The Sequence of Action Model: Leveraging the Sequence of Attempts and Hints

Linglong Zhu	Yutao Wang	Neil T. Heffernan
Department of Computer Science	Department of Computer Science	Department of Computer Science
Worcester Polytechnic Institute	Worcester Polytechnic Institute	Worcester Polytechnic Institute
100 Institute Road, Worcester, MA	100 Institute Road, Worcester, MA	100 Institute Road, Worcester, MA
lzhu@wpi.edu	yutaowang@wpi.edu	nth@wpi.edu

ABSTRACT
Intelligent Tutoring Systems (ITS) have been proven to be efficient providing student assistance and assessing their performance when they do their homework. Researchers have analyzed how students' knowledge grows and predict their performance from within intelligent tutoring systems. 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 the sequence of hints and 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 the previous question to predict students' performance. A two step modeling methodology is put forward in the work and is a combination of Tabling method and the Logistic Regression. We compared SOA with Knowledge Tracing (KT) and Assistance Model (AM) and combinations of SOA/AM and KT. The experimental results showed that the Sequence of Action model has reliably better predictive accuracy than KT and AM and its performance of prediction is improved after combining with KT.

Keywords
Knowledge Tracing, Educational Data Mining, Student Modeling, Sequence of Action, Assistance Model, Ensemble.

1. INTRODUCTION
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. KT model makes use of Bayesian network to model students' learning process and predicate their performance.

A variety of extensions of KT model are put forward in recent years. Baker, Corbett, and Aleven (2008) build a contextual guess and slip model based on KT that provides more accurate and reliable student modeling than KT. Pardos and Heffernan extends KT four parameters model to support individualization and skill specific parameters and get better prediction of students' performance. Qiu and Qi et al. find that forgetting is a more likely cognitive explanation for the over prediction of KT when considering the time students take to finish their tasks.

Alternative methods to KT model have been developed. For example, in order to generate adaptive instructions for students, Pavlik Jr., Cen, and Koedinger (2009) put forward the Performance Factor Analysis (PFA) model that can make predictions for individual students with individual skills. Gong, Beck, and Heffernan (2010) compared KT with PFA using multiple model fitting procedures and showed that there are no real differences in predictive accuracy between these two models.

However, little attention is paid to the data generated when students interact with computer tutors. Shih, Koedinger, and Scheines (2010) 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 a dataset that 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 results of their Stepwise-HMM-Cluster model shows that persistent attempts lead to better performance than hint-scaffolding strategy. Some 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 (2006) 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 (2011) built an Assistance Model (AM) and generated a performance table based on students' behavior of doing the previous question. Hawkins et al.(2013) extended AM by looking at students' behavior for the two previous questions.

These educational data mining models that utilize the number of assistance students request and the number of attempts they make to predict students' performance have ignored the sequencing of students' interaction with ITS. 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 the first requesting a hint and then making a wrong attempt. Would this information (whether he started with an attempt or a hint) 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.

In our previous work, we showed a Sequence of Action (SOA) model that made use of information about the action sequence of attempts and hints for a student in previous question better predicted the correctness of a current question.. We reported experimental results of an improvement upon the KT model. However, we later found a mistake in that experiment. So this paper serves as a correction of the previous results and as a formal presentation of the SOA model to the community. We present the SOA model and compare it to the KT model and the Assistance model, as well as the combined models to see if knowing sequence of action information does improve upon a standard Knowledge Tracing model, or even upon knowing number of hints and number of attempts alone.

The raw data and experiment result is available online: https://sites.google.com/site/assistmentsdata/projects/zhu2014.

1.1 The Tutoring System and Dataset

The data we used originated from the ASSISTments platform, 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."

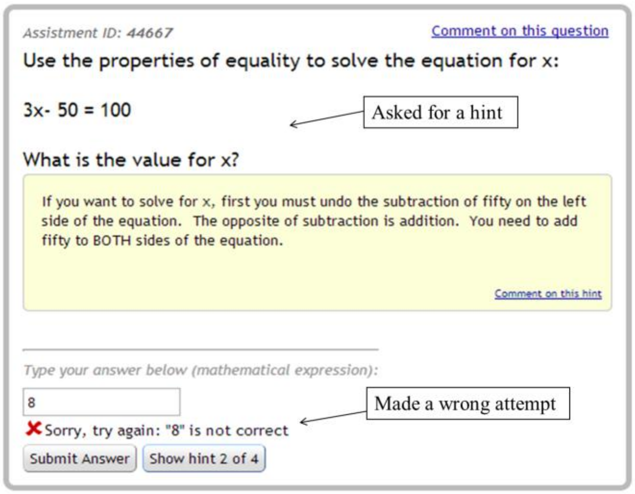

Figure 1. Assistance in ASSISTments. Which is first: asking for a hint or make an attempt?

