redo problems with same type or skill are given on the same page. Students who answer the original problem correctly will not see the redo problems. But students answering the original problem wrong get one redo problem correctly or see all of redo problems will move on to the next question. When doing the redo problems, students are able to see the hints or help messages of the original problem on the same page so that they can figure out the redo problem by themselves.

This work is the beginning of utilizing the sequence of asking for hints and making attempts recorded by intelligent tutoring systems to better predict student performance. There are many open spaces for us to explore. For example, how much can the performance of SOA model be improved after combined with other efficient prediction model such as PFA (Pavlik et al., 2009). What is the SOA model's performance if we use a student action sequence of several previous question when train the model? How does SOA perform after individualization? These are questions that still need to be explored.

Chapter 6 Appendix

6.1 Experimental Result for Six Folds

Table 6-1 MAE of KT, SOA and Ensemble(Avg_KT_SOA) across six folds

	SOA_MAE	KT_MAE	Avg_KT_SOA_MAE
Fold1	0.294798	0.300806	0.297802
Fold2	0.29093	0.296151	0.29354
fold3	0.285727	0.293408	0.289568
Fold4	0.283729	0.290946	0.287337
Fold5	0.30654	0.312822	0.309681
Fold6	0.278536	0.286022	0.282279
Average of six folds	0.290043	0.296692	0.293368

Table 6-2 RMSE of KT, SOA and Ensemble(Avg_KT_SOA) across six folds

	SOA_RMSE	KT_RMSE	Avg_KT_SOA_RMSE
fold1	0.383912	0.386793	0.384212
fold2	0.379632	0.380606	0.378564
fold3	0.37162	0.374765	0.371822
fold4	0.37732	0.378839	0.376785
fold5	0.408328	0.407232	0.406334
fold6	0.367843	0.373198	0.369252
Average of six folds	0.381442	0.383572	0.381161

Table 6-3 AUC of KT, SOA and Ensemble(Avg_KT_SOA) across six folds

	SOA_AUC	KT_AUC	Avg_KT_SOA_AUC
fold1	0.707155	0.698946	0.71059
fold2	0.691061	0.687691	0.695278
fold3	0.693929	0.686278	0.698194
fold4	0.673011	0.671375	0.678617
fold5	0.639882	0.64233	0.64721
fold6	0.698544	0.673937	0.690314
Average of six folds	0.68393	0.67676	0.6867

6.2 Experimental Result for Logistic Regression

Case Processing Summary

Unweighted Cases[a]		N	Percent
Selected Cases	Included in Analysis	28939	82.7
	Missing Cases	0	.0
	Total	28939	82.7
	Unselected Cases	6034	17.3
	Total	34973	100.0

a. If weight is in effect, see classification table for the total number of cases.

Dependent Variable Encoding

Original Value	Internal Value
0	0
1	1

Block 1: Method = Enter

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients				
			Constant	skill_id(1)	skill_id(2)	skill_id(3)	skill_id(4)
Step 1	1	26967.562	-1.169	.237	-.011	1.054	.978
	2	26396.604	-1.631	.316	-.008	2.103	1.891
	3	26366.716	-1.678	.322	-.007	3.143	2.615
	4	26365.669	-1.679	.322	-.007	4.159	3.002

5	26365.580	-1.679	.322	-.007	5.164	3.094
6	26365.554	-1.679	.322	-.007	6.166	3.098
7	26365.545	-1.679	.322	-.007	7.167	3.098
8	26365.541	-1.679	.322	-.007	8.167	3.098
9	26365.540	-1.679	.322	-.007	9.168	3.098
10	26365.540	-1.679	.322	-.007	10.168	3.098
11	26365.539	-1.679	.322	-.007	11.168	3.098
12	26365.539	-1.679	.322	-.007	12.168	3.098
13	26365.539	-1.679	.322	-.007	13.168	3.098
14	26365.539	-1.679	.322	-.007	14.168	3.098
15	26365.539	-1.679	.322	-.007	15.168	3.098
16	26365.539	-1.679	.322	-.007	16.168	3.098
17	26365.539	-1.679	.322	-.007	17.168	3.098
18	26365.539	-1.679	.322	-.007	18.168	3.098
19	26365.539	-1.679	.322	-.007	19.168	3.098
20	26365.539	-1.679	.322	-.007	20.168	3.098

