Ensembling Predictions of Student Post-Test Scores for an Intelligent Tutoring System

ZACHARY A. PARDOS, SUJITH M. GOWDA, RYAN S.J.D. BAKER, NEIL T. HEFFERNAN
Worcester Polytechnic Institute

Over the last few decades, there have been a rich variety of approaches towards modeling student knowledge and skill within interactive learning environments. There have recently been several empirical comparisons as to which types of student models are better at predicting future performance, both within and outside of the interactive learning environment. A recent paper (Baker et al., in press) considers whether ensembling can produce better prediction than individual models, when ensembling is performed at the level of predictions of performance within the tutor. However, better performance was not achieved for predicting the post-test. In this paper, we investigate ensembling at the post-test level, to see if this approach can produce better prediction of post-test scores within the context of a Cognitive Tutor for Genetics. We find no improvement for ensembling over the best individual models and we consider possible explanations for this finding, including the limited size of the data set.

Categories and Subject descriptors: I 2.7 **[Artificial Intelligence]**

General Terms: Student modeling, ensemble methods, Bayesian Knowledge-Tracing, Performance Factors Analysis, Cognitive Tutor

Additional Key Words and Phrases:

1. INTRODUCTION

In recent years, there has been a vigorous debate as to which approach for assessing student knowledge and skill within interactive learning environments achieves the most precise assessment of students' latent knowledge and skills. This debate has been particularly vigorous for the relatively simple case of intelligent tutoring systems where each item is related to a single skill, and for which the item-skill mapping has been well-developed (the issue of how to develop these item-skill mappings is of course a key issue in its own right – cf. Barnes, Bitzer, & Vouk, 2005). For this problem, recent comparisons have studied the difference between variants of Bayesian Knowledge Tracing (Baker, Corbett, & Aleven 2008), the differences between variants of Bayesian Knowledge Tracing and Performance Factors Analysis (Pavlik, Cen, & Koedinger; 2009; Gong, Beck, & Heffernan, 2010), and the differences between these algorithms and baseline approaches such as average performance and lookups based on the correctness of the previous three actions (Baker et al., in press). However, these comparisons have often had contradictory results, likely due to differences between the tutoring systems, populations studied, and exact methods used to make comparisons.

Based on the contradictory results of these comparisons, Baker et al. (in press) proposed that it might be more productive to ensemble the available algorithms (cf. Dietterich, 2000) rather than attempting to determine which algorithm is best. Within ensemble methods, multiple models are integrated into a single predictor. Ensemble

Authors' addresses: Zachary A. Pardos, Sujith M. Gowda, Ryan S.J.d. Baker, Neil T. Heffernan, Department of Social Science and Policy Studies, Department of Computer Science, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA USA 01609. E-mail: zpardos@wpi.edu, sujithmg@wpi.edu, rsbaker@wpi.edu, nth@wpi.edu

methods often achieve better performance than single algorithms, since they are able to leverage each algorithm's strengths in different contexts (Dietterich, 2000; Niculescu-Mizil et al., 2009). In their attempt to do this, Baker and colleagues selected nine algorithms predicting student performance and latent knowledge within intelligent tutoring systems, and evaluated the success of simple linear and logistic ensembling methods in data from a Cognitive Tutor for Genetics. This paper conduced ensembling at the level of individual student actions within the tutor (e.g. each model's prediction of latent student knowledge at a given time within the tutor was ensembled into a single prediction of latent student knowledge). The results were mixed; depending on what assumptions were used, ensemble models were either slightly more successful or slightly less successful than the best single model. In addition, multiple individual models were more successful at predicting post-test scores than ensembling conducted in this fashion.

However, it is known that tutor performance often does not perfectly match post-test performance, and that models trained to predict performance within the tutor software often over-predict post-test performance (Corbett & Anderson, 1995). Predicting post-test scores is important, as it provides a test not just of what students can do within the tutor, but also what knowledge they transfer outside of the tutor software. Recent analyses have also suggested that combining assessment of student knowledge with assessments of student slipping/carelessness can lead to more accurate prediction of post-test performance (Baker et al., 2010). Hence, it may be possible to use ensemble methods to better predict post-test performance by ensembling predictions of post-test scores, including predictions of post-test related constructs such as slipping, instead of ensembling predictions of within-tutor performance. To this end, within this paper, we compare the predictive power of ensemble models and single models for predicting post-test performance among students learning from a Cognitive Tutor.

