The Sum is Greater than the Parts: Ensembling Student Knowledge Models in ASSISTments

S. M. GOWDA,
R. S. J. D. BAKER,
Z. PARDOS
AND
N. T. HEFFERNAN
Worcester Polytechnic Institute, USA

Recent research has had inconsistent results as to the utility of ensembling different approaches towards modeling student knowledge and skill within interactive learning environments. While work in the 2010 KDD Cup data set has shown benefits from ensembling, work in the Genetics Tutor has failed to show benefits. We hypothesize that the key factor has been data set size. We explore the potential for ensembling in a data set drawn from a different tutoring system, The ASSISTments Platform, which contains 15 times the number of responses of the Genetics Tutor data set. Within this data set, ensemble approaches were more effective than any single method with the best ensemble approach producing predictions of student performance 10% better than the best individual student knowledge model.

Keywords: student modeling, ensemble methods, Bayesian Knowledge-Tracing, Performance Factors Analysis

1. INTRODUCTION

Over the last decades, there have been a rich variety of approaches towards assessing student knowledge and skill within interactive learning environments, from Overlay Models, to Bayes Nets, to Bayesian Knowledge Tracing [Corbett & Anderson, 1995], to models based on Item-Response Theory such as Performance Factors Analysis (PFA) [cf. Pavlik et al, 2009b]. Multiple variants within each of these paradigms have also been created – for instance, within Bayesian Knowledge Tracing (BKT), BKT models can be fit using curve-fitting [Corbett & Anderson, 1995], expectation maximization (EM) [cf. Chang et al, 2006; Pardos & Heffernan, 2010a], dirichlet priors on EM [Rai et al, 2009], grid search/brute force [cf. Baker et al, 2010; Pardos & Hefferenan, 2010b], and BKT has been extended with contextualization of guess and slip [cf. Baker et al, 2008; Baker et al, 2010] and student priors [Pardos & Heffernan, 2010a; Pardos & Heffernan, 2010b].

Recent work has asked whether assessment of student knowledge and skill can be made more precise by the use of ensemble selection methods [cf. Caruana & Niculescu-Mizil, 2004] that integrate across several existing paradigms for student assessment. In the KDD2010 student modeling competition, teams competed to predict future data on students using Cognitive Tutors [Koedinger & Corbett, 2006], training on earlier data from the same students. Two successful entries in this competition used ensemble selection methods, including the winning entry [Yu et al., 2010], which ensembled across multiple classification algorithms, and the second-best student entry [Pardos & Heffernan, in press], which ensembled across multiple paradigms for student assessment.

Authors' addresses: Sujith M. Gowda, Ryan S.J.d. Baker, Zachary A. Pardos, 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: sujithmg@wpi.edu, rsbaker@wpi.edu, zpardos@wpi.edu, nth@wpi.edu

However, the structure of the training and test sets used in this competition was not representative of common conditions in student modeling, where it is necessary to train models one on cohort of students and apply the models on a new cohort of students (such as the next cohort of students to use the learning system).

In further work to study ensembling in Cognitive Tutor data, ensembles of classic student modeling approaches, when conducted at the student level, failed to achieve substantial improvements on the best individual models in predicting future within-software performance by new students [Baker et al., in press], or future post-test performance by new students [Pardos et al., in press]. However, these studies had two potentially key limitations: First, these studies used only simple methods for ensemble selection, restricting themselves to linear and logistic regression methods. Second, these studies were conducted in a relatively small data set, with only 23,706 student actions across 76 students. Although student models are often trained with data sets of this size, it is possible that ensemble selection methods may need larger data sets to succeed in this domain. In addition, the classic student modeling methods which were ensembled tend to produce similar predictions [Baker et al., in press], suggesting that ensemble selection may have failed due to having relatively little difference between predictors to leverage.

In the study published here, we attempt to discover which of these factors may have led to ensemble selection failing in this case by replicating the previous analysis with two differences: a substantially larger data set, and by using a substantially broader set of ensemble selection algorithms. By doing so, we can better understand the potential of ensemble selection to improve the precision of student models of knowledge and skills.

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 and 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

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

$$P(L_{n-1}|Incorrect_n) = \frac{P(L_{n-1})*P(S)}{P(L_{n-1})*P(S)+ (1-P(L_{n-1}))*(1-P(G))} \qquad (2)$$

$$P(L_n|Action_n) = P(L_{n-1}|Action_n) + \left((1 - P(L_{n-1}|Action_n)\right) * P(T)) \qquad (3)$$

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.

