Towards Automatically Detecting Whether Student Learning is Shallow

Ryan S.J.d. Baker[1], Sujith M. Gowda[1], Albert T. Corbett[2], Jaclyn Ocumpaugh[1]

[1]Department of Social Science and Policy Studies, Worcester Polytechnic Institute, Worcester, MA USA
{rsbaker, sujithmg, jocumpaugh}@wpi.edu
[2] Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA USA
corbett@cmu.edu

Abstract. Recent research has extended student modeling to infer not just whether a student knows a skill or set of skills, but also whether the student has achieved robust learning – learning that leads the student to be able to transfer their knowledge and prepares them for future learning (PFL). However, a student may fail to have robust learning in two fashions: they may have no learning, or they may have shallow learning (learning that applies only to the current skill, and does not support transfer or PFL). Within this paper, we present an automated detector which is able to identify shallow learners, who are likely to need different intervention than students who have not yet learned at all. This detector is developed using a step regression approach, with data from college students learning introductory genetics from an intelligent tutoring system.

Keywords: robust learning, student modeling, educational data mining, intelligent tutoring system

1 Introduction

Over the last two decades, student models have become effective at predicting which skills a student knows at a given time [cf. 16, 21, 22]. Recent research has gone beyond this to also assess the robustness of student learning [20] – whether students will be able to transfer their knowledge, whether they will be prepared for future learning, and whether they will retain their knowledge over the long-term. [8] presents a model that infers whether a student will perform well on a transfer test after using the tutor software – where the student must succeed at a related skill not taught in the tutor. Similarly, [9] presents a model that infers whether a student will be able to learn a new but related skill from an instructional text, after using the tutor. These types of models represent a step towards intelligent tutoring systems that can respond not just to whether a student has learned a skill, but to whether the student has achieved robust learning that will help them apply the knowledge broadly, in novel situations going forward.

However, while this work is a step towards modeling and remediation of robust learning, it is not sufficient to enable sophisticated differential intervention for shallow and robust learners. The robust learning detectors in [8, 9] only measure the extent to which a student has acquired robust learning. If the student has not acquired robust learning, these detectors cannot differentiate between a student who has acquired shallow knowledge (where the student knows the skills taught in the tutor, but cannot transfer those skills and is not prepared for future learning) and a student who has not learned at all. A student who has not learned at all may simply need more tutor practice [cf. 15], whereas a student who has shallow learning may need support in building from their procedural skill to deeper conceptual understanding. There are now interventions which have been shown to help students acquire robust learning [cf. 11, 13, 23, 24, 27], but not all students may need such interventions. A detector which can identify a student who has shallow learning, when combined with such interventions, may have the potential to enable richer intervention and better learner support than is currently possible.

As a step towards this vision, this paper presents a model designed to identify shallow learners, within a Cognitive Tutor for Genetics problem-solving [17]. This model is generated using a combination of feature engineering and step regression, and is cross-validated at the student level (e.g. repeatedly trained on one group of students and tested on other students). We report this detector's effectiveness at identifying shallow learners, and analyze its internal features, comparing them to features previously used to predict transfer and preparation for future learning (PFL).

2 Data Set

The data analyzed in this study come from 71 undergraduates using the Genetics Cognitive Tutor [17]. The Genetics Cognitive Tutor consists of 19 modules that support problem solving across a wide range of topics in genetics. Various subsets of the 19 modules have been piloted at 15 universities in North America. This study focuses on the data from a tutor module that employs a gene mapping technique called *three-factor cross*, in which students infer the order of three genes on a chromosome based on offspring phenotypes, as described in [5]. The data used in this analysis, first published in [8], were produced by students who were enrolled in genetics or introductory biology classes at Carnegie Mellon University.

These students used Cognitive Tutor-supported activities in two one-hour laboratory sessions, on successive days. In each session, students completed standard three-factor cross problems. During the first lab session, some students piloted cognitive-tutor activities designed to support deeper understanding; however, no differences were found between conditions for any robust learning measure, so in this analysis we collapse across the conditions and focus solely on student behavior and learning within the standard problem-solving activities.

The 71 students completed a total of 22,885 problem-solving actions across 10,966 problem steps in the tutor. Four paper-and-pencil post-tests followed the tutor activities [cf. 5]. Three tests were given immediately after tutor usage: a straightforward problem-solving post-test, a transfer test, and a test of preparation for future learning. A retention test was administered one week later.