2. STUDENT MODELS USED

2.1 Bayesian Knowledge-Tracing

Corbett & Anderson's (1995) Bayesian Knowledge Tracing model is one of the most popular methods for estimating students' knowledge. It underlies the Cognitive Mastery Learning algorithm used in Cognitive Tutors for Algebra, Geometry, Genetics, and other domains (Koedinger & Corbett, 2006).

The canonical Bayesian Knowledge Tracing (BKT) model assumes a two-state learning model: for each skill/knowledge component the student is either in the learned state or the unlearned state. At each opportunity to apply that skill, regardless of their performance, the student may make the transition from the unlearned to the learned state with *learning* probability $P(T)$. The probability of a student going from the learned state to the unlearned state (i.e. forgetting a skill) is fixed at zero. A student who knows a skill can either give a correct performance, or *slip* and give an incorrect answer with probability $P(S)$. Similarly, a student who does not know the skill may *guess* the correct response with probability $P(G)$. The model has another parameter, $P(L_0)$, which is the probability of a student knowing the skill from the start. After each opportunity to apply the rule, the system updates its estimate of student's knowledge state, $P(L_n)$, using the evidence from the current action's correctness and the probability of learning:

$$P(L_{n-1}|Correct_n) = \frac{P(L_{n-1}) * (1 - P(S))}{P(L_{n-1}) * \left(1 - P(S)\right) + \ (1 - P(L_{n-1})) * (P(G))}$$

$$P(L_{n-1}|Incorrect_n) = \frac{P(L_{n-1}) * P(S)}{P(L_{n-1}) * P(S) + \ (1 - P(L_{n-1})) * (1 - P(G))}$$

$$P(L_n|Action_n) = \ P(L_{n-1}|Action_n) + \left((1 - P(L_{n-1}|Action_n)\right) * P(T))$$

The four parameters of BKT, $(P(L_0), P(T), P(S),$ and $P(G)$, are learned from existing data, historically using curve-fitting (e.g. Corbett & Anderson, 1995), but more recently using expectation maximization (*BKT-EM*) (Chang et al., 2006) or brute force/grid search (*BKT-BF*) (cf. Baker et al., 2010; Pardos & Heffernan, 2010). Within this paper we use BKT-EM and BKT-BF as two different models in this study. Within BKT-BF, for each of the 4 parameters all potential values at a grain-size of 0.01 are tried across all the students (for e.g.: 0.01 0.01 0.01 0.01, 0.01 0.01 0.01 0.02, 0.01 0.01 0.01 0.03…… 0.99 0.99 0.3 0.1). The sum of squared residuals (SSR) is minimized. For *BKT-BF*, the values for Guess and Slip are bounded in order to avoid the "model degeneracy" problems that arise when performance parameter estimates rise above 0.5 (Baker, Corbett, & Aleven, 2008). For *BKT-EM* the parameters were unbounded and initial parameters were set to a $P(G)$ of 0.14, $P(S)$ of 0.09, $P(L_0)$ of 0.50, and $P(T)$ of 0.14, a set of parameters previously found to be the average parameter values across all skills in modeling work conducted within a different tutoring system.

In addition, we include three other variants on BKT. The first variant changes the data set used during fitting. BKT parameters are typically fit to all available students' performance data for a skill. It has been argued that if fitting is conducted using only the most recent student performance data, more accurate future performance prediction can be achieved than when fitting the model with all of the data (Pardos & Heffernan, in press). In this study, we included a BKT model trained only on a maximum of the 15 most recent student responses on the current skill, *BKT-Less Data* (Nooraei B et al., in press).

