Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems

Ryan S.J.d. Baker[1], Zachary A. Pardos[2], Sujith M. Gowda[1], Bahador B. Nooraei[2], Neil T. Heffernan[2]

[1] Department of Social Science and Policy Studies, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609 USA
rsbaker@wpi.edu, sujithmg@wpi.edu
[2] Department of Computer Science, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609 USA
zpardos@wpi.edu, bahador@wpi.edu, nth@wpi.edu

Abstract: Over the last 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. However, these comparisons have produced contradictory results. Within this paper, we examine whether ensemble methods, which integrate multiple models, can produce prediction results comparable to or better than the best of nine student modeling frameworks, taken individually. We ensemble model predictions within a Cognitive Tutor for Genetics, at the level of predicting knowledge action-by-action within the tutor. We evaluate the predictions in terms of future performance within the tutor and on a paper post-test. Within this data set, we do not find evidence that ensembles of models are significantly better. Ensembles of models perform comparably to or slightly better than the best individual models, at predicting future performance within the tutor software. However, the ensembles of models perform marginally significantly worse than the best individual models, at predicting post-test performance.

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

1 Introduction

Over the last decades, there have been a rich variety of approaches towards modeling student knowledge and skill within interactive learning environments, from Overlay Models, to Bayes Nets, to Bayesian Knowledge Tracing [6], to models based on Item-Response Theory such as Performance Factors Analysis (PFA) [cf. 13]. 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 [6], expectation maximization (EM) [cf. 4, 9], dirichlet priors on EM [14], grid

search/brute force [cf. 2, 10], and BKT has been extended with contextualization of guess and slip [cf. 1, 2] and student priors [9, 10]. Student models have been compared in several fashions, both within and across paradigms, including both theoretical comparisons [1, 3, 15] and empirical comparisons at predicting future student performance [1, 2, 7, 13], as a proxy for the models' ability to infer latent student knowledge/skills. These empirical comparisons have typically demonstrated that there are significant differences between different modeling approaches, an important finding, as increased model accuracy can improve optimization of how much practice each student receives [6]. However, different comparisons have in many cases produced contradictory findings. For instance, Pavlik and colleagues [13] found that Performance Factors Analysis predicts future student performance within the tutoring software better than Bayesian Knowledge Tracing, whether BKT is fit using expectation maximization or brute force, and that brute force performs comparably to or better than expectation maximization. By contrast, Gong et al. [7] found that BKT fit with expectation maximization performed equally to PFA and better than BKT fit with brute force. In other comparisons, Baker, Corbett, & Aleven [1] found that BKT fit with expectation maximization performed worse than BKT fit with curve-fitting, which in turn performed worse than BKT fit with brute force [2]. These comparisons have often differed in multiple fashions, including the data set used, and the type (or presence) of cross-validation, possibly explaining these differences in results. However, thus far it has been unclear which modeling approach is "best" at predicting future student performance.

Within this paper, we ask whether the paradigm of asking which modeling approach is "best" is a fruitful approach at all. An alternative is to use all of the paradigms at the same time, rather than trying to isolate a single best approach. One popular approach for doing so is ensemble selection [16], where multiple models are selected in a stepwise fashion and integrated into a single predictor using weighted averaging or voting. Up until the recent KDD2010 student modeling competition [11, 18], ensemble methods had not used in student modeling for intelligent tutoring systems. In this paper, we take a set of potential student knowledge/performance models and ensemble them, including approaches well-known within the student modeling community [e.g. 7, 16] and approaches tried during the recent KDD2010 student modeling competition [cf. 11, 18]. Rather than selecting from a very large set of potential models [e.g. 16], a popular approach to ensemble selection, we ensemble existing models of student knowledge, in order to specifically investigate whether combining several current approaches to student knowledge modeling is better than using the best of the current approaches, by itself. We examine the predictive power of ensemble models and original models, under cross-validation.

2 Student Models Used

2.1 Bayesian Knowledge-Tracing

Corbett & Anderson's [6] 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 [8].

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. The equations are as follows:

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

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

$$P(L_n|Action_n) = P(L_{n-1}|Action_n) + \big((1 - P(L_{n-1}|Action_n)\big) * P(T)) \quad (3)$$

The four parameters of BKT, $(P(L_0), P(T), P(S),$ and $P(G)$, are learned from existing data, historically using curve-fitting [6], but more recently using expectation maximization (*BKT-EM*) [5] or brute force/grid search (*BKT-BF*) [cf. 2, 10]. 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 [1]. 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 [11]. 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 [1]. 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 [1] 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 [2]. Hence, we include average *Contextual Slip* so far as an additional potential model.

