Extending the Assistance Model: Analyzing the Use of Assistance over Time

William Hawkins, Neil Heffernan, Yutao Wang
Department of Computer Science
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA
{bhawk90, nth, yutaowang}@wpi.edu

Ryan S.J.d. Baker
Department of Human Development
Teachers College Columbia University
New York, NY
baker2@exchange.tc.columbia.edu

ABSTRACT

In the field of educational data mining, there are competing methods for predicting student performance. One involves building complex models, such as Bayesian networks with Knowledge Tracing (KT), or using logistic regression with Performance Factors Analysis (PFA). However, Wang and Heffernan showed that a raw data approach can be applied successfully to educational data mining with their results from what they called the Assistance Model (AM), which takes the number of attempts and hints required to answer the previous question correctly into account, which KT and PFA ignore. We extend their work by introducing a general framework for using raw data to predict student performance, and explore a new way of making predictions within this framework, called the Assistance Progress Model (APM). APM makes predictions based on the relationship between the assistance used on the two previous problems. KT, AM and APM are evaluated and compared to one another, as are multiple methods of ensembling them together. Finally, we discuss the importance of reporting multiple accuracy measures when evaluating student models.

Keywords

Student Modeling, Knowledge Tracing, Educational Data Mining, Assistance Model, Assistance Progress Model

1. INTRODUCTION

Understanding and modeling student behavior is important for intelligent tutoring systems (ITS) to provide assistance to students and help them learn. For nearly two decades, Knowledge Tracing (KT) [5] and various extensions to it [12, 16, 18] have been used to model student knowledge as a latent using Bayesian networks, as well as to predict student performance. Other models used to predict student performance include Performance Factors Analysis (PFA) [14] and Item Response Theory [8]. However, these models do not take assistance information into account. In most systems, questions in which hints are requested are marked as wrong, and students are usually required to answer a question correctly before moving on to the next one. Therefore, the number of hints and attempts used by a student to answer a question correctly is likely valuable information.

Previous work has shown that using assistance information helps predict scores on the Massachusetts Comprehensive Assessment Systems math test [6], can help predict learning gains [1], and can be more predictive than binary performance [17]. Recently, it has been shown that using simple probabilities derived from the data based on the amount of assistance used, an approach called the Assistance Model (AM), can improve predictions of performance when ensembled with KT [15].

This work continues research in the area of using assistance information to help predict performance in three ways:

1. Specifying a framework for building "tabling methods" from the data, a generalization of AM
2. Experimenting with a new model within this framework called the Assistance Progress Model (APM), which makes predictions based on the relationship between the assistance used on the previous two problems
3. Experimenting with new ways of ensembling these models to achieve better predictions

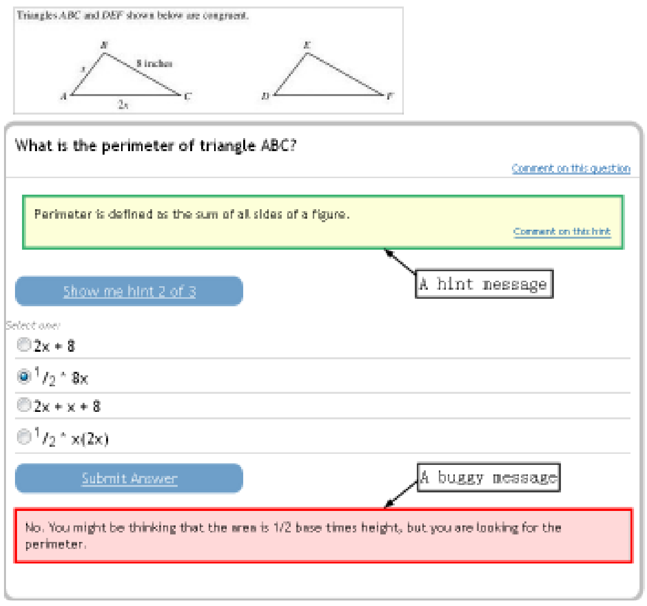
Additionally, the importance of reporting multiple accuracy measures when evaluating student models is discussed, as well as why three of the most commonly reported measures (mean absolute error (MAE), root mean squared error (RMSE) and area under the ROC curve (AUC)) do not always agree on which model makes the most accurate predictions.

Section 2 describes the tutoring system and dataset used. Section 3 describes the methodology: the models and ensembling methods used, the tabling method framework, and the procedure for evaluating the models. Section 4 presents the results, followed by discussion and possible directions for future work in Section 5.

2. DATA

The data used here was the same used in [15], which introduced AM. This dataset comes from ASSISTments, a freely available web-based tutoring system for 4th through 10th grade mathematics.

While working on a problem within ASSISTments, a student can receive assistance in two ways: by requesting a hint, or by entering an incorrect answer, as shown in Figure 1.