The four parameters of BKT, $(\mathrm{P}(\mathrm{L}_0), \mathrm{P}(\mathrm{T}), \mathrm{P}(\mathrm{S}),$ and $\mathrm{P}(\mathrm{G})$, are learned from existing data, historically using curve-fitting [Corbett], but more recently using expectation maximization (BKT-EM) [Corbett et al. 2010] or brute force/grid search (BKT-BF) [cf. Baker et al. 2010; Pardos & Heffernan, 2010a]. 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 et al. 2008]. For BKT-EM the parameters were unbounded and initial parameters were set to a $\mathrm{P}(\mathrm{G})$ of 0.14, $\mathrm{P}(S)$ of 0.09, $\mathrm{P}(\mathrm{L}_0)$ of 0.50, and $\mathrm{P}(\mathrm{T})$ 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.

The second variant, the BKT-CGS (Contextual Guess and Slip) model, is an extension of BKT [Baker et al. 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 et al. 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 $\mathrm{P}(\mathrm{T})$ and $\mathrm{P}(\mathrm{L}_0)$ 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 [Pardos & Heffernan, 2010a], breaks from the standard BKT assumption that each student has the same incoming knowledge, $\mathrm{P}(\mathrm{L}_0)$. 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. 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 is 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, in press].

2.2 Performance Factors Analysis

Performance Factors Analysis (PFA) [Pavlik et al. 2009a; 2009b] is a logistic regression model, an elaboration of the Rasch model from Item Response Theory. PFA predicts student correctness based on the student's number of prior failures F on that skill (weighted by a parameter ρ fit for each skill) and the student's number of prior successes S on that skill (weighted by a parameter γ fit for each skill). An overall difficulty parameter β is also fit for each skill [Pavlik et al. 2009b] or each item [Pavlik et al. 2009a] – in this paper we use the variant of PFA that fits β for each skill. The PFA equation is:

$$m(i, j \in KCs, s, f) = \beta_j + \sum(\gamma_j S_{ij} + \rho_j F_{ij}) \qquad (4)$$

2.3 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 given skill is the student's average correctness on that skill, up until the current point.

3. ENSEMEBLE METHODS

We use 5 fold cross-validation to evaluate the ensemble methods and we compute A' (also called AUC, the Area under the Receiver-Operating Curve) [Hanley & McNeil, 1982] between the predictions obtained from the models and the correctness of each student action as the goodness metric. The individual student model predictions were ensembled using the following methods.

3.1 Straightforward Averaging

In straightforward averaging, each of the individual models' predictions is averaged for each first attempt at each problem step.

3.2 Regression

We use linear, logistic and stepwise regression methods to ensemble the individual student model predictions. Linear and logistic regression models were developed without feature selection (e.g. ensembles included all the student models). Another variant of regression we use is stepwise regression. In stepwise regression, the best single-parameter model (in terms of RMSE) is chosen, and then the parameter that most improves the model is repeatedly added, until no more parameters can be added which improve the model.

3.3 Ada Boost

Ada Boost is an adaptive boosting algorithm [Freund and Schapire, 1996]. This algorithm uses the same data set over and over, focusing on improving prediction for incorrectly classified data. In this analysis we use two base learners: J48 and Decision Stumps, and use the default settings for the AdaBoost algorithm in RapidMiner [Mierswa et al. 2006]

3.4 Neural Network

In developing the neural net ensemble model, we train the neural nets by varying the size of the hidden layer {10, 25, 50, 100, and 125} and choose default setting for other parameters for Neural Net in RapidMiner version 4.6 [Mierswa et al. 2006]. We use 5-fold cross-validation to evaluate the model. During the training phase of each fold, we split the training data into 2 sub-folds and then train the neural nets with different sizes on one sub-fold and use the test sub-fold to select the best hidden layer size. After selecting the best size, we train the neural net on whole training set and apply the model on the test fold. We follow the same procedure during all the 5 folds.

3.5 Random Forest

Also referred to as bagged random decision trees, Random Forests [Brieman 2001] is an ensemble algorithm that trains many decision trees, each tree using a random resampling of the data (with replacement) and random sampling of the features of the data. In our case the features of the data comprise of predictions from the eight knowledge models. When making a prediction, each tree predicts the probability of a correct response and the average of the votes is taken as the final prediction of the Random Forest. We used 200 trees in the training of our Random Forests with a default feature sampling of 1/3rd and minimum data points per tree leaf of 5. The number of trees was increased from 50 used in previous work [Baker et al. in press] due to the increased size of this dataset from the previous dataset.