The third BKT variant, the *BKT-PPS* (Prior Per Student) model [9], 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. 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 [11].

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* [17].

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 representing 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 percentage value as its prediction.

2.3 Performance Factors Analysis

Performance Factors Analysis (PFA) [12, 13] 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 [13] or each item [12] – in this paper we use the variant of PFA that fits β for each skill. The PFA equation is:

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

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 [18]. The prediction of student performance on a given skill is the student's average correctness on that skill, up until the current point.

3 Genetics Dataset

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 [5]. 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

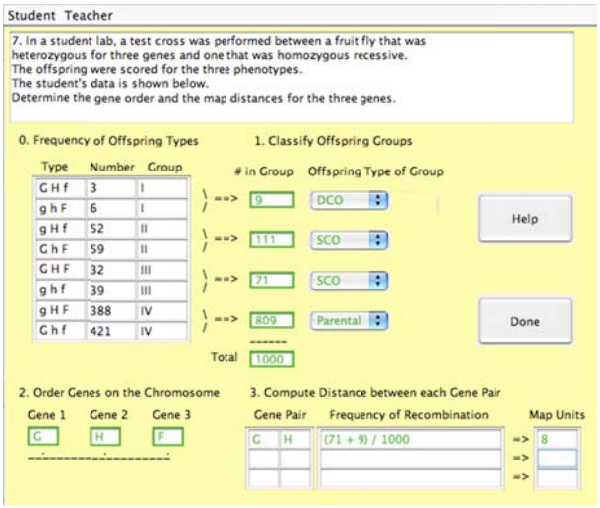

Fig. 1. The Three-Factor Cross lesson of the Genetics Cognitive Tutor

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 an assignment conducted in two lab sessions lasting an hour apiece. 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 problem-solving pre-test and post-test consisted of two problems (counterbalanced across tests), each consisting 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 Evaluation of Models

4.1 In-tutor Performance of Models, at Student Level

To evaluate each of the student models mentioned in section 2, we conducted 5-fold cross-validation, at the student level. By cross-validating at the student level rather than the action level, we can have greater confidence that the resultant models will generalize to new groups of students. The variable fit to and predicted was whether each student first attempt on a problem step was Correct or Not Correct. We used 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 [2] for more detail on this statistical method).

The performance of each model is given in Table 1. As can be seen, the best single model was BKT-PPS (A'=0.7029), with the second-best single model BKT-BF (A'=0.6969) and the third-best single model BKT-EM (A'=0.6957). None of these three BKT models was significantly different than each other (the difference closest to significance was between BKT-PPS and BKT-BF, Z=0.11, p=0.91). Interestingly, in light of previous results [e.g. 16], each of these three models was significantly better than PFA (A'= 0.6629) (the least significant difference was between BKT-PPS and PFA, Z=3.21, p=0.01). The worst single model was BKT-CGS (A'=0.4857), and the second-worst single model was CFAR (A'=0.5705).

Table 1. A' values averaged across students for each of the models

Model	Average A'
BKT-PPS	0.7029
Ensemble: linear regression without feature selection (BKT-PPS, BKT-EM, Contextual Slip)	0.7028
Ensemble: linear regression without feature selection (BKT-PPS, BKT-EM)	0.6973
BKT-BF	0.6969
BKT-EM	0.6957
Ensemble: linear regression without feature selection	0.6945
Ensemble: stepwise linear regression	0.6943
Ensemble: logistic regression without feature selection	0.6854
BKT-LessData (maximum 15 data points per student, per skill)	0.6839
PFA	0.6629
Tabling	0.6476
Contextual Slip	0.6149
CFAR	0.5705
BKT-CGS	0.4857

These models' predictions were ensembled using three algorithms: linear regression without feature selection (e.g. including all models), stepwise linear regression (e.g. starting with an empty model, and repeatedly adding the model that most improves fit, until no model significantly improves fit), and logistic regression without feature selection (e.g. including all models). When using stepwise regression, we discovered that for each fold, the first three models added to the ensemble were BKT-PPS, BKT-EM, and Contextual Slip. In order to test these features alone, we turned off feature selection and tried linear regression ensembling using only these three features, and linear regression ensembling using only BKT-PPS and BKT-EM (the first two models added). Interestingly, these restricted ensembles appeared to result in better A' than the full-model ensembles, although the difference was not statistically significant (comparing the 3-model linear regression vs. the full linear regression without feature selection – the best of the full-model ensembles – gives $Z=0.87$, $p=0.39$).