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients					
		skill_id(5)	skill_id(6)	skill_id(7)	skill_id(8)	skill_id(9)	skill_id(10)
Step 1	1	.880	-.142	.450	.274	.721	.266
	2	1.601	-.140	.653	.369	1.190	.359
	3	1.994	-.137	.679	.375	1.334	.366
	4	2.082	-.137	.679	.375	1.343	.366
	5	2.086	-.137	.679	.375	1.343	.366
	6	2.086	-.137	.679	.375	1.343	.366

	7	2.086	-.137	.679	.375	1.343	.366
	8	2.086	-.137	.679	.375	1.343	.366
	9	2.086	-.137	.679	.375	1.343	.366
	10	2.086	-.137	.679	.375	1.343	.366
	11	2.086	-.137	.679	.375	1.343	.366
	12	2.086	-.137	.679	.375	1.343	.366
	13	2.086	-.137	.679	.375	1.343	.366
	14	2.086	-.137	.679	.375	1.343	.366
	15	2.086	-.137	.679	.375	1.343	.366
	16	2.086	-.137	.679	.375	1.343	.366
	17	2.086	-.137	.679	.375	1.343	.366
	18	2.086	-.137	.679	.375	1.343	.366
	19	2.086	-.137	.679	.375	1.343	.366
	20	2.086	-.137	.679	.375	1.343	.366

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients					
		skill_id(11)	skill_id(12)	skill_id(13)	skill_id(14)	skill_id(15)	skill_id(16)
Step 1	1	-.568	.624	.690	.803	.034	-.284
	2	-.623	.990	1.123	1.400	.060	-.309
	3	-.624	1.075	1.243	1.654	.064	-.308
	4	-.624	1.078	1.249	1.687	.064	-.307
	5	-.624	1.078	1.249	1.687	.064	-.307
	6	-.624	1.078	1.249	1.687	.064	-.307
	7	-.624	1.078	1.249	1.687	.064	-.307
	8	-.624	1.078	1.249	1.687	.064	-.307

	9	-.624	1.078	1.249	1.687	.064	-.307
	10	-.624	1.078	1.249	1.687	.064	-.307
	11	-.624	1.078	1.249	1.687	.064	-.307
	12	-.624	1.078	1.249	1.687	.064	-.307
	13	-.624	1.078	1.249	1.687	.064	-.307
	14	-.624	1.078	1.249	1.687	.064	-.307
	15	-.624	1.078	1.249	1.687	.064	-.307
	16	-.624	1.078	1.249	1.687	.064	-.307
	17	-.624	1.078	1.249	1.687	.064	-.307
	18	-.624	1.078	1.249	1.687	.064	-.307
	19	-.624	1.078	1.249	1.687	.064	-.307
	20	-.624	1.078	1.249	1.687	.064	-.307

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients					
		skill_id(17)	skill_id(18)	skill_id(19)	skill_id(20)	skill_id(21)	skill_id(22)
Step 1	1	.358	.407	.795	.852	.302	.356
	2	.500	.578	1.364	1.525	.413	.497
	3	.514	.597	1.585	1.859	.422	.511
	4	.514	.597	1.609	1.919	.422	.511
	5	.514	.597	1.609	1.921	.422	.511
	6	.514	.597	1.609	1.921	.422	.511
	7	.514	.597	1.609	1.921	.422	.511
	8	.514	.597	1.609	1.921	.422	.511
	9	.514	.597	1.609	1.921	.422	.511
	10	.514	.597	1.609	1.921	.422	.511

	11	.514	.597	1.609	1.921	.422	.511
	12	.514	.597	1.609	1.921	.422	.511
	13	.514	.597	1.609	1.921	.422	.511
	14	.514	.597	1.609	1.921	.422	.511
	15	.514	.597	1.609	1.921	.422	.511
	16	.514	.597	1.609	1.921	.422	.511
	17	.514	.597	1.609	1.921	.422	.511
	18	.514	.597	1.609	1.921	.422	.511
	19	.514	.597	1.609	1.921	.422	.511
	20	.514	.597	1.609	1.921	.422	.511

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients					
		skill_id(23)	skill_id(24)	skill_id(25)	skill_id(26)	skill_id(27)	skill_id(28)
Step 1	1	.822	.346	.429	.601	.525	.873
	2	1.432	.476	.619	.941	.777	1.586
	3	1.690	.488	.642	1.014	.815	1.975
	4	1.724	.488	.642	1.016	.815	2.060
	5	1.724	.488	.642	1.016	.815	2.064
	6	1.724	.488	.642	1.016	.815	2.064
	7	1.724	.488	.642	1.016	.815	2.064
	8	1.724	.488	.642	1.016	.815	2.064
	9	1.724	.488	.642	1.016	.815	2.064
	10	1.724	.488	.642	1.016	.815	2.064
	11	1.724	.488	.642	1.016	.815	2.064
	12	1.724	.488	.642	1.016	.815	2.064