Triangles ABC and DEF shown below are congruent.

What is the perimeter of triangle ABC?

Perimeter is defined as the sum of all sides of a figure.

- 2x + 8
- [✓ $^1/_2$ * 8x]
- 2x + x + 8
- $^1/_2$ * x(2x)

No. You might be thinking that the area is 1/2 base times height, but you are looking for the perimeter.

Figure 1. Examples of assistance within ASSISTments (from Wang and Heffernan, 2011)

The dataset comes from four Mastery Learning classes conducted in 2009, where students worked on problem sets until achieving some criterion, usually specified as answering three questions in a row correctly. The questions in these problem sets were generated randomly from templates, with the difficulty of each question assumed to be the same as all other questions generated from the same template. No problem selection algorithm was used to select the next question.

Two hundred 12-14 year old 8th grade students participated in these classes, generating 17,776 problem logs from 93 problem sets. However, due to the nature of the models studied in this paper, data from two of these students could not be used since these two students never answered more than one question within the same problem set.

Since two of the models cannot be used to predict performance on the first question of a problem set, as they rely on assistance usage on previous problems, these models were not trained or evaluated on the first question answered by a student on a given problem set. This reduced the dataset for these models to 12,099 problem logs. KT models were still trained using the entire dataset, but only evaluated on the 12,099 logs they had in common with the other models.

3. METHODS

This section begins by giving an overview of KT, then introduces a framework for building data-driven student models called "tabling methods," and describes two such methods: AM and APM. Next, the approaches used to ensemble these individual models together are briefly discussed. Finally, the procedure and measures used to evaluate all models are discussed.

3.1 Knowledge Tracing

KT is a well-studied student model introduced in [5] that keeps track over time of the probability that a student has mastered a given skill, given their past performance as evidence. The probability that a skill for a given student is in the "known" (vs. the "unknown") state can then be used to predict future performance.

Constructing KT models involves learning four parameters:

1. Initial Knowledge (L_0) – the probability the student has mastered the skill before attempting the first question
2. Learn Rate (T) – the probability the student will have mastered the skill after attempting a given question if they have not mastered the skill already, independent of performance
3. Guess Rate (G) – the probability the student will answer correctly despite not having mastered the skill
4. Slip Rate (S) – the probability the student will answer incorrectly despite having mastered the skill

KT models can be represented as static, "unrolled" Bayesian networks, as shown in Figure 2. The level of knowledge K_m at time step m influences performance on question Q_m. Initial knowledge influences K_0, while knowledge at time step m-1 influences knowledge at time step m for $m > 0$. The learned T, G and S parameters are the same across all practice opportunities, meaning that the conditional probability tables (CPTs) for all nodes K_m where $m > 0$ have the same values, and the CPTs for all Q nodes have the same values.

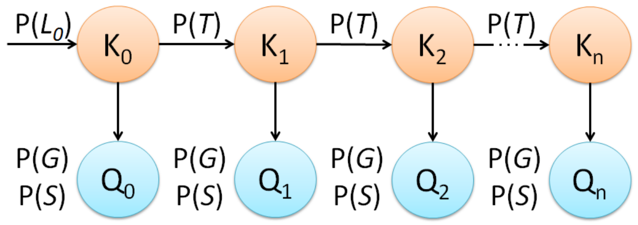


Figure 2. Static Bayesian network representation of Knowledge Tracing

In this work, the Bayes Net Toolbox for Matlab [9] is used to create separate KT models for each problem set. The parameters for each model are learned using Expectation-Maximization, with initial values of 0.3 for L_0, 0.09 for T, 0.1 for G and 0.09 for S.

3.2 Tabling Methods

In previous work [15], a data-driven approach called AM was used to predict performance based on the number of attempts and hints used on the previous problem. This was done by creating a table of probabilities of the student answering the next question correctly on the first attempt without any hints, indexed by the number of attempts and hints used on the previous problem. These probabilities were computed simply by computing the percentage of questions answered correctly on the first attempt with no hints, parameterized by the number of attempts and hints used on the previous problem.

Then, unseen test data was predicted by using the number of attempts and hints used on the previous problem to do a table lookup. The corresponding probability of getting the next question correct in the table was assigned as the prediction.

In this work, we present a generalization of this approach that serves as a framework for data-driven approaches for student modeling. The general procedure is as follows:

1. Create a table based on one or more attributes of the training data.

2. Compute the probability of answering a question correctly for each combination of values of the attributes selected in Step 1, and insert these probabilities into the proper cells in the table.
3. For each previously unseen test case, do a table lookup based on the attributes of the test case to obtain the probability (over the training data) of the student answering the question correctly.
4. Assign the retrieved probability as the prediction for the test case.