The second variant, the *BKT-CGS* (Contextual Guess and Slip) model, is an extension of BKT (Baker, Corbett, & Aleven, 2008). In this approach, Guess and Slip probabilities are no longer estimated for each skill; instead, they are computed each time a student attempts to answer a new problem step, based on machine-learned models of guess and slip response properties in context (for instance, longer responses and help requests are less likely to be slips). The same approach as in (Baker, Corbett, & Aleven, 2008) is used to create the model, where 1) a four-parameter BKT model is obtained (in this case *BKT-BF*), 2) the four-parameter model is used to generate labels of the probability of slipping and guessing for each action within the data set, 3) machine learning is used to fit models predicting these labels, 4) the machine-learned models of guess and slip are substituted into Bayesian Knowledge Tracing in lieu of skill-by-skill labels for guess and slip, and finally 5) parameters for *P(T)* and $P(L_0)$ are fit.

Recent research has suggested that the average Contextual Slip values from this model, combined in linear regression with standard BKT, improves prediction of post-test performance compared to BKT alone (Baker et al., 2010). Hence, we include average *Contextual Slip* so far as an additional potential model.

The third BKT variant, the *BKT-PPS* (Prior Per Student) model, breaks from the standard BKT assumption that each student has the same incoming knowledge, $P(L_0)$.This individualization is accomplished by modifying the prior parameter for each student with the addition of a single node and arc to the standard BKT model (Pardos &

Heffernan, 2010). The model can be simplified to only model two different student knowledge priors, a high and a low prior. No pre-test needs to be administered to determine which prior the student belongs to; instead their first response can be used. If a student answers their first question of the skill incorrectly they are assumed to be in the low prior group. If they answer correctly, they assumed to be in the high prior group. The prior of each group can be learned or it can be set *ad-hoc*. The intuition behind the *ad-hoc* high prior, conditioned upon first response, is that it should be roughly 1 minus the probability of guess. Similarly, the low prior should be equivalent to the probability of slip. Using PPS with a low prior value of 0.10 and a high value of 0.85 has been shown to lead to improved accuracy at predicting student performance (Pardos & Heffernan, 2010).

2.2 Tabling

A very simple baseline approach to predicting a student's performance, given his or her past performance data, is to check what percentage of students with that same pattern of performance gave correct answer to the next question. That is the key idea behind the student performance prediction model called *Tabling*.

In the training phase, a table is constructed for each skill: each row in that table represents a possible pattern of student performance in n most recent data points. For $n = 3$ (which is the table size used in this study), we have 8 rows: 000, 001, 010, 011, 100, 101, 110, 111. (0 and 1 represent incorrect and correct responses respectively.) For each of those patterns we calculate the percentage of correct responses immediately following the pattern. For example, if we have 47 students that answered 4 questions in a row correctly (111 1), and 3 students that after answering 3 correct responses, failed on the 4th one, the value calculated for row 111 is going to be 0.94 (47/(47+3)). When predicting a student's performance, this method simply looks up the row corresponding to the 3 preceding performance data, and uses the percent correct value as its prediction.

2.3 Performance Factor Analysis

Performance Factors Analysis (PFA) (Pavlik, Cen, & Koedinger, 2009) is a logistic model, an elaboration of the Rasch item response model, which predicts student performance based on the student's number of prior failures f and successes s for that skill, with skill-specific weightings γ and ρ for failures and successes. PFA also includes an overall difficulty parameter β for each skill or item, depending on the formulation. Within this paper, we fit β at the skill level. The PFA equation is:

$$m(i, j \in KCs, s, f) = \beta_j + \sum(\gamma_j S_{ij} + \rho_j F_{ij})$$

2.4 CFAR

CFAR, which stands for "Correct First Attempt Rate", is an extremely simple algorithm for predicting student knowledge and future performance, utilized by the winners of the educational data KDD Cup in 2010 (Yu et al., 2010). The prediction of student performance on a specific knowledge component (KC) is the student's average correctness on that KC, up until the current point.