	13	1.724	.488	.642	1.016	.815	2.064
	14	1.724	.488	.642	1.016	.815	2.064
	15	1.724	.488	.642	1.016	.815	2.064
	16	1.724	.488	.642	1.016	.815	2.064
	17	1.724	.488	.642	1.016	.815	2.064
	18	1.724	.488	.642	1.016	.815	2.064
	19	1.724	.488	.642	1.016	.815	2.064
	20	1.724	.488	.642	1.016	.815	2.064

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients					
		skill_id(29)	skill_id(30)	skill_id(31)	skill_id(32)	skill_id(33)	skill_id(34)
Step 1	1	.455	.066	.152	.389	-.031	.044
	2	.661	.084	.194	.546	-.032	.060
	3	.688	.085	.196	.562	-.031	.062
	4	.688	.085	.196	.562	-.031	.062
	5	.688	.085	.196	.562	-.031	.062
	6	.688	.085	.196	.562	-.031	.062
	7	.688	.085	.196	.562	-.031	.062
	8	.688	.085	.196	.562	-.031	.062
	9	.688	.085	.196	.562	-.031	.062
	10	.688	.085	.196	.562	-.031	.062
	11	.688	.085	.196	.562	-.031	.062
	12	.688	.085	.196	.562	-.031	.062
	13	.688	.085	.196	.562	-.031	.062
	14	.688	.085	.196	.562	-.031	.062

	15	.688	.085	.196	.562	-.031	.062
	16	.688	.085	.196	.562	-.031	.062
	17	.688	.085	.196	.562	-.031	.062
	18	.688	.085	.196	.562	-.031	.062
	19	.688	.085	.196	.562	-.031	.062
	20	.688	.085	.196	.562	-.031	.062

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients					
		skill_id(35)	skill_id(36)	skill_id(37)	skill_id(38)	skill_id(39)	skill_id(40)
Step 1	1	.496	.707	-.200	.377	.650	-.874
	2	.733	1.171	-.224	.530	1.039	-.900
	3	.768	1.313	-.224	.546	1.134	-.898
	4	.768	1.322	-.223	.546	1.138	-.898
	5	.768	1.322	-.223	.546	1.138	-.898
	6	.768	1.322	-.223	.546	1.138	-.898
	7	.768	1.322	-.223	.546	1.138	-.898
	8	.768	1.322	-.223	.546	1.138	-.898
	9	.768	1.322	-.223	.546	1.138	-.898
	10	.768	1.322	-.223	.546	1.138	-.898
	11	.768	1.322	-.223	.546	1.138	-.898
	12	.768	1.322	-.223	.546	1.138	-.898
	13	.768	1.322	-.223	.546	1.138	-.898
	14	.768	1.322	-.223	.546	1.138	-.898
	15	.768	1.322	-.223	.546	1.138	-.898
	16	.768	1.322	-.223	.546	1.138	-.898

	17	.768	1.322	-.223	.546	1.138	-.898
	18	.768	1.322	-.223	.546	1.138	-.898
	19	.768	1.322	-.223	.546	1.138	-.898
	20	.768	1.322	-.223	.546	1.138	-.898

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients					
		skill_id(41)	skill_id(42)	skill_id(43)	skill_id(44)	skill_id(45)	skill_id(46)
Step 1	1	.116	.209	-.055	.434	-.302	.089
	2	.161	.276	-.058	.624	-.343	.111
	3	.165	.281	-.056	.647	-.343	.112
	4	.165	.281	-.056	.647	-.343	.112
	5	.165	.281	-.056	.647	-.343	.112
	6	.165	.281	-.056	.647	-.343	.112
	7	.165	.281	-.056	.647	-.343	.112
	8	.165	.281	-.056	.647	-.343	.112
	9	.165	.281	-.056	.647	-.343	.112
	10	.165	.281	-.056	.647	-.343	.112
	11	.165	.281	-.056	.647	-.343	.112
	12	.165	.281	-.056	.647	-.343	.112
	13	.165	.281	-.056	.647	-.343	.112
	14	.165	.281	-.056	.647	-.343	.112
	15	.165	.281	-.056	.647	-.343	.112
	16	.165	.281	-.056	.647	-.343	.112
	17	.165	.281	-.056	.647	-.343	.112
	18	.165	.281	-.056	.647	-.343	.112