The attributes selected in Step 1 can be anything available or computable from the data, such as the number of hints and attempts used on the previous problem as AM does, or the correctness of the previous problem, the time taken, the type of skill, etc. These attributes could also represent which bin an instance falls into, where bins are constructed by splitting up students and/or problems based on some criteria.

Cells may need to be added to the table when values for one or more of the attributes are not available, depending on the nature of the attributes. If there are not enough data points for certain cells, it may help to simply combine them with others. Finally, depending on the nature of the selected attributes and the data, it may be useful to split certain cells based on some criterion.

In this work, two data-driven approaches that follow this framework are explored: the Assistance Model, as described by Heffernan and Wang and further described below, and the Assistance Progress Model (APM), which constructs a table based on the relationships between hints and attempts used on the previous two problems.

3.2.1 *Assistance Model*

As described previously, AM consists of a table of probabilities of a student answering a question correctly based on the number of attempts and the percentage of available hints used on the previous problem of the same skill. Attempts are broken into three bins: 1, (1, 6] and (6, ∞), while the percentage of hints is broken into four: 0, (0, 50], (50, 100) and 100. The AM table constructed from the entire dataset is shown in Table 1.

Table 1. AM table for entire dataset

		Attempts		
Hint %		**1**	**(1, 6]**	**(6, ∞)**
	0	0.778	0.594	0.480
	(0, 50]	0.560	0.623	0.444
	(50, 100)	0.328	0.461	0.444
	100	0.264	0.348	0.374

For instance, according to Table 1, when students answered correctly on the first attempt with no hints, they answered the next question correctly 77.8% of the time. On the other hand, if they required over six attempts and used all of the hints available, they answered the next question correctly only 37.4% of the time.

According to Table 1, when attempts are held constant, the general trend is that as hint usage increases, the probability that the student will answer the next question correctly decreases. This makes sense since hints are more likely to be used by students with lower knowledge of the skill.

When hints are held constant, different patterns occur with respect to the number of attempts used. When no hints are used, the probability of answering the next question correctly decreases as the number of attempts increases. This relationship is reversed when all hints are used. Finally, if just some of the hints are used, making a few attempts (between 2 and 6, inclusive) helps more than making one attempt, but making many attempts (> 6) decreases the probability of answering the next question correctly.

The pattern for no hints can be explained as more attempts required being indicative of lower student knowledge. For all hints being used, more attempts may indicate the student is attempting to learn rather than just requesting hints until the answer is given to them. Using some of the hints suggests the student has not mastered the skill, but has some knowledge of it and is attempting to learn. The relationship between making one attempt and making a few attempts can be explained by the more attempts the student makes, the more they learn, to a point. The use of excessive amounts of attempts probably indicates the student is not learning, despite using some of the hints.

The highest probability in the table, 0.778, corresponds to the case where the previous question was answered correctly. This is unsurprising since in this case, the student likely has mastered the skill. The lowest probability, 0.264, corresponds to making only one attempt while requesting all of the hints. This corresponds to the case where the student requests hints until the answer is given to them. This could be caused by the student simply not understanding the skill, or by the student "gaming the system," or "attempting to succeed in an interactive learning environment by exploiting properties of the system rather than by learning the material" [2]. In either case, not much learning takes place.

In [15], the AM table was constructed using 80% of the data and used to predict the remaining 20%. In this work, all models were evaluated using five-fold cross-validation.

3.2.2 *Assistance Progress Model*

AM only takes into account the number of attempts and percentage of hints required on the previous question to predict the student's performance on the following question, without considering the progress the student is making over time in terms of attempts and hints used. APM, on the other hand, takes into account the relationships between the attempts and percentage of hints used on the previous two problems to predict performance on the next question.

The initial model looked like Table 2, each entry corresponding to a case where the second of the two previous problems requires a lower, equal or higher number of attempts or percentage of hints than the one before it. The number of data points for each cell appears in parentheses.

Table 2. Initial APM table for the entire dataset

		Hint % Relationship		
Attempts Relationship		<	=	>
	<	0.672 (586)	0.611 (1410)	0.567 (60)
	=	0.649 (248)	0.734 (8309)	0.590 (83)
	>	0.541 (85)	0.552 (1019)	0.512 (299)

However, it was necessary to extend the model to handle the case where there were fewer than two previous questions, so a separate cell was added for this situation (it had been treated as (equal attempts, equal hint %)). Next, it was observed that certain cells had few observations, so these cells were combined. Finally, it was realized that the (equal attempts, equal hint %) cell combined data of two very different situations: the case where both questions being compared were answered correctly, and the case where they were both answered incorrectly. Therefore, this cell was split into two cells according to correctness. The final APM table, with probabilities taken over the entire dataset, is shown in Table 3. Cells without enough data on their own have been merged, and the (equal attempts, equal hint %) cell has been split in two: the top cell corresponds to the case when both questions are answered correctly, and the bottom to when both are answered incorrectly. The top-left cell contains the probability that questions with fewer than two predecessors will be answered correctly. The number of data points per cell are in parentheses.