3. GENETICS DATA SET

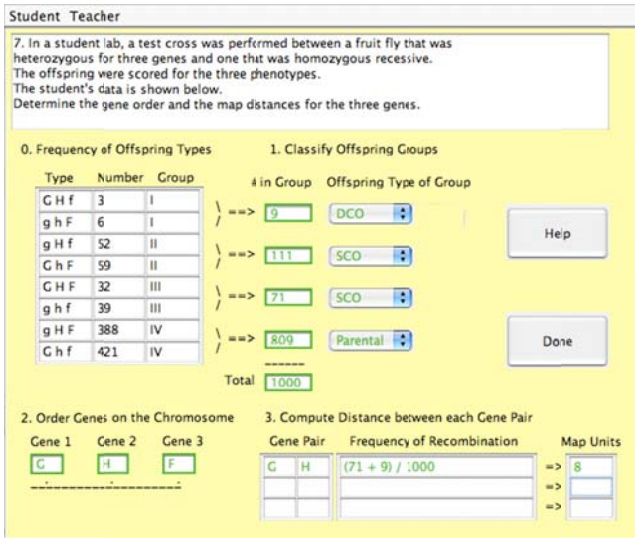

Fig.1 The Three-Factor Cross lesson of the Genetics Cognitive Tutor

The dataset contains the results of in-tutor performance data of 76 students on 9 different skills, with data from a total of 23,706 student actions (entering an answer or requesting help). This data was taken from a Cognitive Tutor for Genetics (Corbett et al., 2010). This tutor consists of 19 modules that support problem solving across a wide range of topics in genetics (Mendelian transmission, pedigree analysis, gene mapping, gene regulation and population genetics). Various subsets of the 19 modules have been piloted at 15 universities in North America.

This data set is drawn from a Cognitive Tutor lesson on three-factor cross, shown in Figure 1. In three factor-cross problems, two organisms are bred together, and then the patterns of phenotypes and genotypes on a chromosome are studied. In particular, the interactions between three genes on the same chromosome are studied. During meiosis, segments of the chromosome can "cross over," going from one paired chromosome to the other, resulting in a different phenotype in the offspring than if the crossover did not occur. Within this tutor lesson, the student identifies, within the interface, the order and distance between the genes on the chromosome by looking at the relative frequency of each pattern of phenotypes in the offspring. The student also categorizes each phenotype in terms of whether it represents the same genotype as the parents (e.g. no crossovers during meiosis), whether it represents a single crossover during meiosis, or whether it represents two crossovers during meiosis.

In this study, 76 undergraduates enrolled in a genetics course at Carnegie Mellon University used the three-factor cross module as a homework assignment. The 76 students completed a total of 23,706 problem solving attempts across 11,582 problem steps in the tutor. On average, each student completed 152 problem steps (SD=50). In the first session, students were split into four groups with a 2x2 design; half of students spent half their time in the first session self-explaining worked examples; half of students spent half their time in a forward modeling activity. Within this paper, we focus solely on behavior logged within the problem-solving activities, and we collapse across the original four conditions.

The post-test, given on paper-and-pencil, consisted of four activities: a straightforward problem-solving post-test, a transfer test, a test of preparation for future learning, and a delayed retention test. Each of the two problems on the problem-solving test (administered at pre-test and post-test) consisted of 11 steps involving 7 of the 9 skills in the Three-Factor Cross tutor lesson, with two skills applied twice in each problem and one skill applied three times. The average performance on the pre-test was 0.33, with a standard deviation of 0.2. The average performance on the post-test was 0.83, with a standard deviation of 0.19. This provides evidence for substantial learning within the tutor, with an average pre-post gain of 0.50.

4. ENSEMBLE METHODS

The premise behind ensembling is to combine the prediction of different models such that the combination results in a more accurate prediction than any single model could produce. We evaluated six methods of combining post-test predictions to investigate the utility of ensembling with our dataset. The six methods evaluated were: Uniform averaging, linear regression, stepwise regression, stepwise model selection with uniform averaging, logistic regression and Random Forests (Brieman, 2001). More detail on the trade-offs between different methods for ensembling can be found in (Dietterich, 2000;Brown, Wyatt, & Tino, 2005). A description of each method is as follows:

Uniform averaging: Uses the mean of the 9 model predictions. While this is the most simple approach, it is also the only approach that is not susceptible to overfitting (since it does not use any form of training).
Linear regression: Fits coefficients to each model's prediction. This approach assumes that there is a linear weighting of models that can lead to better prediction. Predictions lower than zero are changed to zero and predictions higher than one are changed to one.
Stepwise regression: The same as linear regression except that this approach removes models that are not significantly contributing to a decrease in training set error.
Stepwise model selection with uniform averaging: Uses stepwise regression to remove bad models and then averages the prediction of the remaining models. This approach assumes that model selection could be useful but that fitting coefficients might overfit.
Logistic regression: Similar to linear regression except that coefficients are fit based on a logistic curve. The logistic function outputs probabilistic values (between 0 and 1).
Random Forests: Trains many decision trees, each based on a random sampling of models and random resampling of the data, and then averages each decision tree's prediction. This is the only ensemble technique we evaluate which combines model predictions non-linearly. Default MATLAB parameters were used with 20 trees.