The ensembling models appeared to perform worse than BKT-PPS, the best single model. However, the difference between BKT-PPS and the worst ensembling model, logistic regression, was not statistically significant, $Z=0.90$, $p=0.37$.

In conclusion, contrary to the original hypothesis, ensembling of multiple student models using regression does not appear to improve ability to predict student performance, when considered at the level of predicting student correctness in the tutor, cross-validated at the student level.

4.2 In-tutor Performance of Models at Action Level

In the KDD Cup, a well-known Data Mining and Knowledge Discovery competition, 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. 2]), 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 2.

For this estimation method, ensembling appears to generally perform better than single models, although the difference between the best ensembling method and best single model is quite small ($A'=0.7451$ versus $A'=0.7348$). (Note that statistical results are not given, because conducting known statistical tests for A' at this level violates independence assumptions [cf. 2]). This finding suggests that how data is organized can make a difference in findings on goodness. However, once again, ensembling does not appear to make a substantial difference in predictive power.

4.3 Models predicting Post-test

Another possible level where ensembling may be beneficial is at predicting the post-test; for example, if individual models over-fit to specific details of in-tutor behavior,

Table 2. A' computed at the action level for each of the models

Model	A' (calculated for the whole dataset)
Ensemble: linear regression without feature selection (BKT-PPS, BKT-EM, Contextual Slip)	0.7451
Ensemble: linear regression without feature selection	0.7428
Ensemble: stepwise linear regression	0.7423
Ensemble: logistic regression without feature selection	0.7359
Ensemble: linear regression without feature selection (BKT-PPS, BKT-EM)	0.7348
BKT-EM	0.7348
BKT-BF	0.7330
BKT-PPS	0.7310
PFA	0.7277
BKT-LessData (maximum 15 data points per student, per skill)	0.7220
CFAR	0.6723
Tabling	0.6712
Contextual Slip	0.6396
BKT-CGS	0.4917

Table 3. Correlations between model predictions and post-test

Model	Correlation to post-test
BKT-LessData (maximum 15 data points per student, per skill)	0.565
BKT-EM	0.552
BKT-BF	0.548
Ensemble: linear regression without feature selection (BKT-PPS, BKT-EM)	0.540
CFAR	0.533
BKT-PPS	0.499
Ensemble: logistic regression without feature selection	0.480
Ensemble: linear regression without feature selection (BKT-PPS, BKT-EM, Contextual Slip)	0.438
Ensemble: linear regression without feature selection	0.342
PFA	0.324
Tabling	0.272
Ensemble: stepwise linear regression	0.254
Contextual Slip	0.057
BKT-CGS	-0.237

a multiple-model ensemble may avoid this over-fit. In predicting the post-test, we account for the number of times each skill will be utilized on the test (assuming perfect performance). Of the eight skills in the tutor lesson, one is not exercised on the test, and is eliminated from post-test prediction. Of the remaining seven skills, four are exercised once, two are exercised twice and one is exercised three times, in each of the two posttest problems. These first two skills are each counted twice and the latter skill three times in our attempts to predict the post-test. We utilize this approach in all attempts to predict the post-test in this paper. We use Pearson's correlation as the goodness metric since the model estimates and the post-test scores are both numerical. Correlation between each model and the post-test is given in Table 3.

From the table we can see that BKT-LessData does better than all other individual models and ensemble models and achieves a correlation of 0.565 to the post-test. BKT-EM and BKT-BF perform only slightly worse than BKT-LessData, respectively achieving correlations of 0.552 and 0.548. Next, the ensemble involving just BKT-PPS and BKT-EM achieves a correlation of 0.54. The difference between BKT-LessData (the best individual model) and the best ensemble was marginally statistically significant, $t(69)=1.87$, $p=0.07$, for a two-tailed test of the significance of the difference between correlations for the same sample. At the bottom of the pack are BKT-CGS and Contextual Slip.

5 Discussion and Conclusions

Within this paper, we have compared several different models for tracking student knowledge within intelligent tutoring systems, as well as some simple approaches for ensembling multiple student models at the action level. We have compared these models in terms of their power to predict student behavior in the tutor (cross-validated) and on a paper post-test. Contrary to our original hypothesis, ensembling at the action level did not result in unambiguously better predictive power across analyses than the best of the models taken individually. Ensembling appeared slightly better for flat (e.g. ignoring student) assessment of within-tutor behavior, but was equivalent to a variant of Bayesian Knowledge Tracing (BKT-PPS) for student-level cross-validation of within-tutor behavior, and marginally or non-significantly worse than other variants of Bayesian Knowledge Tracing for predicting the post-test.