4. ASSISTMENT DATASET

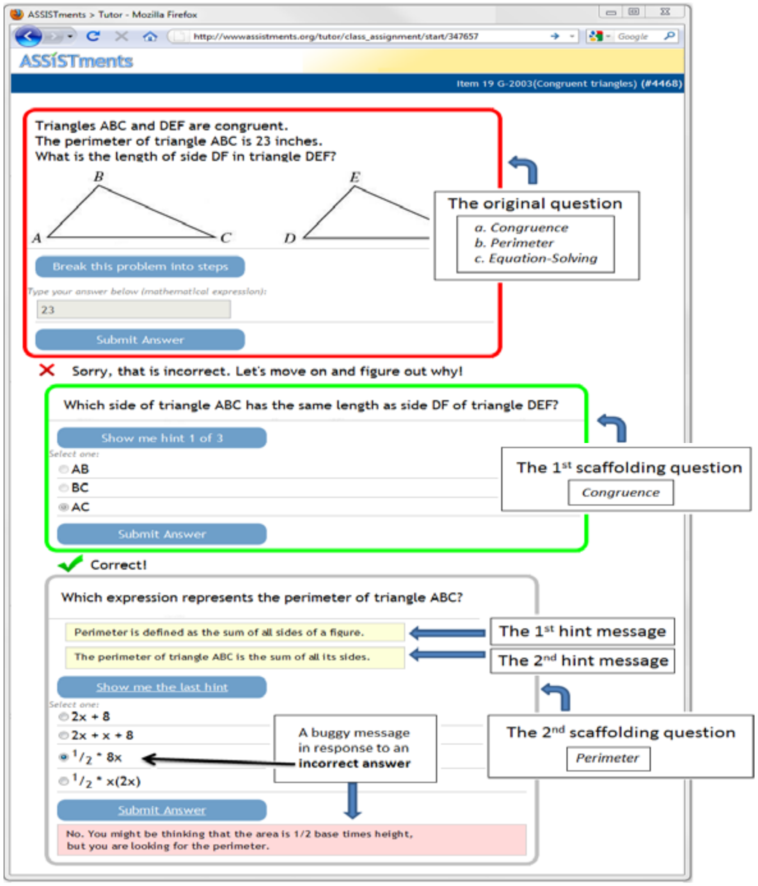
Our dataset comes from student use of the ASSISTments Plaform during the 2005-2006 school year. The students were from 7th and 8th grade Geometry and Algebra classes with ages 12-14. Classes came from different schools and the teachers of the classes would take students to the computer lab to answer questions on ASSISTments about once every two weeks throughout the school year. There were 178,434 total student actions in this dataset produced by 5,422 students. Students received a random selection of math problems from varying skills based on previously released state test items.

The knowledge models used in this paper require problems to be associated with a particular skill. The ASSISTments Platform created a skill model [Razzaq et al. 2007] of 106 skills that provided this association. Some problems were tagged with more than one skill. Since the knowledge models that were used assume a single skill association, these problems were replicated in the dataset for each skill they were associated with such that if a problem was associated with three skills, each student response to that problem would show up three times, once in each of the three skill data files that were created for the models.

5. EVALUATION OF MODELS

5.1 In-tutor Performance of Models, at Student Level

We evaluate both student models and ensemble models using 5-fold cross-validation, at the student level. We balance the students in each of the folds to have equal number of actions and to have equal percent correct. By cross-validating at the student level rather than the action level, we can have greater confidence that the resultant models will



ASSISTments > Tutor - Mozilla Firefox

http://wwwassistments.org/tutor/class_assignment/start/347657

Item 19 G-2003(Congruent triangles) (#4468)

Triangles ABC and DEF are congruent.
The perimeter of triangle ABC is 23 inches.
What is the length of side DF in triangle DEF?

Type your answer below (mathematical expression):

23

Sorry, that is incorrect. Let's move on and figure out why!

Which side of triangle ABC has the same length as side DF of triangle DEF?

Select one:

○ AB
○ BC
● AC

Correct!

Which expression represents the perimeter of triangle ABC?

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

The perimeter of triangle ABC is the sum of all its sides.

Select one:

○ 2x + 8
○ 2x + x + 8
● 1/2 * 8x
○ 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.