Table 3. APM table for the entire dataset

	Hint % Relationship			
	0.708 (2722)	<	=	>
Attempt Relationship	<	0.672 (586)	0.611 (1410)	0.580 (143)
	=	0.649 (248)	0.791 (5028)	
			0.352 (559)	
	>	0.551 (1104)		0.512 (299)

According to Table 3, when the relationship between attempts is held constant, the general pattern is that the probability of correctness decreases as the relationship between the percentage of hints used worsens. The (equal attempts, equal hint %) cells do not fit this pattern, though this could be because they are split based on correctness. However, the same cell from Table 2 also does not fit the pattern. The same relationship exists between the attempt relationship and probability of answering correctly when the hint % relationship is held constant, again with the exception of the (equal attempts, equal hint %) cells. These patterns are intuitive, as students who are learning the material should require less assistance from one problem to the next and are likelier to answer the next question correctly, whereas those who are not learning will generally require the same amount of assistance or more to proceed, and are less likely to answer the next question correctly without assistance.

The highest probability in the table corresponds to the case where the hints and attempts used are the same for the previous two questions, and both are answered correctly (0.791). The lowest is when they are the same and are both answered incorrectly (0.352). The former result is intuitive since it corresponds to the case where the student answers two questions in a row correctly, the best situation represented in the table. The latter corresponds to no progress in terms of assistance over the previous two questions, indicating that little if any learning has taken place.

3.3 Ensembling Models

As shown in [15], ensembling models can give better results than any individual model on its own. There are two goals in this work regarding ensembled models: improving the predictive power of AM by ensembling it with APM, and improving the predictive power of KT using both AM and APM. Wang and Heffernan already showed that ensembling KT with AM gives better results than KT on its own. It remains to be seen whether including APM will result in further improvements.

In addition to using means and linear regression models, as done in [15], this work also uses decision trees and random forests.

3.4 Evaluation

To evaluate the models, three metrics are computed: MAE, RMSE, and AUC. These metrics are computed by obtaining predictions using five-fold cross-validation (using the same partition for each model), then computing each metric per student. Finally, the individual student metrics are averaged across students to obtain the final overall metrics. Computing the average across students for each metric in this way avoids favoring students with more data than others, and avoids statistical independence issues when it comes to computing AUC. For these reasons, Pardos et al used average AUC per student as their accuracy measure in their work in evaluating several student models and various ways of ensembling them [11].

All three of these metrics are reported because they are concerned with different properties of the set of predictions and therefore do not always agree on which model is best. MAE and RMSE are concerned with how close the real-valued predictions are, on average, to their actual binary values. On the other hand, AUC is concerned with how separable the predictions for positive and negative examples are, or how well the model is at predicting binary classes rather than real-valued estimates.

For example, in Table 4, the first two sets of predictions (P1 and P2) achieve AUCs of 1 since both perfectly separate the two classes (0 and 1). However, P2 achieves much better MAE (0.3960) and RMSE (0.6261) values than P1 (0.5940 and 0.7669, respectively). What's more, P3 achieves an AUC of only 0.5, but outperforms both P1 and P2 in terms of RMSE (0.5292) and P1 in terms of MAE (0.4400).

Table 4. Example dataset

Actual Value	P1	P2	P3
0	0	0.99	0.8
1	0.01	1	0.8
1	0.01	1	0.8
0	0	0.99	0.8
1	0.01	1	0.8

Therefore, it is important to report all of these metrics. As shown above, they do not necessarily agree with each other. Additionally, although MAE and RMSE are similar, not even they always agree on the best model, as RMSE punishes larger errors more than MAE does.

4. RESULTS

In this section, the results for both the individual models and the ensemble models are reported. Given the importance of reporting multiple accuracy measures as discussed in the preceding section, three measures are reported for each model: MAE, RMSE and AUC. Each measure is computed by first computing the measure for each individual student, then averaging across students. The individual student measures are obtained by using predictions made using five-fold cross-validation, where the folds used are identical for every model.

4.1 Individual Models

The results of the three individual models are shown in Table 5. As described before, each of the three metrics are measured for each individual student, and then averaged across students.

In addition to the individual models discussed in Section 3, the results for a baseline model (always predicts 1, the majority class) are reported to serve as a baseline for the other models.

