The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs

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

Abstract

An important aspect of Intelligent Tutoring Systems is providing assistance to students as well as assessing them. The standard state-of-the-art algorithms (Knowledge Tracing and Performance Factor Analysis) for tracking student knowledge, however, only look at the correctness of student first response and ignore the amount of assistance students needed to eventually answer the question correctly. In this paper, we propose the Assistance Model (AM) for predicting student performance using information about the number of hints and attempts a student needed to answer the previous question. We built ensemble models that combine the state-of-the-art algorithms and the Assistance Model together to see if the Assistance Model brings improvements. We used an ASSISTments dataset of 200 students answering a total of 4,142 questions generated from 207 question templates. Our results showed that the Assistance Model did in fact reliably increase predictive accuracy when combined with the state-of-the-art algorithms.

1. Introduction

Understanding student behavior is crucial for Intelligent Tutoring Systems to improve and to provide better tutoring for students. For decades, researchers in ITS have been developing various methods of modeling student behavior using their performance as observations. One example is the Knowledge Tracing model (Corbett and Anderson 1995), which uses a dynamic Bayesian network to model student learning. A second, called Performance Factor Analysis (Pavlik, Cen and Koedinger 2009), has recently been outperforming Knowledge Tracing (Gong, Beck and Heffernan 2010) by using a logistic regression to predict student performance. In these models, however, the amount of assistance a student requires to answer a question correctly is not utilized. Only the student's first attempt is taken into account, and if a hint is requested, the question is marked wrong. But what would the effect on predictive accuracy be if the number of hints and attempts requested was factored into the model? Presumably, students that require more hints or attempts have lower knowledge (even though there are rare instances where this generalization does not hold; see Shih, Koedinger 2008). We use the term "assistance" to refer to the two quantities: the number of hints and the number of attempts required by a student to answer a question. For those not familiar with ITS, most will not let a student progress to the next question until they have answered the current question correctly; thus, all students eventually get each question right. Our intuition is that low knowledge students are perhaps more likely to require additional hints and attempts.

Feng and Heffernan(2010) showed that they could use this sort of information to better predict a state test score (i.e. the Massachusetts Comprehensive Assessment Systems math test). However, they did not give a model that would function "online" as students are working; they limited themselves to only predicting state test scores. Arroyo, Cooper, etc.(2010) showed how to use this information to predict learning gains. Their work suggests that using hints and attempts to model student behavior online could be effective. Furthermore, in our previous work (Wang, Heffernan 2010) we found even more evidence that the number of hints and attempts contain more predictive power than binary performance, and have the potential to enhance current student modeling techniques.

In this paper, we continue to explore the possibility of utilizing assistance information in tutoring systems to

better model student behavior and better predict student performance.

In the rest of this section, we describe the tutoring system and dataset used in our experiments. Section 2 contains a brief introduction to KT and PFA models, and proposes a simple version of an Assistance Model. Section 3 explores two different methods of combining Assistance Models with KT and PFA models. Experimental results are shown in Section 4. In Section 5 and 6 we discuss our conclusions and future directions for our work.

1.1 The Tutoring System and Dataset

The data used in the analysis presented here came from the ASSISTments system, a freely available web-based tutoring system for 4th through 10th grade mathematics. The system gives tutorial assistance if a student makes a wrong attempt or asks for help. Fig.1. shows an example of a hint, which is one type of assistance. A second type of assistance is presented if they click on (or type in) an incorrect answer, at which point the student is given feedback that they answered incorrectly (sometimes, but by no means always, students will get a context-sensitive message we call a "buggy message").