Each ensembling method was evaluated with 5-fold cross validation, where four folds were used to train and one to test (note that cross-validating uniform averaging has no impact on goodness of fit, since no training occurs). The same fold assignments were used for this cross-validation as for training and testing the 9 individual models. simplicity we decided not to use ensemble methods which have parameters that need to be tuned such as Neural Networks and Support Vector Machines. This would have added yet another level of cross-validation within the training set. The effectiveness of those methods is a topic for future exploration.

4.1 Using Ensemble Methods in the Real World

It is worth noting that, in evaluating these methods, different forms of cross-validation are required for ensembles versus for the individual models. The individual models are cross-validated at the action level, when the models are trained. They are not fit to the post-test performance, and thus are not cross-validated at this level. However, in ensembling, it is necessary to cross-validate, because fitting occurs at this level.

This form of cross-validation fits to how ensemble methods would typically be used for this problem in the real world. Typically, intelligent tutors use knowledge-tracing (or other assessment) model parameters trained based on data from a previous cohort of students. The tutor software then traces the current year's student responses and makes predictions about their knowledge on the post-test. In predicting post-test scores, an individual model's predictions can be used (and currently, this is the approach commonly used). If a researcher wanted to integrate multiple models in predicting the post-test, uniform averaging could be used post-hoc but with no fitting, in a direct fashion. Neither of these approaches risk over-fitting, as neither use the post-test data to make predictions. However, if the researcher wanted to use ensembling methods as discussed above, one way to do so without risk of over-fitting would be to conduct ensembling using a previous cohort's post-test data, and then to apply the ensembled model to the current cohort's data. The success of the ensembling approach relies on whether the weights learned from last year's data generalize to this year's data. If the ensemble selection is generalizable, it should perform better than uniform averaging and also better than the single model predictions, for the new population. In our analysis, the previous cohort's data is analogous to the four training folds of the 5-fold cross validation whereas the fifth fold, the test fold, represents the current cohort of students.

5. RESULTS

In predicting the post-test using individual model we calculate the n+1 predictions for each skill and student, by applying the rule, where time n equals the last student action in the tutor, and time n+1 equals the student's action after the last action in the tutor (e.g. their action on the post-test):

$$P(correct_{n+1}) = P(L_n) * (1 - P(S)) + (1 - P(L_n)) * P(G)$$

After applying the above rule, we account for the number of times each skill is used in the post-test. Of the eight skills in the tutor lesson, one is not exercised on the test, and is eliminated from the individual models predicting the post-test. Of the remaining seven skills, one is exercised three times, two are exercised twice and four skills are exercised once, in each of the post-test problems. In order to predict the post-test with maximal accuracy, we weight the tutor's prediction of student knowledge of each skill in line with the number of times it appears in the post-test. As each of the individual models can already predict performance on the n+1 step, the opportunity to practice after tutor usage, we do not fit any function to post-test using the individual models; hence the cross-validation gives the same results as not using cross-validation. We use RMSE (root mean squared error) and the Pearson correlation between the model predictions and post-test score, as the model estimates and the post-test scores are both numeric.

As seen in table I, the individual model with the best value of RMSE for predicting the post-test is CFAR, achieving an RMSE of 0.1636 and a correlation of 0.533 to the post-test. However, although BKT-LessData, BKT-EM, and BKT-BF perform slightly worse in terms of RMSE, achieving RMSEs between 0.1754 and 0.1834, each of them

achieves better correlation to the post-test than CFAR. The model with the best correlation to the post-test is BKT-LessData, achieving a correlation of 0.565.