Table 5. Results for the individual models

	MAE	RMSE	AUC
Baseline	0.2510	0.4642	0.5000
AM	0.3657	0.4129	0.5789
APM	0.3844	0.4221	0.5618
KT	0.3358	0.4071	0.6466

Unsurprisingly, KT performs reliably better than AM and APM in MAE ($t(197) = -8.45$, $p < .0001$; $t(197) = -13.55$, $p < .0001$) and AUC ($t(187) = 6.35$, $p < .0001$; $t(187) = 5.97$, $p < .0001$), and at least marginally better in RMSE ($t(197) = -1.75$, $p = .0824$; $t(197) = -4.44$, $p < .0001$), as KT is a full student model, whereas AM and APM do not attempt to model student knowledge and make predictions solely on the basis of table lookups. Additionally, AM outperforms APM in MAE and RMSE ($t(197) = -12.88$, $p < .0001$; $t(197) = -5.61$, $p < .0001$), which is also not surprising considering that APM does not consider the actual number of attempts or percentage of hints used, only the relationships between them for the previous two questions. APM also has fewer parameters (9) than AM (12). The difference in AUC was not reliable ($t(187) = 1.62$, $p = .1063$).

The baseline model reliably outperforms all other models in terms of MAE ($t(197) = -15.30$, $p < .0001$; $t(197) = -18.36$, $p < .0001$; $t(197) = -10.62$, $p < .0001$), and reliably underperforms all other models in terms of RMSE ($t(197) = 5.87$, $p < .0001$; $t(197) = 4.92$, $p < .0001$; $t(197) = 6.01$, $p < .0001$) and AUC ($t(187) = -9.72$, $p < .0001$; $t(187) = -6.34$, $p < .0001$; $t(187) = -12.80$, $p < .0001$). It makes sense that the baseline performs well in terms of MAE, given that the mean value of the target attribute, the correctness of a question, is 0.6910. RMSE punishes larger differences more than MAE, making the baseline pay more for its wrong predictions of all cases where the student got the question wrong. Finally, since all predictions share the same value, the baseline cannot do any better than chance at separating the data. Therefore, it earns an AUC value of 0.5000.

These drastic differences in performance for the baseline alone across measures highlight the need for reporting multiple accuracy measures when evaluating student models.

4.2 Ensembled Models

In this subsection, various ways of ensembling the individual models are evaluated. Since KT was the best performer of the individual models in all three measures by at least marginally reliable margins, the ensembled models here are compared to KT. In the results for each ensemble method, <u>underlined</u> type indicates measures that are reliably worse than those for KT, **boldface** type indicates measures that are reliably better than those for KT, and regular type indicates there is no reliable difference between the measures for KT and the model in question. Statistical significance was determined using two-tailed pairwise t-tests and Benjamini and Hochberg's false discovery rate procedure [4].

4.2.1 Mean

The first ensembling method involved taking the simple mean of the predictions given by the various models. This was done in five ways: 1) with AM and APM to determine if it outperformed AM and APM on their own; combining KT with 2) AM and 3) APM to determine if either AM or APM improved predictions over using KT on its own; 4) with all three models to determine if it outperformed any of the individual models, and 5) taking the mean of AM and APM first, then taking the mean of those results with KT. The intuition for the last method is that KT performs better than AM, and most likely APM as well. Therefore, taking the mean of AM and APM first gives KT more influence in the final result while still incorporating both AM and APM. The results for these models are shown in Table 6.

Table 6. Results for the mean models

	MAE	RMSE	AUC
AM, APM	<u>0.3751</u>	0.4137	<u>0.5917</u>
KT, AM	<u>0.3508</u>	**0.4006**	0.6472
KT, APM	<u>0.3601</u>	0.4033	0.6409
KT, AM, APM	<u>0.3620</u>	0.4032	0.6433
KT, (AM, APM)	<u>0.3554</u>	**0.4010**	0.6469

According to the table above, taking the mean of KT and any combination of AM and APM predictions produces results that do as well as or reliably outperform KT in RMSE and AUC but reliably underperform in MAE. There is no reliable difference between the top two performing models, "KT, AM" and "KT, (AM, APM)" except in MAE, where "KT, AM" performs reliably better ($t(197) = -12.88$, $p < .0001$). Therefore, at least when taking means, adding APM to a model that already includes AM and KT does not reliably improve accuracy in any measure.

Additionally, taking the mean of the AM and APM models yields predictions that are comparable in RMSE and AUC, while reliably worse in MAE ($t(197) = 12.88$, $p < .0001$). Therefore, including APM predictions in mean models does not appear to improve predictive accuracy.

4.2.2 Linear Regression

The second ensembling method is linear regression. In this method, the training data for each fold was used to construct AM, APM and KT models. Predictions were then made for each training instance using these models, and then a linear regression model was built using the three individual predictions as predictors, along with the number of attempts and percentage of hints used, and nominal attributes describing the relationship between the attempts and hints used on the previous two problems. This model was then applied to the fold's test data, whose instances were augmented with predictions from the AM, APM and KT models built from the fold's training data.