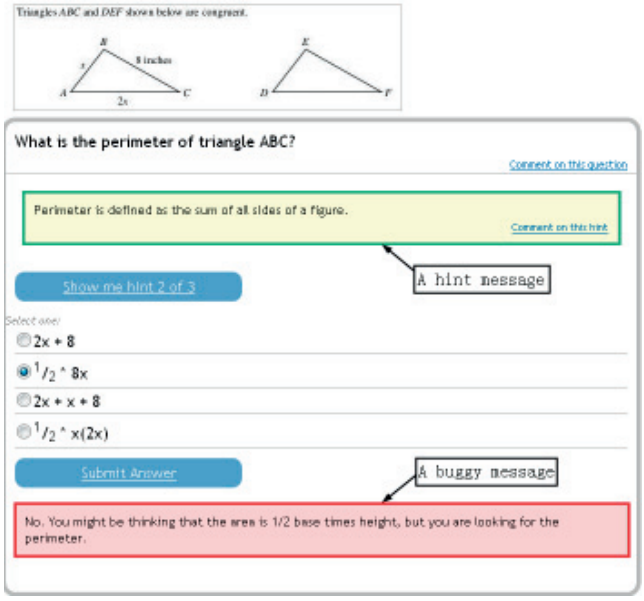

Fig. 1. Assistance in ASSISTment

We used data from four Mastery Learning classes conducted in 2009. Mastery Learning is a strategy that requires students to continually work on a problem set until they have achieved a 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. We assume the difficulty of each question is dependent on its template (even though in theory, some randomly generated numbers might be easier than others). Two hundred 12-14 year old 8th grade students participated in these classes and generated 17,776 problem logs from 93 problem sets. Each problem set was generated from an average of 2.2 templates. The correctness of each answer was logged, as well as the 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.

2. Individual Models

We chose two popular yet very different models: KT and PFA for comparison when exploring the probability of adding assistance information into currently successful student models. We then developed an Assistance Model to infer the probability of a correct response to a given question based on previous assistance information.

2.1 KT

The Knowledge Tracing model shown in Fig.2 has been widely used in ITS and many variants have been developed to improve its performance (Baker et al. 2010, Pardos and Heffernan 2010). It uses 4 parameters for each skill, with two for student knowledge (initial knowledge and probability of learning the skill) and the other two for student performance (the probability of guessing correctly when the student doesn't know the skill and the probability of slipping when the student does know the skill).

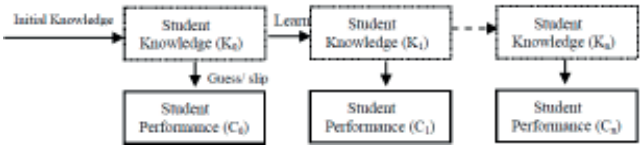

Fig. 2. Knowledge Tracing model
(Figure comes from Gong, Beck et al. 2010)

In our experiment, we used the Bayes Net Toolbox for Matlab developed by Murphy (2001) to implement Knowledge Tracing, and the Expectation Maximization (EM) algorithm to fit the model to the dataset. The EM algorithm finds a set of parameters that maximize the likelihood of the data by iteratively running an expectation step to calculate expected likelihood given student performance data and a maximization step to compute the parameters that maximize that expected likelihood. There have been reported issues of local maxima when using the EM algorithm. Pardos and Heffernan (2001) concluded, based on a simulation study, that with the initial parameters of this algorithm in a reasonable range (the sum of initial

guess and slip value is smaller than 0.5), the algorithm will always converge to a point near the true parameter value. In our experiments, we choose initial parameters for each skill as follows: *initial knowledge* = 0.5, *learning* = 0.1, *guess* = 0.1, *slip* = 0.1.

2.2 PFA

Another model we used in our experiments is Performance Factor Analysis. PFA is a more recently proposed student modeling approach which has been shown to be superior to Knowledge Tracing in several papers (Pavlik et al. 2009, Gong, Beck et al. 2010).

The model uses a logistic regression with student performance as the dependent variable, question identities as factors and the same skill practicing matrix used in Learning Factor Analysis (Cen, Koedinger and Junker 2006) as independent variables. The skill practicing matrix contains the number of prior successes and the number of prior failures for each skill at each point (question) in the problem set. After training, the model gains a parameter for each question representing its difficulty and two parameters for each skill representing the impact of previous successes and previous failures on this question.

Due to the fact that questions are randomly generated as a student progresses through a problem set, we used question template identity as factors rather than question identity in the logistic regression model. Moreover, we did not bound the PFA parameters to prevent negative learning rate.

2.3 Assistance Model

Our previous work on using assistance information (Wang and Heffernan 2010) was based on the intuition that the more assistance a student requires in answering a question, the lower the probability that student possesses the knowledge. In this study we develop a purely data driven model which makes no assumptions about how assistance information reflects student knowledge. The motivation behind building this model is to see directly from the data what the connection is between requiring assistance with a question and the probability of getting the next question right. To do so, we build a parameter table 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. For this value, we simply use the percentage of students who answered the current question correct when the previous question satisfies the row and column.

Statistically, using percent correct as a representation of the probability of correctness requires a large amount of data. In order to ensure each cell in the parameter table contains a sufficient number of data points while still preserving the granularity needed for distinguishing different assistance requirements, we separated the possible number of attempts into 3 interpretable bins:

- One attempt: the student only tried once to get the correct answer.
- Small amount of attempts: the student tried a reasonably small number (set to 2~5 in our experiments) of times.
- Large amount of attempts: the student tried 6 or more times to get the correct answer. It is likely that the student had difficulties in solving the problem, or was gaming the system.