	19	.165	.281	-.056	.647	-.343	.112
	20	.165	.281	-.056	.647	-.343	.112

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Iteration History[a,b,c,d]

Iteration		Coefficients				
		skill_id(47)	skill_id(48)	skill_id(49)	skill_id(50)	partial_credit
Step 1	1	-1.341	.431	-.093	.333	2.558
	2	-1.398	.620	-.116	.459	3.223
	3	-1.396	.642	-.117	.470	3.284
	4	-1.396	.642	-.117	.470	3.286
	5	-1.396	.642	-.117	.470	3.286
	6	-1.396	.642	-.117	.470	3.286
	7	-1.396	.642	-.117	.470	3.286
	8	-1.396	.642	-.117	.470	3.286
	9	-1.396	.642	-.117	.470	3.286
	10	-1.396	.642	-.117	.470	3.286
	11	-1.396	.642	-.117	.470	3.286
	12	-1.396	.642	-.117	.470	3.286
	13	-1.396	.642	-.117	.470	3.286
	14	-1.396	.642	-.117	.470	3.286
	15	-1.396	.642	-.117	.470	3.286
	16	-1.396	.642	-.117	.470	3.286
	17	-1.396	.642	-.117	.470	3.286
	18	-1.396	.642	-.117	.470	3.286
	19	-1.396	.642	-.117	.470	3.286
	20	-1.396	.642	-.117	.470	3.286

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 28333.733

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	1968.193	51	.000
	Block	1968.193	51	.000
	Model	1968.193	51	.000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	26365.539[a]	.066	.105

a. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	20.962	8	.007

Contingency Table for Hosmer and Lemeshow Test

		next_correct = 0		next_correct = 1		
		Observed	Expected	Observed	Expected	Total
Step 1	1	1191	1185.678	1784	1789.322	2975
	2	820	871.842	2123	2071.158	2943
	3	811	767.721	2188	2231.279	2999

4	690	672.909	2352	2369.091	3042
5	597	598.274	2724	2722.726	3321
6	491	491.075	2628	2627.925	3119
7	338	357.483	2367	2347.517	2705
8	217	180.869	1609	1645.131	1826
9	289	320.647	3509	3477.353	3798
10	121	118.501	2090	2092.499	2211

Chapter 7 Reference

[1] K. K. Corbett, A.T. and J. Anderson. Intelligent tutoring systems. In In M.G. Helander, T.K. Landauer & P.V. Prabhu, eds., Handbook of human-computer interaction, pages 849-874, 1997.

[2] Ahuja, Neelu Jyothi, and Roohi Sille. "A Critical Review of Development of Intelligent Tutoring Systems: Retrospect, Present and Prospect." International Journal of Computer Science Issues (IJCSI) 10.4 (2013).

[3] G. R. Shute, V. and K. Raghaven. Inference and discovery in an exploratory laboratory. In Learning and Individual Di erences, pages 279-326. San Francisco: Freeman, 1989.

[4] S. M. Beck, J. and E. Haugsjaa. Applications of AIi in education. In Crossroads, 3, pages 11-15, 1996.

[5] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. The Journal of the Learning Sciences, 4(2):167-207, 199

[6] Rose C., Donmez P., Gweon G., Knight A., Junker B., Cohen W., Koedinger K., Heffernan N.T. (2005). Automatic and semi-automatic skill coding with a view towards supporting on-line Assessment. In Looi, McCalla, Bredeweg, & Breuker (Eds.) The 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam. ISO Press. pp. 571-578.

[7] Arroyo, Ivon, et al. "Web-based intelligent multimedia tutoring for high stakes achievement tests." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2004.

[8] William J. Clancey, 1979, Dialogue Management for Rule-Based Tutorials, Proc: International Joint Conference on AI, pp 155-161.

[9] Brown, I. S., Burton, R. R., Dekleer, J., 1982, Pedagogical natural language, and knowledge engineering techniques in SOPHIE I, II, and m, Intelligent Tutoring Systems, pp. 227-282.

[10] Burton, R. R., 1982, Diagnosing bugs in a simple procedural skill, Cognitive Science, 4, pp. 157-183.

[11] Brown, I. S., Burton, R. R., Dekleer, J., 1982, Pedagogical natural language, and knowledge engineering techniques in SOPHIE I, II, and m, Intelligent Tutoring Systems, pp. 227-282.