Linear regression models were built with six different subsets of the aforementioned features:

1. AM – includes the AM prediction as well as the number of attempts and percentage of hints used on the previous problem
2. AM, KT – the AM set, along with the KT prediction
3. AM, APM* – the AM set, along with the two nominal attributes indicating the relationships between the attempts and hints used for the previous two problems
4. AM, APM*, KT – the AM, APM* set, along with the KT prediction
5. AM, APM – the AM, APM* set along with the APM prediction
6. AM, APM, KT – the AM, APM*, KT set along with the APM prediction

The motivation for testing these subsets of attributes is to determine the relative improvements attained by progressively adding more assistance relationship information to the model, both with and without KT. These models are built in Matlab using the LinearModel class. The results for the linear regression models are shown in Table 7.

Table 7. Results for the linear regression models

	MAE	RMSE	AUC
AM	0.3701	0.4148	0.5770
AM, KT	**0.3338**	**0.4024**	0.6500
AM, APM*	0.3671	0.4127	0.5753
AM, APM*, KT	**0.3319**	**0.4005**	0.6341
AM, APM	0.3647	0.4112	0.5874
AM, APM, KT	**0.3316**	**0.4000**	0.6379

Not surprisingly, models that incorporate KT predictions all outperform their counterparts that lack KT predictions across all three measures. AM and APM together do better than AM, but not when KT is included. The best combination of models for linear regression is AM and KT, as it was for the mean models.

Unlike its corresponding mean model, the linear regression model that combines AM and KT reliably outperforms KT in MAE and RMSE, and is comparable in terms of AUC. This is consistent with the previous finding that combining AM and KT using linear regression outperforms KT [15], though their model did reliably better than KT for all three measures, which were taken over the entire dataset rather than averaged across students.

4.2.3 *Decision Trees*

Next, decision tree models were built from the results of the three individual models in the same way that the linear regression models were built, with the exception that the minimum number of data points per leaf and the level of pruning were optimized using brute force search per fold by using sub-fold cross-validation. The search varied the pruning level from 0 to 100% of the model in steps of 5%, and varied the minimum data points per leaf from 5 to 50 in steps of 5.

The decision trees were given the same set of attributes as the linear regression models, and were tested using the same six subsets of those attributes as described above for the linear regression models. The decision trees were built in Matlab using classregtree, specifying the method as 'regression'.

The same sub-folds were used for each fold for all decision tree models. The results for these models are reported in Table 8. The model names correspond to the same subsets of attributes used for the linear regression models.

Table 8. Results for the decision tree models

	MAE	RMSE	AUC
AM	0.3637	0.4119	0.5793
AM, KT	**0.3293**	**0.4009**	0.6385
AM, APM*	0.3586	0.4087	0.5847
AM, APM*, KT	**0.3286**	**0.4008**	0.6358
AM, APM	0.3586	0.4090	0.5860
AM, APM, KT	**0.3290**	**0.4012**	0.6351

As for the linear regression models, the models that include KT predictions perform better than those that did not, across all three accuracy measures. Adding APM* to AM reliably improves accuracy, but there is no difference between this and combining AM and APM. Adding APM features of any kind do not improve models that include KT predictions. As for the linear regression models, the decision tree that performs the best is the one that only includes KT and AM, which reliably outperforms KT in both MAE and RMSE, with no reliable difference in AUC.

4.2.4 *Random Forest*

The final ensembling method used in this work was Random Forest, which is a collection of decision trees where each individual decision tree was built from a random subset of the attributes and a random subset of the data. In this work, random forests consisted of 1,000 such trees, which were each built randomly from any subset of the attributes and between 10% and 90% of the data. The prediction of the random forest as a whole for a given test instance was the simple mean of the predictions given by each individual tree within the forest. The trees were regression trees and required a minimum of five data points per leaf node. No pruning was done, as varying the pruning levels did not appear to significantly affect the predictive accuracy of the forests for this dataset.

The same set of attributes used for linear regression and decision trees were used in the random forest models, and the same six attribute subsets were tested separately as for the other methods.

With the exception of MAE (many of the predictions were 1, which happens to be the majority class), these models performed worse than the other ensembling methods. This could be due to most of the trees being overfit to the training data, as sub-fold cross-validation with brute force search of optimal pruning parameters was not performed for these trees as it was for the individual decision trees reported on in the previous section.

However, averaging these models with KT produced better results, as shown in Table 9.

Table 9. Results for averaging the KT and random forest models

	MAE	RMSE	AUC
AM	0.3505	**0.4002**	0.6461
AM, KT	**0.3054**	0.4117	0.6313
AM, APM*	0.3479	**0.3985**	0.6477

AM, APM*, KT	**0.3005**	0.4109	0.6358
AM, APM	<u>0.3485</u>	**0.3990**	0.6468
AM, APM, KT	**0.2997**	0.4090	0.6375

Unlike other ensembling methods, when random forest predictions are averaged with those of KT, progressively more APM data improves accuracy, though not always significantly. Otherwise, adding APM predictions appears to worsen results.