Most responses that fall into the third bin come from fill in questions due to the fact that there are only a small amount of choices in multi-choice questions.

We also separated the possible number of hints into 4 different bins. To normalize the difference in the number of hints contained in each question, we used the percentage of total hints as a measurement rather than the raw number of hints.

- No hint: the student didn't ask for any hint;
- Small amount of hints: the hint percentage the student requested is in the range of (0, 50%];
- Large amount of hints: the hint percentage the student requested is in the range of [50%~100%);
- To the bottom hint: the student asked for all of the hints.

Table 1 shows an example of parameter table we computed from the training data.

	attempt =1	0<attempt <6	attempt >=6
hint_percent=0	0.7376	0.7169	0.6328
0<hint_percent<=.5	0.6454	0.6926	0.6528
.5<hint_percent<1	0.6269	0.6058	0.5409
hint_percent=1	0.3929	0.4835	0.4382

Table 1. Parameter table in AM

From Table 1 we can observe that in general, the more attempts or hints a student requires, the lower the probability that the student can answer the next question correctly. This confirms our intuition about assistance information: students requiring more assistance to solve a problem probably have less corresponding knowledge. Another interesting observation that can be made is that the lowest probability in the table occurs when students required all of the hints and then attempted only once to get the previous question right. This indicates that an alternating behavior of asking for a hint and then attempting to answer could be a more effective pattern of learning.

The Assistance Model has only 12 parameters that inform the relationship between different assistance patterns and the probability of a correct response to the next question. It does not take into account any skill, student or question information and does not model student

learning since it only looks into one previous question to make a prediction. The model utilizes assistance information exclusively. The computational cost of the model is extremely low, which makes it a good complement to other models that do not take into account assistance information.

3. Model Combination

In explaining student behavior and predicting student performance, Knowledge Tracing, Performance Factor Analysis and the Assistance Model will give very different results due to the different pieces of information they use and the different assumptions they make to build the models. By combining the Assistance Model with Knowledge Tracing and Performance Factor Analysis models, we add assistance information into these algorithms, which we believe will give us better prediction results than these models alone. In this section, we explored two different methods of combining models.

3.1 Averaging

Pardos & Heffernan (2010) demonstrated that a simple averaging technique can lead to higher prediction accuracy than either of the two methods that they were comparing by themselves. Similarly, we decided to average the Assistance Model and Knowledge Tracing/Performance Factor Analysis models together. Presumably, if a group of models have high accuracies and uncorrelated errors, we can get lower error by averaging them.

3.2 Regression

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. To address this problem in our experiments, we also constructed a linear regression model with student performance as the dependent variable and prediction results from the Assistance Model and other models as independent variables, in order to find the best weights for the models we intend to combine. If one of the models is more useful than the other, this regression will allow us to learn which model should be weighted more heavily in making a prediction.

4. Results

In order to evaluate the Assistance Model and test the model combination methods, we ran experiments on the dataset described in Section 1.1 with random 80% of students' data in each skill used as training data and random 20 % unseen students' data as test data.

4.1 Individual model results

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 AUC (Area Under ROC Curve). Lower values for MAE and RMSE indicate better model fit while higher values for AUC reflect a better fit. The number of parameters in each model is also reported for comparison purposes.

Table 2 shows the results of the comparison for the three metrics. We used a method that always predicts the mean value as the baseline. In all of the three evaluation metrics, the performance of KT is better than the performance of PFA, which is better than AM. The fact that AM has the lowest predicting accuracy is reasonable considering the small number of total parameters in the model.

	MAE	RMSE	AUC	# of params
Baseline	0.4231	0.4600	0.5	1
AM	0.3894	0.4449	0.6116	12
PFA	0.3797	0.4435	0.7004	393
KT	0.3535	0.4254	0.7329	372

Table 2. Accuracy results of three individual models.

In the Mastery Learning dataset, each problem set contains questions of a single skill, and no multi-skill questions are considered; thus, the number of parameters in PFA = 2*# of problem sets + # of question templates = 2*93+207, while the number of parameters in KT = 4*# of problem sets = 4*93.

We determine whether the difference between two models is statistically significant by computing each evaluation metric's value for each student to account for the non-independence of their actions, then comparing each pair of individual models using a two tailed paired t-test. The results are shown in Table 3. We compute the p value of all three metrics between pairs such as (AM, PFA) to see if the models are reliably different from each other. The bold values in Table 3 show the statistically significant differences between corresponding pairs and metrics. As shown in the table, the differences in RMSE and AUC between AM and PFA, and the difference in AUC between KT and PFA are not significant. Other than those, most of the metric values of the different models in Table 2 are significantly different from each other.