Table I. RMSEs and Correlations between the individual models and the post-test, sorted on RMSE

Model	RMSE	Correlation
AvgCFAR	0.1636	0.5326
Avg-BKT-LessData	0.1754	0.5646
Avg-BKT-EM	0.1781	0.5518
Avg-BKT-BF	0.1834	0.5476
Avg-BKT-PPS	0.1840	0.4988
AvgPFA	0.1895	0.3243
AvgTabling	0.1901	0.2719
Avg-BKT-CGS	0.2812	-0.2367
AverageSlip	0.4279	0.0571

Table II. RMSE and correlations between the ensemble models and the post-test.

Ensemble Model	RMSE	Correlation
Stepwise	0.1652	0.5151
StepwiseWithAveraging	0.1773	0.5477
RandomForest	0.1787	0.3960
UniformAveraging	0.1806	0.5177
LinearRegression	0.2009	0.2671
LogisticRegression	0.2074	0.2645

As discussed above, 5-fold cross-validation is used when evaluating the predictive power of the ensemble models, as data from the post-test is used in creating these models. By cross-validating we can have a greater confidence that the ensemble models generalize to new groups of students.

The cross-validated RMSE and cross-validated correlations (between each of the ensemble models and the post-test) is summarized in table II. The stepwise ensemble model achieves the best RMSE among the ensemble models, achieving an RMSE of 0.1652 and a correlation of 0.515. However, stepwise ensembling achieves a poorer RMSE than the individual model with the highest RMSE, AvgCFAR. The ensemble model with the best correlation was stepwise-with-averaging, which achieved an RMSE of 0.1773 and a correlation of 0.548. However, this was a lower correlation than the top three individual models for this metric, BKT-LessData, BKT-EM, and BKT-BF.

6. CONCLUSIONS

Within this paper, we have considered the effectiveness of ensemble methods to improve prediction of post-test scores for students using a Cognitive Tutor for Genetics. Nine algorithms for predicting latent student knowledge in the post-test were used. Unlike in previous research (e.g. Baker et al., in press), we conducted ensembling at the level of the post-test rather than at the level of performance within the tutor software. It was hypothesized that doing so would lead to superior prediction of the post-test, based on past successes of combined algorithms at predicting the post-test (e.g. Baker et al., 2010), but in fact, the best individual models predict the post-test better than any ensemble

method. Using Root Mean Squared Error (RMSE) as the goodness metric, the CFAR model achieves the best post-test prediction (0.1636), followed closely by stepwise ensembling (0.1652), the best ensemble method. Using correlation as the goodness metric, the BKT-LessData model achieves the best post-test prediction, with stepwise with averaging ensembling (the best ensemble method) tied for the third best correlation with post-test.

Given the past success of ensembling methods in education (Pardos & Heffernan, in press) and other domains (Niculescu-Mizil et al., 2009), this lack of success is highly surprising. There are a few possible reasons for this. First of all, the data set used in this study was relatively small, with only 76 students. Ensembling methods can be expected to be more effective for larger data sets, as more complex models can only achieve optimal performance for large data sets. This is a general problem for analyses of post-test prediction. While large data sets are increasingly available for educational data (Koedinger et al., 2010), it is rare to have post-tests tailored to specific tutor lessons administered to large numbers of students. That said, tutor data sets are increasingly aligned to standardized tests such as the MCAS (Feng, Heffernan & Koedinger, 2009). It is possible to distill post-test-equivalent measures of specific skills from these standardized tests, potentially making it possible to study ensembling of knowledge transferred outside the tutor software at a larger scale, where the benefits of ensembling may be more salient. That said, it is important to be able to improve student models with relatively limited amounts of data (by the time large amounts of data have been collected, many students will have used a version of the tutoring software which is not optimized). One potential option is to optimize a generally successful model (e.g. any of the more successful models in this study) as a first pass on model improvement, and then try ensemble selection on a larger data set, when that data set becomes available.

Another reason why ensemble selection may have been less effective was high inter-correlation between the models' predictions. The predictions made by the nine models were highly correlated (except for the average slip model and BKT-CGS model), with an average inter-correlation of 0.692 (excluding those two models), and the variants on BKT were even more correlated, with an average inter-correlation of 0.927 (excluding BKT-CGS). This high degree of correlation reduces the potential gain from using models in tandem, and may suggest that ensembling may be more effective if methods for post-test prediction which leverage different information are used.