4.2.5 Overall

For the first three ensembling methods, those that included only AM and KT performed the best. However, for random forests, it was the average of KT with the random forest consisting of predictions from all three individual models. Table 10 reproduces these results, with **bold-faced** type indicating values that are reliably better than KT, and <u>underlined</u> type indicating values that are reliably worse. Table 11 reports the p-values of the differences between these models for each accuracy measure, with values indicating reliable differences in **bold-faced** type.

Table 10. Results for the best of each ensembling method

	MAE	RMSE	AUC
MEAN	<u>0.3508</u>	**0.4006**	0.6472
LR	**0.3338**	**0.4024**	0.6500
TREE	**0.3293**	**0.4009**	0.6385
RF	**0.2997**	0.4090	0.6375

Table 11. Significance tests for the best ensembling methods

	MAE	RMSE	AUC
MEAN, LR	**0.0000**	0.1659	0.4274
MEAN, TREE	**0.0000**	0.8803	0.1116
MEAN, RF	**0.0000**	**0.0022**	0.1400
LR, TREE	**0.0000**	0.1669	0.0223
LR, RF	**0.0000**	**0.0026**	0.0406
TREE, RF	**0.0000**	**0.0001**	0.8476

From Tables 10 and 11, it appears that either the decision tree or random forest (averaged with KT) models could be considered the best model, depending on which measure is considered the most important. The random forest model is reliably better than the decision tree in terms of MAE, but reliably worse in terms of RMSE.

In general, it appears there is some value in comparing the usage of assistance over the previous two problems, as ensembling APM with AM consistently gives better results than using AM on its own, except when taking means. Despite this, ensemble methods that use only KT and AM perform better than any other model studied in this work, including all of those using APM. One explanation could be that one important thing that APM captures is learning over the previous two questions, which is already modeled in KT. The one exception is when a random forest of all individual models is averaged with KT, which indicates that there is information that APM takes into account that neither AM nor KT considers. Right now, it is not clear which of these ensemble models is best given the disagreement among the metrics. It depends on the relative importance placed on each metric.

5. DISCUSSION AND FUTURE WORK

In this work, we generalized an existing raw data model, AM, into a framework for predicting student performance by tabling raw data. This framework provides an efficient way for adding new sources of information into existing student models. From there, we developed a new model, APM, which makes predictions based on the relationship between the assistance used on the previous two problems. Finally, we evaluated these models and KT, and then explored several ways of ensembling these models together.

We found that although APM is not as predictive as AM, combining the two with various ensembling methods produces models that reliably outperform AM on its own. This shows that prediction accuracy can be strengthened by recognizing the progress a student makes in terms of the assistance they use. However, for the most part, the best models studied in this paper were those that only ensembled KT and AM. Adding APM to such models did not improve accuracy, except in the case of random forests averaged with KT. Despite this, it is still evident that there is value in considering student progress in terms of assistance. Perhaps there are better methods of incorporating that information into predictive models that will yield better results.

We also confirmed that ensembles of AM and KT reliably outperform KT, in line with previous work [15]. Whereas previous work showed this was the case when computing the measures across all problem logs, this work shows it also holds when the measures are computed as averages across students.

We reported three different accuracy measures to fairly compare models against each other, and argued that reporting multiple measures is necessary since they measure different properties of the predictions and therefore do not always agree on which model is best. We also argued that computing these measures per student, then averaging across students is more reliable than treating all problems as equal since the latter approach favors models that are biased towards students with more data.

Although we found that the ensemble methods perform better than KT at predicting performance, such models are difficult to interpret and therefore may be limited in usefulness. Fitting a KT model for a given skill yields four meaningful parameters that describe the nature of that skill, whereas ensemble methods in this work give models of how to computationally combine predictions from KT and AM to maximize predictive accuracy. Since KT models student knowledge, it can be used to guide an ITS session. KT can also be extended to quantify the effects of help [3], gaming [7], and individual items [10], among other factors, on learning and performance. It appears the usefulness of the ensemble methods is limited to prediction of the next question, a task that serves as a good measure of the validity of a student model but does not appear to be useful in guiding ITS interaction.

On the other hand, AM and APM are simple to compute and do not suffer from the identifiability problem that KT does [13]. AM and APM consist of summaries of the raw data rather than inferred parameters. Although not as predictive as KT, AM and APM give interpretable statistics with little chance of overfitting. Additionally, they consider the assistance used, which could indicate the usefulness of a system's help features. Other tabling methods could be used to study the effects of other aspects of ITS, though likely with lower predictive accuracy than KT due to the limited set of values such methods can use as predictions.