[12] A. Sarrafzadeh,, 2008, How do you know that I don't understand? A look at the future of intelligent tutoring systems, Computers in Human Behavior, Vol 24, no 4, pp 1342-1363.

[13] Lee, WeeSan, et al. "Newton's Pen: A pen-based tutoring system for statics." Computers & Graphics 32.5 (2008): 511-524.

[14] Ke, Fengfeng. "Computer-game-based tutoring of mathematics." Computers & Education 60.1 (2013): 448-457.

[15] Hill Jr, Randall W., et al. Pedagogically structured game-based training: Development of the ELECT BiLAT simulation. UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY CA INST FOR CREATIVE TECHNOLOGIES, 2006.

[16] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. In User Modeling and User-Adapted Interaction, pages 253-278, 1995.

[17] Beck, Joseph E., and June Sison. "Using knowledge tracing to measure student reading proficiencies." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2004.

[18] Kasurinen, Jussi, and Uolevi Nikula. "Estimating programming knowledge with Bayesian knowledge tracing." ACM SIGCSE Bulletin 41.3 (2009): 313-317.

[19] d Baker, Ryan SJ, Albert T. Corbett, and Vincent Aleven. "More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2008.

[20] Pardos, Zachary A., and Neil T. Heffernan. "Modeling individualization in a bayesian networks implementation of knowledge tracing." User Modeling, Adaptation, and Personalization. Springer Berlin Heidelberg, 2010. 255-266.

[21] Pardos, Zachary A., and Neil T. Heffernan. "KT-IDEM: Introducing item difficulty to the knowledge tracing model." User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, 2011. 243-254.

[22] Qiu, Yumeng, et al. "Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing." EDM. 2011.

[23] Ritter, Steven, et al. "Reducing the Knowledge Tracing Space." International Working Group on Educational Data Mining (2009).

[24] Pardos, Zachary A., and Neil T. Heffernan. "Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm." EDM 2010 (2010): 161-170.

[25] Pavlik Jr, Philip I., Hao Cen, and Kenneth R. Koedinger. "Performance Factors Analysis--A New Alternative to Knowledge Tracing." Online Submission (2009).

[26] Cen, H., Koedinger, K.R., Junker, B., Learning Factors Analysis - A general method for cognitive model evaluation and improvement: Proceedings of the 8th International Conference on Intelligent Tutoring Systems. Springer Berlin / Heidelberg (2006), 164-175.

[27] Gong, Yue, Joseph E. Beck, and Neil T. Heffernan. "Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2010.

[28] Shih, Benjamin, Kenneth R. Koedinger, and Richard Scheines. "Discovery of Student Strategies using Hidden Markov Model Clustering." Proceedings of the 3rd International Conference on Educational Data Mining, 2010.

[29] Feng, M., Heffernan, N. & Koedinger, K.R. Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In Ikeda, Ashley & Chan (Eds.). Proceedings of the Eighth International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 31-40. 2006.

[30] Wang, Yutao, and Neil T. Heffernan. The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. FLAIRS Conference. 2011.

[31] Hawkins, W., Heffernan, N., Wang, Y. & Baker, S,J,d.. Extending the Assistance Model: Analyzing the Use of Assistance over Time. In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM2013). Memphis, TN. pp. 59-66. 2013.

[32] Feng, Mingyu, Neil Heffernan, and Kenneth Koedinger. "Addressing the assessment challenge with an online system that tutors as it assesses." User Modeling and User-Adapted Interaction 19.3 (2009): 243-266.

[33] M. C. Desmarais, P. Montral, and R. S. J. D. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. 2011

[34] K. R. Koedinger, R. S. J. D. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The pslc datashop.In Handbook of Educational Data Mining, Lecture Notes in Computer Science.

[35] Feng, M., Heffernan, N. T., & Koedinger, K. R. Addressing the testing challenge with a Web-based e-assessment system that tutors as it assesses. Proceedings of the Fifteenth International World Wide Web Conference (WWW-06). New York, NY: ACM Press. ISBN:1-59593-332-9. pp. 307-316.2006

[36] G. R. Shute, V. and K. Raghaven. Inference and discovery in an exploratory laboratory.In Learning and Individual Differences, pages 279{326. San Francisco: Freeman, 1989.

[37] R. C. Atkinson. Ingredients for a theory of instruction. In American Psychologis, volume 27, pages 921{931}. 1972