Hence, the final conclusion of this paper is negative, but the possibility remains that alternate takes on ensemble selection may be found to be a valuable part of the repertoire of methods for modeling student knowledge in intelligent tutoring systems.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation via grant "Empirical Research: Emerging Research: Robust and Efficient Learning: Modeling and Remediating Students' Domain Knowledge", award number DRL0910188, and by a "Graduates in K-12 Education" (GK-12) Fellowship, award number DGE0742503. We would like to thank Albert Corbett for providing the data set used in this paper, and for comments and suggestions.

REFERENCES

BAKER, R.S.J.d., CORBETT, A.T., ALEVEN, V. (2008) Improving Contextual Models of Guessing and Slipping with a Truncated Training Set. In Proceedings of 1st International Conference on Educational Data Mining, 67-76.

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.

BAKER, R.S.J.d., PARDOS, Z.A., GOWDA, S.M., NOORAEI, B.B., HEFFERNAN, N.T. (in press) Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems. To appear in Proceedings of the 19th International Conference on User Modeling, Adaptation, and Personalization.

BARNES, T., D. BITZER, VOUK, M. (2005) Experimental analysis of the q-matrix method in knowledge discovery. Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems.

BREIMAN, L. (2001) Random Forests. Machine Learning, 45, 1, 5-32.

BROWN, G., WYATT, J. L., TINO, P. (2005). Managing diversity in regression ensembles. Journal of Machine Learning Research, 6, 1621-1650.

CHANG, K.-m., BECK, J., MOSTOW, J., CORBETT, A. (2006) A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 104-113.

CORBETT, A.T., ANDERSON, J.R. (1995) Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction, 4, 253-278.

CORBETT, A., KAUFFMAN, L., MCLAREN, B., WAGNER, A., JONES, E. (2010) A Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. Journal of Educational Computing Research, 42, 219-239.

DIETTERICH, T. G. (2000). Ensemble Methods in Machine Learning. Proceedings of First International Workshop on Multiple Classifier Systems, 1-15.

FENG, M., HEFFERNAN, N.T., KOEDINGER, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. User Modeling and User-Adapted Interaction: The Journal of Personalization Research, 19, 243-266.

GONG, Y., BECK, J.E., 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, 35-44.

KOEDINGER, K.R., BAKER, R.S.J.d., CUNNINGHAM, K., SKOGSHOLM, A., LEBER, B., STAMPER, J. (2010) A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press, 43-56.

KOEDINGER, K. R., CORBETT, A. T. (2006) Cognitive tutors: Technology bringing learning science to the classroom. In K. Sawyer (Ed.), The Cambridge handbook of the learning sciences (pp. 61-78). New York: Cambridge University Press.

NICULESCU-MIZIL, A., PERLICH, C., SWIRSZCZ, G., SIND-HWANI, V., LIU, Y., MELVILLE, P., et al. (2009) Winning the KDD Cup Orange Challenge with Ensemble Selection. Journal of Machine Learning Research W & CP, 7, 23-34.

NOORAEI B, B., PARDOS, Z.A., HEFFERNAN, N.T., BAKER, R.S.J.d. (in press) Less Is More: Improving the Speed and Prediction Power of Knowledge Tracing by Using Less Data. To appear in Proceedings of the 4th International Conference on Educational Data Mining.

PARDOS, Z. A., HEFFERNAN, N. T. (2010) Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. Proceedings of the 3rd International Conference on Educational Data Mining, 161-170.

PARDOS, Z.A., HEFFERNAN, N. T. (in press) Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. To appear in Journal of Machine Learning Research W & CP.

PAVLIK, P.I., CEN, H., KOEDINGER, K.R. (2009) Learning Factors Transfer Analysis: Using Learning Curve Analysis to Automatically Generate Domain Models. Proceedings of the 2nd International

YU, H-F., LO, H-Y., HSIEH, H-P., LOU, J-K., MCKENZIE, T.G., et al. (2010) Feature Engineering and Classifier Ensemble for KDD Cup 2010. Proceedings of the KDD Cup 2010 Workshop, 1-16.