Fig 1. An Example of an ASSISTment item.

generalize to new groups of students. We compute the A' (also called AUC, the Area under the ROC) between the predictions obtained from the models and the correctness of each student first attempt on a problem step. We use A' as the goodness metric since it is a suitable metric to be used when predicted variable is binary and the predictions are numerical (predictions of knowledge for each model). To facilitate statistical comparison of A' without violating statistical independence, A' values were calculated for each student separately and then averaged across students (see [Baker, et. al. 2010] for more detail on this statistical method).

The average A' values are summarized in Table I. The best ensemble model was Neural Net (A'=0.7719) and the best individual student model was PFA (A' =0.6994). Neural Net achieved statistically significantly higher performance than PFA, Z=27.21, p<0.001, indicating that the best ensembling methods performed substantially better than the best individual model. Unlike the previous results in the Genetics Tutor [Baker et al, in press] the ensemble models generally appeared to outperform the individual student models, except for Ada Boost with Decision stumps (A'=0.6840) which performed comparably to PFA and the BKT variants, and Averaging (A'=0.6616) which was significantly outperformed by PFA and most of the BKT variants (the smallest difference in terms of statistical significance was between Averaging and BKT-BF, Z=2.18, Table I**.** A' values averaged across students for each of the models

Model	**Average A'**
En-NeuralNet	0.7719
En-RandomForest	0.7662
En-AdaBoost-J48	0.7495
En-Logit	0.7162
En-LinReg	0.7129
En-StepWiseReg	0.7124
PFA	0.6994
En-AdaBoost-DecisionStumps	0.6840
BKT-EM	0.6817
BKT-LessData	0.6816
BKT-BF	0.6649
En-Average	0.6616
BKT-PPS	0.6548
CGS	0.6464
Cslip	0.5103
CFAR	0.5092

p=0.03). The worst single model was CFAR (A'=0.5092), and the second-worst single model was Contextual Slip (A'=0.5103). All the other models achieved statistically significantly higher performance than CFAR and Contextual Slip at p<0.001.

5.2 In-tutor Performance of Models at Action Level

In this evaluation method, the prediction ability of different models is compared based on how well each model predicts each first attempt at each problem step in the data set, instead of averaging within students and then across students. This is a more straightforward approach, although it has multiple limitations: it is less powerful for identifying individual students' learning, less usable in statistical analyses (analyses conducted at this level violate statistical independence assumptions [cf. Baker et al. 2010]), and may bias in favor of predicting students who contribute more data.

Note that we do not re-fit the models in this section; we simply re-analyze the models with a different goodness metric. When we do so, we obtain the results shown in Table II. For this estimation method, the models follow the same pattern as the previous section. The ensemble models again outperform the individual models. The best ensemble model is again Neural Net (A'=0.7693), which is substantially better than best individual model, which is again PFA (A'=0.7053). As in the previous estimation method, the ensemble models again generally perform better than individual models.

Table II**.** A' computed at the action level for each of the models

Model	**A' (calculated for the whole dataset)**
En-NeuralNet	0.7693
En-RandomForest	0.7651
En-AdaBoost-J48	0.7362
En-Logit	0.7183
En-StepWiseReg	0.7182
En-LinReg	0.7182
PFA	0.7053
BKT-LessData	0.7011
BKT-EM	0.7011
BKT-BF	0.6981
En-Average	0.6977
En-AdaBoost-DecisionStumps	0.6804
BKT-PPS	0.6716
Cslip	0.6148
CGS	0.6104
CFAR	0.6067

6. DISCUSSION AND CONCLUSIONS

Within this paper, we have analyzed the effectiveness of a range of approaches for ensembling multiple student knowledge models at the action level, within intelligent tutoring system data. We compare ensembling approaches to the best individual student models of student knowledge. We have compared these models in terms of their power to predict student behavior with in the tutor (cross-validated) and evaluated them at the student and action level. We have found that with this ASSISTments dataset, ensemble methods were unambiguously successful at predicting student performance with greater accuracy than single models. This is contrary to our previous finding [Pardos et al., in press] with a different tutor's dataset, where we found that ensemble methods showed improvement in prediction accuracy compared to single models when evaluated at the action level, but did not show improvement when evaluated at the student level. Also, in that analysis, even at the action level, there was only a 1.4% improvement from the best single model to the best ensemble model. In this current work, we find an 8.3% improvement in prediction accuracy.

Within [Pardos et al., 2011], we hypothesized that there were three possible explanations for the observed lack of improvement using ensemble selection: 1) the knowledge models were too similar to each other; 2) differing number of student responses and different items between the training and test sets led to reduced generalizability; 3) the data set was too small for ensemble selection to be effective in this domain. Since the first two attributes are still true in the ASSISTments data set, we conclude that the size of the data set is the key difference leading to greater success in the current study than in that previous study. It is worth noting, in addition, that the more sophisticated ensemble methods used in the current paper (Neural Networks, Adaboost and Random Forest) also proved substantially more effective than the simple regression-based ensemble selection methods employed in previous work [Pardos et al. in press].