Figure 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 shows that the student typed in eight and got feedback that this was wrong. Though Figure 1 shows the number of hints and attempts, interestingly you cannot tell whether the student asked a hint first or made an attempt first. This paper's argument is that information is very important.

ASSISTments records all the details about how a student does his or her homework and tests 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 problem, the time interval between an attempt, if he or she asks for a hint, the number of attempts a student makes, the number of hints a student asks for, as well as the answer and result for each attempt a student makes.

Figure 2 shows an example of a detailed sequence of action recorded by the system. The row in blue means that the answer is correct, the row in red means that the answer is wrong, and the row in orange means the student asked for a hint. We can see that this student answered correctly on his first attempt for the first problem PRAQM5U. The sequence of action is 'a' ('a' represents an attempt). For the second problem PRAQM2W, he asked three hints continuously before making the correct answer. The sequence of action is 'hhha' ('h' represents a hint). For the third problem PRAQM2F, he alternatively asked for hints and made attempts, and the sequence of action is 'hahaha'. For the last problem PRAQZPN, he made one wrong attempt before making the correct answer and its action sequencing is 'aa.'

Assignment: 18 - Equivalent Fractions (2) 4.NF.A.1

Time	Action	Object ID / Input text
Wed Feb 29 13:00:49 -0500 2012	Started a problem	PRAQM5U
1 mins 18 secs	Answered	269/17
Wed Feb 29 13:02:18 -0500 2012	Started a problem	PRAQM2W
5 mins 27 secs	Asked for a hint	
3 mins 8 secs	Asked for a hint	
0 mins 26 secs	Asked for a hint	
0 mins 9 secs	Answered	11 6/17
Wed Feb 29 13:11:41 -0500 2012	Started a problem	PRAQM2F
0 mins 5 secs	Asked for a hint	
1 mins 32 secs	Answered	3.6
0 mins 2 secs	Asked for a hint	
0 mins 24 secs	Answered	3/2
0 mins 2 secs	Asked for a hint	
0 mins 7 secs	Answered	3 2/5
Wed Feb 29 13:12:30 -0500 2012	Started a problem	PRAQZPN
0 mins 6 secs	Answered	76000
0 mins 15 secs	Answered	80000

Figure 2. Students' action records in ASSISTments

We used data from one Mastery Learning class. 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. 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.

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 where: 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 make any attempts and no answer was recorded.

Here we only consider the question pairs 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 computed the prediction accuracy on the sixth group. We did this for all six groups.

2. INDIVIDUAL MODELS

2.1 KT

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. The KT models is an Hidden Markov Model (HMM) with a hidden node (student knowledge node) and an observed node (student performance node). It assumes that a skill has four parameters; two knowledge parameters and two performance parameters. 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 parameter 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.

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.

2.2 Assistance Model

Motivated by the intuition that students who need more assistance have lower probability possessing the knowledge, Wang and Heffernan (2011) built a purely data driven "Assistance" model to discover the relationship between assistance information and students' knowledge.

A parameter table was built in which rows represent the number of attempts a student required in the previous question and columns represent the number of hints the student asked for. Each cell contains the probability that the student will answer the current question correctly. The attempts are separated into three bins: one attempt, small number of attempts (2-5 times), and large numbers of attempts (more than five attempts). Hints are separated into four bins: no hint, small number of hints (1, 50%], large number of hints [50%, 100%), and all hints where students for all hints. Table 1 shows the parameter table gained from our dataset. As with Wang and Heffernan's experimental results, the parameter table confirms that students requiring more assistance to solve a problem probably have less corresponding knowledge.

Table 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

2.3 The Sequence of Action Model

The Sequence of Action (SOA) model we present takes advantage of the order information about how students make attempts and ask for hints. Different students have different sequences 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 and we believe they were learning by themselves. In the data, there are 217 different sequences of actions. Intuitively, students' actions reflect their study attitude and this determines their performance. Based on the assumption that students who make more attempts 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 getting 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; and (5) Alternative, Hint First: the students asked for hint and made attempts alternatively and asked for a hint first. Table 2 shows the division and some examples of the action sequences in each category.

Table 2. 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 2, '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 moves on to the next question.

2.3.1 Sequence of Action Tabling

After dividing all of sequence of actions into five categories, we use a Tabling method, which gets the next percent correct directly from the training data. 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 2. Next correct percent table of training group of fold 1

Bin Name	**Total Count**	**Next Correct Count**	**Next Correct Percent**
'a'	22964	19157	0.834
'a+'	3538	2690	0.760
'h+'	335	172	0.513
'a-mix'	2030	1318	0.649
'h-mix'	72	37	0.513

Table 3 shows the table computed for fold 1. Tables for other folds are similar. From Table 3, 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 'a+' bin, 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 'h+' 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.