	MAE	RMSE	AUC
(AM, Baseline)	**0.0000**	**0.0000**	**0.0209**
(KT, AM)	**0.0000**	**0.0000**	0.4903
(PFA, AM)	**0.0365**	0.5717	0.2049
(KT, PFA)	**0.0000**	**0.0000**	0.3618

Table 3. Reliability of difference between two individual models.

Our result showing PFA having a lower accuracy than KT is inconsistent with some other works comparing PFA

with KT. The difference may have been caused by a certain property of our experimental dataset. Because PFA uses 1 parameter per question and two parameters per skill while KT uses 4 parameters per skill, it is possible that KT works better in a dataset where skill differences are greater than question variety. Mastery Learning data contains data from different skills, and questions of a particular skill are of similar difficulty levels in order to serve the purpose of random selection.

The accuracy of AM is lower than both that of PFA and KT, yet significantly higher than the baseline. This indicates that although applying AM alone may lead to worse prediction due to a lack of complexity in the model, it still contains valuable information that has the potential to help improve other models' results.

4.2 Combined model results

In order to answer the question of whether or not adding the assistance information into an existing model could lead to a more accurate student performance prediction, we still use MAE, RMSE and AUC as evaluation metrics and continue to report the number of parameters in the final model as measurement of model complexity.

The result is shown in Table 4. For comparison, we also computed the accuracy of combining PFA and KT.

	MAE	RMSE	AUC	# of params
AVG(AM,PFA)	0.3845	0.4304	0.7191	405
LR(AM,PFA)	0.3732	0.4303	0.718	407
AVG(AM,KT)	0.3714	0.4261	0.7358	384
LR(AM,KT)	0.3515	0.4216	0.7369	386
AVG(PFA,KT)	0.3666	0.4246	0.7381	765
LR(PFA,KT)	0.3532	0.4236	0.7396	767

Table 4. Accuracy results of four combined models.

In Table 4, AVG represents the averaging combining method and LR represents the linear regression combining method. The number of parameters used in averaging combining method is equal to the sum of the parameter numbers of each individual method, while the linear regression combining method requires two additional parameters to indicate the impact of each model with respect to the final prediction.

In the linear regression combining method, the regression model is trained on a training dataset. The resulting formula for combining PFA and AM on our dataset is:

-0.3052+0.8241*AM_prediction+0.6003*PFA_predicion;

The formula for combing KT and AM is:

-0.1026+0.2078*AM_prediction+0.9373*KT_prediction;

The formula for combing KT and PFA is:

-0.0600+0.1689*PFA_prediction+0.9145*KT_prediction.

The parameters indicate that in the LR(AM, PFA) model, AM is the main influencer of the final prediction, while in the LR(AM, KT) and LR(PFA, KT) model, KT is the main influencer.

Comparing Table 4 and Table 3, we find that LR(AM, PFA) is better than PFA and LR(AM, KT) and LR(PFA, KT) are better than KT in all metrics. The averaging combining method, on the other hand, does not demonstrate such clear trends of improvement.

As in Section 4.1, we also did reliability analysis by computing metric values for each student to account for the non-independence of actions within each student's dataset, and then compared each pair of models using a two tailed paired t-test. The p values are reported in Table 5, in which bold values indicate the differences are statistically significant.

	MAE	RMSE	AUC
AVG(AM, PFA), PFA	**0.0365**	**0.0000**	0.8788
LR(AM, PFA), PFA	**0.0406**	**0.0000**	**0.0145**
AVG(AM, KT), KT	**0.0000**	**0.0420**	0.8398
LR(AM, KT), KT	**0.0009**	**0.0013**	**0.0095**
AVG(PFA, KT), KT	**0.0000**	**0.0042**	**0.0251**
LR(PFA, KT), KT	0.5536	0.4456	0.7725

Table 5. Reliability of difference between combined models and individual models.

From Table 5 we can see that using the linear regression method to combine the AM model with PFA or KT will reliably increase the accuracy of PFA or KT respectively in all metrics. The fact that the accuracy of using linear regression to combine PFA and KT is not reliably different with KT alone could be caused by the low coefficient PFA has in the regression formula.

5. Discussion and Future Work

The model we proposed in this paper is a simple and fast method of utilizing assistance information. Experiments show this model alone doesn't provide better performance prediction than other more complicated models; however, combining this model with other models will reliably improve the predictive accuracy of that model.

This work is the beginning of an attempting to utilize assistance information in intelligent tutoring systems in order to better predict student performance. There are several questions that we are interested in exploring.