Since the ensemble models outperformed KT but appear to be limited to predicting a student's performance on the next question, finding a way to use such predictions within ITS would be a useful contribution. Question selection could be a possible application (i.e. selecting an easier question if the model predicts the student will answer the next question incorrectly).

Another direction for future work could be determining other useful specializations of the framework we presented for building models from raw data. AM and APM focus on assistance, but other attributes could prove useful. Additionally, this work did not investigate grouping students or problems.

Another future direction could be determining why some models, like APM, can reliably improve a model such as AM when ensembled with it, but not improve results when a third model such as KT is involved. It appears that the information that APM uses is important, but may not be used by APM in the best way possible. Examining the use of assistance over the course of more than just the previous two problems may also prove useful.

Finally, experimenting with other methods of ensembling the models described here and other raw data models within this framework is also worth looking into. Previous work experimented with means and linear regression [15], and this work expanded upon those methods by including decision trees and random forests. However, other work in ensembling student models suggests that neural networks may perform better [11].

6. ACKNOWLEDGMENTS

This research was supported by a Graduate Assistance in Areas of National Need (GAANN) fellowship and Neil Heffernan's CAREER grant. We also acknowledge the many additional funders of the ASSISTments Platform found here: http://www.webcitation.org/5ym157Yfr

All of the opinions expressed in this paper are those solely of the authors and not those of our funding organizations.

7. REFERENCES

[1] Arroyo, I., Cooper, D.G., Burleson, W., and Woolf, B.P. Bayesian Networks and Linear Regression Models of Students' Goals, Moods, and Emotions. in *Handbook of educational data mining*, CRC Press, Boca Raton, FL, 2010, 323-338.

[2] Baker, R.S.J.d., Is Gaming the System State-or-Trait? Educational Data Mining Through the Multi-Contextual Application of a Validated Behavioral Model. in *Complete On-Line Proceedings of the Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling*, (Corfu, Greece, 2007), 76-80.

[3] Beck, J.E., Chang, K., Mostow, J., Corbett, A. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. *Intelligent Tutoring Systems*, Springer Berlin Heidelberg, 2008, 383-394.

[4] Benjamini, Y., Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society*, *Series B*, *57*(1), 289–300.

[5] Corbett, A. and Anderson, J. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253-278.

[6] Feng, M., and Heffernan, N.T., 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 Learning During the Test)? in *Proceedings of the 3rd International Conference on Educational Data Mining*, (Pittsburgh, PA, 2010), Springer Berlin Heidelberg, 309-311.

[7] Gong, Y., Beck, J., Heffernan, N., Forbes-Summers, E, The impact of gaming (?) on learning at the fine-grained level. in *Proceedings of the 10th International Conference on Intelligent Tutoring Systems*, (Pittsburgh, PA, 2010), Springer, 194-203.

[8] Johns, J., Mahadevan, S., Woolf, B. Estimating student proficiency using an item response theory model. *Intelligent Tutoring Systems*, Springer Berlin Heidelberg, 2006, 473-480.

[9] Murphy, K. The bayes net toolbox for matlab. *Computing science and statistics*, *33*(2), 1024-1034.

[10] Pardos, Z.A., Dailey, M.D., Heffernan, N.T. Learning what works in ITS from non-traditional randomized controlled trial data. *International Journal of Artificial Intelligence in Education*, *21*(1), 47-63.

[11] Pardos, Z.A., Gowda, S. M., Baker, R.S.J.d. and Heffernan, N. T. The Sum is Greater than the Parts: Ensembling Models of Student Knowledge in Educational Software. *ACM SIGKDD Explorations*, *13*(2), 37-44.

[12] Pardos, Z. A., Heffernan, N. T., Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. in *Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization*, (Big Island, Hawaii, 2010), 255-266.

[13] Pardos, Z. A., Heffernan, N. T., Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. in *Proceedings of the 3rd International Conference on Educational Data Mining*. (Pittsburg, PA, 2010), Springer Berlin Heidelberg, 161-170.

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

[15] Wang Y., Heffernan N. T., The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. in *Proceedings of the 24th International FLAIRS Conference*, (Palm Beach, FL, 2011)

[16] Wang, Y., Heffernan, N.T., The Student Skill Model. in *Proceedings of the 11th International Conference on Intelligent Tutoring Systems*, (Chania, Greece, 2012), 399-404.

[17] Wang, Y., Heffernan, N.T. and Beck, J.E., Representing Student Performance with Partial Credit. in *Proceedings of the 3rd International Conference on Educational Data Mining*, (Pittsburgh, PA, 2010), Springer Berlin Heidelberg, 335-336

[18] Xu, Y., Mostow, J., Comparison of methods to trace multiple subskills: Is LR-DBN best? in *Proceedings of the Fifth International Conference on Educational Data Mining*. (Chania, Greece, 2012), 41-48.