The alternative sequence of action reflects students' learning process. Intuitively, these students have positive attitudes for study. They want to get some information from the hint based on which they try to solve the next 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 decides 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 makes their performance as bad as the students asking exclusively for hints first.

2.3.2 Logistic Regression

In this section, we are going to introduce the second part of the SOA model that 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 enough to be used to predict students' performance. However, we can use it as a feature in our logistic regression prediction model.

The dependent variable ***Next Correct*** of the logistic regression model has two states: correct and incorrect. The independent variables are ***Skill_ID*** and ***Credit*** (the next correct percentage generated by the tabling method). ***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 before, there are six folds and each fold has their own next correct percentage table.

We used Binary Logistic Regression in SPSS to train the model. Logistic coefficients are fitted through Expectation Maximization of at most 20 steps. Parts of coefficients of the first fold are shown in Table 4.

Table 4. Coefficients of logistic regression model of fold 1

Parameters	**Value**
β_0(Intercept)	-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
…….	……
$\beta_{1,50}$(skill_id 371)	0.470
β_2(Credit)	3.286

3. MODEL COMBINATION

Since the SOA model uses completely different information from KT, we expected a potential improvement from combing SOA results with the predictions from KT. We combined models using two different methods.

The first method was simply average the SOA and KT predictions. Presumably, if a group of models have high accuracies and uncorrelated errors, we can get lower error by averaging them. To compare with the combination of AM model and KT model, we also computed the average of these two models.

The second method was a linear regression model with student performance as the dependent variable. This method takes into account the fact that different models' predictions may have different weight in the final prediction. If one of the models is more useful than the other, this method will allow us to learn which model should be weighted more heavily. SPSS was used to train linear regression models. The function for KT and AM is:

-0.322+0.639*AM_prediction+0.769*KT_prediction;

The function for KT and SOA is:

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

We did not combine AM and SOA, because both of them use information about hints and attempts. From the functions, we can tell that SOA weights heavier than KT, which indicates that SOA is more useful than KT in making a prediction.

4. EXPERIMENTAL RESULTS

4.1 Compare AM, SOA and KT

To evaluate how well each of the individual models (SOA, AM, KT) and the combined 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 and higher values for AUC indicate better model fit.

Table 5. 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**

Table 5 shows values of the three metrics from a six-fold across validation, which are calculated by averaging corresponding numbers obtained from each validation. As with Wang and Heffernan's results (Wang & Heffernan, 2011), the performance of linear regression combination of AM and KT, called as 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 combination of any two models have higher prediction accuracy and this is especially true that for the average ensemble of SOA and KT, called as AVG(SOA, KT), which 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 SOA and KT. However, from the two tailed paired t-test results shown in Table 6, the statistical difference between any two pairs of model combinations are not significant.

To examine whether there is significant difference between these models, we performed a 2-tailed paired t-test. The p values are shown in Table 6. We observe that most of the differences between two models are reliable, except for when we compare some AM and KT combined models with SOA and KT combined models. Both SOA and AM use the information about students' actions of hints and attempts. There might be a chance that SOA and LR(AM, KT) have some prediction overlap.

Table 6. Reliability when compare 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**	**0.037**
SOA vs LG(AM, KT)	0.298	**0.030**	0.083
SOA vs LR(SOA, KT)	**0.000**	**0.001**	**0.006**
SOA vs AVG(SOA, KT)	**0.020**	**0.000**	**0.003**
KT vs LR(AM, KT)	**0.000**	**0.000**	**0.000**
KT vs LR(SOA, KT)	**0.000**	**0.000**	**0.000**
KT vs AVG(SOA, KT)	**0.000**	**0.000**	**0.000**
LR(AM, KT) vs LR(SOA, KT)	0.265	0.296	0.469
LR(AM, KT) vs AVG(SOA, KT)	0.271	0.138	0.079
LR(SOA, KT)vs AVG(SOA, KT)	0.258	**0.001**	**0.010**

4.2 Further Analysis for SOA and KT

From the last section, we observed the best model is the AVG(SOA,KT) model. In order to better investigate this combination, we ran student level and skill level analysis.

Tables 7 and 8 shows the student level result across 66 students to account for the non-independence of their actions. Take MAE as an example, 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. Table 8 shows the t-test p value for each pair of these three models, where the remaining degrees of freedom on all the tests is 65.

Table 7. Student Level accuracy of KT, SOA and Ensemble

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

Table 8. Student level reliability of difference of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT vs SOA	**0.0000**	**0.0000**	0.0551
KT vs AVG	**0.0000**	**0.0000**	**0.0000**
SOA vs AVG	**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.