One question is how to improve the Assistance Model by adding more parameters. Currently we use only 12 parameters for all of the data, which assumes that all of the skills and all of the students share the same parameters given a certain assistance pattern. Skill identity parameters and student individualizing parameters can be easily added into the Assistance Model by computing parameter tables

for each skill or each student separately. The resulting model will contain much more information than the Assistance Model currently has, thus theoretically resulting in better predictive accuracy. Although this modification will significantly increase the number of parameters in the model, the cost will remain low due to the fact that the Assistance Model is a table querying method, which requires no complex computation. The Assistance Model could also be extended to take into account the learning curve by building a parameter table using assistance information from a sequence of problems rather than only the previous problem. All of these modifications will face the problem that the data points for computing each table's parameters may become sparse and may not contain enough information. Thus analysis of parameter table reliability is required.

Another question worth exploring is the development of a more interpretable combining method of different models. Intuitively, each model has its own advantages and disadvantages. We would like to know if there are rules which can guide us to choose one model over another given certain circumstances.

6. Contribution

For many years, most assessment work inside Intelligent Tutoring Systems has looked only at student first response and ignored the amount of assistance a student needed to eventually get a problem correct. While we could have lived to have figure out an elegant way to invariable this information into a Bayesian network, in this work, we took a much simpler approach, and simply predicted student correctness on questions by "tabulating" up the number of times students got the next question correct, broken out by the number of hints and attempts the student had to make. This method ignores everything the student did other than the assistance they get on the previous question. Therefore, it makes senses that when coupled (i.e, ensemble) with Knowledge Tracing or Performance Factor Analysis, it does better than either model alone. We encourage the field to look at other ways of modeling the multiple different sources of information. There clearly is a lot of information in the number of hints and attempts.

The method can be easily applied to any current student models. All they have to do is tabulate the percentage change a student get a problem correct, broken out by the number of hints and attempts used on the previous problem.

Acknowledgement

This research was made possible by the U.S. Department of Education, Institute of Education Science (IES) grants #R305K03140 and #R305A070440, the Office of Naval Research grant # N00014-03-1-0221, NSF CAREER award to Neil Heffernan, and the Spencer Foundation. All the opinions, findings, and conclusions expressed in this article are those of the authors, and do not reflect the views of any of the funders.

References

Ivon Arroyo, David G. Cooper, Winslow Burleson, and Beverly P. Woolf. 2010. Bayesian Networks and Linear Regression Models of Students' Goals, Moods, and Emotions. *Handbook of educational data mining*: 323-338. Boca Raton, FL: CRC Press.

Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., MacLaren, B.M., Kauffman, L.R., Mitchell, A.P., Giguere, S. 2010. Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization,* 52-63.

Cen, H., Koedinger, K., Junker, B. 2006. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In Ikeda, M., Ashley, K.D., Chan, T.-W. (des.) ITS 2006. LNCS, vol. 4053, pp. 164-175. Springer, Heidelberg.

Corbett, A., Anderson, J. 1995. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction* 4:253-278.

Feng, M., and Heffernan, N.T. 2010. Can We Get Better Assessment From a Tutoring System Compared to Traditional Paper Testing? Can We Have Our Cake (Better Assessment) and Eat It Too (Student Leaning During the Test)? *In Proceedings of the 3rd International Conference on Educational Data Mining.*

Gong, Y., Beck, J.E. and Heffernan. N.T. 2010. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. *In Proceedings of the 10th International Conference on Intelligent Tutoring Systems*.

Murphy, K.P. 2001, The Bayes Net Toolbox for Matlab, Computing Science and Statistics: *Proceedings of Interface*, 33.

Pardos, Z.A., Heffernan, N.T. 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. *In Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization.* pp. 225-266.

Pardos, Z.A., Heffernan, N.T. 2010. Navigating the parameter space of Bayesian Knowledge Tracing models: Visualization of the convergence of the Expectation Maximization algorithm. In *Proceedings of the 3rd International Conference on EDM.*

Pardos, Z.A., Heffernan, N.T. 2010. Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Proceedings of the 2010 KDD Cup workshop*.

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

Shih, B., Kenneth R. Koedinger, and Richard Scheines. 2008. A Response Time Model For Bottom-Out Hints as Worked Examples. *In Proceedings of the 1st International Conference on Educational Data Mining*. pp. 117-126.

Wang, Y., Heffernan, N.T. and Beck, J.E. 2010. Representing Student Performance with Partial Credit. *Proceedings of the 3rd International Conference on Educational Data Mining.*