One possible explanation for the lack of a positive finding for ensembling is that the models may have been (overall) too similar for ensembling to function well. Another possible explanation is that the differing number of problem steps per student may have caused the current ensembling method to over-fit to students contributing larger amounts of data. Thirdly, it may be that the overall data set was too small for ensembling to perform effectively, suggesting that attempts to replicate these results should be conducted on larger data sets, in order to test this possibility.

A second interesting finding was the overall strong performance of Bayesian Knowledge Tracing variants for all comparisons, with relatively little difference between different ways of fitting the classic BKT model (BKT-EM and BKT-BF) or a recent variant, BKT-PPS. More recent approaches (e.g. PFA, CFAR, Tabling)

performed substantially worse than BKT variants on all comparisons. In the case of PFA, these findings contradict other recent research [7, 13] which found that PFA performed better than BKT. However, as in that previous research, the differences between PFA and BKT were relatively small, suggesting that either of these approaches (or for that matter, most variants of BKT) are acceptable methods for student modeling. It may be of greater value for future student modeling research to attempt to investigate the question of *when* and *why* different student model frameworks have greater predictive power, rather than attempting to answer which framework is best overall.

Interestingly, among BKT variants, BKT-CGS performed quite poorly. One possible explanation is that this data set had relatively little data and relatively few skills, compared to the data sets previously studied with this method [e.g. 1], another potential reason why it may make sense to study whether these results replicate within a larger data set. BKT-CGS has previously performed poorly on other data sets from this same tutor [2], perhaps for the same reason. However, the low predictive power of average contextual slip for the post-test does not contradict the finding in [2] that average contextual slip plus BKT predicts the post-test better than BKT alone; in that research, these two models were combined at the post-test level rather than within the tutor. In general, average contextual slip was a productive component of ensembling models (as the third feature selected in each fold) despite its poor individual performance, suggesting it may be a useful future component of student models.

Overall, this paper suggests that Bayesian Knowledge-Tracing remains a highly-effective approach for predicting student knowledge. Our first attempts to utilize ensembling did not perform substantially better than BKT overall; however, it may be that other methods of ensembling will in the future prove more effective.

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

1. Baker, R.S.J.d., Corbett, A.T., Aleven, V.: 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. (2008)
2. 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.: 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. (2010)
3. Brusilovsky, P., Millán, E.: User models for adaptive hypermedia and adaptive educational systems. In: P. Brusilovsky, A. Kobsa and W. Neidl (eds.): The Adaptive Web: Methods and

Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321, Berlin Heidelberg New York: Springer-Verlag, pp. 3-53 (2007)
4. Chang, K.-m., Beck, J., Mostow, J., Corbett, A.: A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. In: Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 104-113. (2006)
5. Corbett, A., Kauffman, L., Maclaren, B., Wagner, A., Jones, E.: A Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. Journal of Educational Computing Research, 42, 219-239 (2010).
6. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction, 4, 253-278. (1995)
7. Gong, Y., Beck, J.E., Heffernan, N.T.: 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 (2010).
8. Koedinger, K. R., Corbett, A. T.: 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. (2006)
9. Pardos, Z. A., Heffernan, N. T.: 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 (2010)
10. 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, 161-170 (2010).
11. 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.
12. Pavlik, P.I., Cen, H., Koedinger, K.R.: 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 (2009).
13. Pavlik, P.I., Cen, H., Koedinger, K.R.: Performance Factors Analysis – A New Alternative to Knowledge Tracing. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, 531-538 (2009). 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.
14. Rai, D, Gong, Y, Beck, J. E.: Using Dirichlet priors to improve model parameter plausibility. In: Proceedings of the 2nd International Conference on Educational Data Mining, Cordoba, Spain, 141-148 (2009)
15. Reye, J.: Student modeling based on belief networks. International Journal of Artificial Intelligence in Education 14, 1-33. (2004)
16. Caruana, R., Niculescu-Mizil, A.: Ensemble selection from libraries of models. In: Proceedings of the 21st International Conference on Machine Learning (ICML'04), (2004).
17. Wang, Q.Y., Pardos, Z.A., Heffernan, N.T.: Fold Tabling Method: A New Alternative and Complement to Knowledge Tracing. Manuscript under review.
18. Yu, H-F., Lo, H-Y., Hsieh, H-P., Lou, J-K., McKenzie, T.G., Chou, J-W., et al.: Feature Engineering and Classifier Ensemble for KDD Cup 2010. Proceedings of the KDD Cup 2010 Workshop, 1-16 (2010)