Table 9 and 10 shows the skill level result across all 51 skills. From Table 9 we observe a very low AUC value for all the models, which indicates these models do not make a good classification at skill level. The t-test p value with remaining degrees of freedom 50 is shown in table 10.

Table 9. Skill level accuracy of KT, SOA and Ensemble

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

Table 10. Skill Level reliability of difference of KT, SOA and Ensemble

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

The student and skill level analysis generate similar conclusions, that SOA and ensemble outperform KT in all of the three metrics. When we compare the ensemble model with SOA alone, the result is not so clear.

5. DISCUSSION AND FUTURE WORK

In this paper, we put forward a Sequence Of Action model that makes use of sequence of students attempts to answer questions and asking for hints. The SOA model consists of two parts. First, the sequence of students' actions are divided into five categories. A tabling method shows that students who only make attempts tend to answer the next question more correctly than students who only ask for hints. This could be caused by students who make more attempts are trying to figure problems out by themselves and it is an efficient way to master knowledge when they are told the steps to answer these questions by asking for hints. Second, we built a logistic regression model with next question correct percentage as dependent variable and skill_id, credits of sequence of action bins as independent variables.

We conducted six-fold cross validation experiments. The experimental result showed that SOA had reliably higher prediction accuracy than the Knowledge Tracing model and Assistance Model. The average combination of the SOA and KT had the highest prediction. In sum, the sequence of students' actions provided important information in predicting students' performance.

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, the experimental data we used is from ASSISTments, does SOA model still makes a big difference if use data from other intelligent tutor systems? How much can the performance of SOA model be improved if 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 questions when we train the model? How does SOA perform after individualization? These are some of the questions that still need to be explored.

6. CONTRIBUTION

Predicting student performance is an important part of the student modeling task in Intelligent Tutoring Systems. A large portion of papers at EDM have focused on this. Many models and techniques have been used to model and investigate students' performance. However, little attention been paid to the temporally sequential actions of student when interacting with the tutoring system. To our knowledge we are the first to use the temporal sequencing of hints and attempts. It turns out that by paying attention to this we can better predict student performance. In this paper, we introduce the Sequence of Action model which makes use of the click-stream data of the sequence of making attempts and asking for hints when students do their homework using an Intelligent Tutoring System. Students' actions can be very different from each other, but we found there are some useful patterns.

We can think of several ways to improve upon this. First, our five bins that we put students into were somewhat arbitrary. There could be more bins or fewer. If we use more bins, we might have very different predictions. The downside is that for some of these bins we might not have enough data points to reliably fit the parameters. One way to make the model better might be to split the "All Hints" bin into one that has "Reached Bottom out Hint" and one that is "All hints excluding those that reached the bottom out." We could also try to pay attention to features like response time between hints or the response time after a hint in making an attempt.

According to our six-fold cross validation experiments and paired two-tailed t-test, both on student level and skill level, our Sequence of Action model had reliably higher prediction accuracy than KT and AM, the later uses the number of hints students ask for and the number of attempts students make. Furthermore, we combined SOA and KT using average and linear regression methods, and the ensemble model's prediction performance was much better than either SOA or KT. We also compared combination of SOA and KT with combination of AM and KT. The experimental result show that SOA contributes more useful information than AM alone, which indicates that the sequential information of action does convey more information about students' learning than the statistics information of actions students make.

7. ACKNOWLEDGMENTS

We acknowledge funding from NSF (#1316736, 1252297, 1109483, 1031398, 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024).

8. REFERENCES

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

Baker, R.S.J.d., Corbett, A.T. & Aleven, V. (2008). More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Wolf, B., Aimeur, E., Nkambou, R., Lajoie, S. (Eds.) *Intelligent Tutoring Systems. LNCS, 5091*, Springer Berlin. pp. 406-415.

Feng, M., Heffernan, N. & Koedinger, K.R. (2006a). 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.

Gong, Y., Beck, J. & Heffernan, N. (2010). Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting. In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer-Verlag, Berlin. pp. 35-44.

Hawkins, W., Heffernan, N., Wang, Y. & Baker, S,J,d.. (2013). 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.

Pardos, Z. & Heffernan, N. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Paul De Bra, Alfred Kobsa, David Chin, (Eds.) *The 18th Proceedings of the International Conference on User Modeling, Adaptation and Personalization*. Springer-Verlag. pp. 255-266.

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

Qiu, Y., Qi, Y., Lu, H., Pardos, Z. & Heffernan, N. (2011). Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining*. pp. 139-148.

Shih, B., Kenneth K., & Richard S. (2010). Unsupervised Discovery of Student Strategies. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*. pp. 201-210.

Wang, Y. & Heffernan, N. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *The 24th International FLAIRS Conference.* Florida.