Another contribution of this work is showing the relative predictive performance of the dominant knowledge models in the field on another data set. Within this data set, PFA was the best single model followed by the BKT models and then CGS, CSlip and CFAR. It is worth noting that while BKT-PPS was the best model in the Pardos paper, it is the worst model among the BKT models in this work. PFA was also below all BKT models in [Pardos et al., in press]. In general, the relative performance of different student models has been quite unstable between studies. This finding across studies suggests that there is currently no best model; relative model performance appears to be dependent on the data set. It is not yet clear what features of a specific data set (and the tutor it comes from) are associated with better or worse performance for specific types of student models. This reinforces the motivation behind ensembling models instead of choosing a single model.

Overall this paper demonstrates that ensemble methods can be effective at substantially improving student performance prediction in an ITS, given sufficient amounts of data. It is not yet known exactly how much data is needed in this domain for ensemble selection methods to be effective – for future work, it may be valuable to obtain samples of different sizes from a data set, and test the accuracy of the ensemble for various sample sizes. An additional open research question is whether an ensemble trained on one year's cohort of students will be effective at predicting the next year's cohort. This would be a more rigorous test of ensemble methods in ITS.

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 the additional funders of the ASSISTments Platform found here: http://www.webcitation.org/5ym157Yfr

REFERENCES

BAKER, R.S.J.D., CORBETT, A.T., ALEVEN, V., 2008. More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Proc. of the 9th International Conference on Intelligent Tutoring Systems, 406-415.

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. In: Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization, 52-63.

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

BREIMAN L.,2001. Statistical modeling: the two cultures. Stat Sci 2001;16:199–231

CARUANA, R., NICULESCU-MIZIL, A, 2004. Ensemble selection from libraries of models. In: Proceedings of the 21st International Conference on Machine Learning (ICML'04).

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

CORBETT, A., KAUFFMAN, L., MACLAREN, 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.

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

FREUND, Y. & SCHAPIRE, R. E., 1996. Experiments with a new boosting algorithm. In Proceedings of the 13th International Conference on Machine Learning, pp. 148-146. Morgan Kaufmann.

HANLEY, J. A., & MCNEIL, B. J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143(1), 29-36.

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.

MIERSWA, I., WURST, M., KLINKENBERG, R., SCHOLZ, M., EULER, T. 2006. YALE: Rapid Prototyping for Complex Data Mining Tasks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006), 935-940.

PARDOS, Z. A., GOWDA, S. M., BAKER, R.S.J.D., HEFFERNAN, N. T. (in press) Ensembling Predictions of Student Post-Test Scores for an Intelligent Tutoring System. To appear in Proceedings of the 4th International Conference on Educational Data Mining.

PARDOS, Z. A., HEFFERNAN, N. T., 2010a. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In P. De Bra, A. Kobsa, and D. Chin (Eds.): UMAP 2010, LNCS 6075, 225-266. Springer-Verlag: Berlin

PARDOS, Z. A., HEFFERNAN, N. T., 2010b. 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, 161-170.

PARDOS, Z.A., HEFFERNAN, N. T., 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., 2009a. Learning Factors Transfer Analysis: Using Learning Curve Analysis to Automatically Generate Domain Models. In: Proceedings of the 2nd International Conference on Educational Data Mining, 121-130.

PAVLIK, P.I., CEN, H., KOEDINGER, K.R., 2009b. Performance Factors Analysis – A New Alternative to Knowledge Tracing. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, 531-538. Version of paper used is online at http:// http://eric.ed.gov/PDFS/ED506305.pdf, retrieved 1/26/2011. This version has minor differences from the printed version of this paper.

RAI, D, GONG, Y, BECK, J. E, 2009. Using Dirichlet priors to improve model parameter plausibility. In: Proceedings of the 2nd International Conference on Educational Data Mining, Cordoba, Spain, 141-148.

RAZZAQ, L., HEFFERNAN, N.T., FENG, M., PARDOS, Z.A., 2007. Developing Fine-Grained Transfer Models in the ASSISTment System. Journal of Technology, Instruction, Cognition, and Learning, Vol. 5. Number 3. Old City Publishing, Philadelphia, PA. 2007. pp. 289-304.

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