Within this paper we focus analysis on the immediate problem-solving post-test, and the transfer test of robust learning. The problem-solving post-test consisted of two problems, and had two test forms, counterbalanced with the pre-test. Each of the two problems on each test form consisted of 11 steps involving 7 of the 8 skills in the three-factor cross tutor lesson, with two skills applied twice in each problem and one skill applied three times. The transfer test included two problems intended to tap students' understanding of the underlying processes of three-factor cross. The first was a three-factor cross problem that could not be solved with the standard solution method and required students to improvise an alternative method. The second problem asked students to extend their reasoning to four genes. It provided a sequence of four genes on a chromosome and asked students to reason about the crossovers that must have occurred in different offspring groups.

Students demonstrated successful learning in this tutor, with an average pre-test performance of 0.31 (SD=0.19), and an average post-test performance of 0.81 (SD=0.18). Students were also successful on the transfer test, with an average score of 0.85 (SD=0.18). The correlation between the problem-solving post-test and the transfer test was 0.59, suggesting that, although problem-solving skill and transfer skill were related, transfer may be predicted by more than just simply skill at problem-solving within this domain.

3 Shallowness Detector

3.1 Label Generation

The first step towards developing a data-mined model to predict which students have shallow learning is to create an operational definition of shallow learning that can be used as a training label (e.g. a "ground truth" label of the construct being predicted) for our shallowness detector. We employed data from the post-test of problem-solving skill and the transfer test posttest to do this. We operationalized shallow learning as the difference between a student's problem-solving test score and their transfer test score. Better performance on the problem-solving test than the transfer test indicates the student has acquired basic problem-solving knowledge, but in a shallow fashion, without the deep understanding that enables the application of that knowledge in novel situations.

Given the approximately equal average performance on the two tests, we can take simple percent correct on each test to assess whether a student is a shallow learner or

not (if the tests had radically different average performance, it might be better to use percentile rank on each test, or Z scores). As such, the present analysis treats students who achieve higher scores on the problem-solving post-test than on the transfer test as having shallow learning.

According to this operational definition, 24 of the 71 students in this study are labeled as shallow learners. Of the remaining 47 students, treated as not having shallow learning, 17 had perfect scores on both the transfer test and post-test. No other students had the same score on the two tests. The other 30 students had higher scores on the transfer test than the post-test. Among the 24 students labeled as shallow learners, there was an average of a 0.14 point difference between performance on the two tests (standard deviation = 0.10), with an average score of 0.87 on the problem-solving post-test, and an average score of 0.73 on the transfer test.

3.2 Data Features

The next step in our process of developing a model that could automatically identify shallow learning was to identify properties of students' problem-solving actions in the Cognitive Tutor that may be hallmarks of shallow learning. Towards this end, we selected a set of action-level features based on a combination of theory and prior work to model and detect related constructs. In particular, prior research on detectors of transfer [8] and PFL [9] influenced our design of features. As in that work, we can infer which students had shallow learning, using the method discussed in the previous section; but we do not know exactly what actions are associated with the shallow learning in advance. Hence, we take features calculated at the level of actions, and aggregate them across actions. We do so using two kinds of computations: the proportion of time specific behaviors occurred, and average quantitative values across actions. The 24 features used in this analysis included two categories of basic features, and two categories of complex features.

The first category of basic features focused on overall response time and time spent processing tutor-provided assistance, including: (1) average response time, (2) the average unitized response time (in standard deviations above or below the mean for students on the current skill), (3) the proportion of actions that involved a fast response after the student received a bug message (bug messages indicate why the system thinks the student made an error), (4) the proportion of slow responses after a bug message, (5) the proportion of fast responses after requesting a hint, (6) the proportion of slow responses after requesting a hint, (7) the proportion of slow actions after receiving a hint and entering a correct answer [cf. 25], and (8) the proportion of fast actions after receiving a hint and entering a correct answer.

The second category of basic features focused on the content of a student action: (9) the proportion of correct answers, (10) the proportion of help requests, and (11) the proportion of answers that were incorrect and received bug messages.

The first category of complex features involved Bayesian Knowledge Tracing estimates of the student's knowledge of relevant skills and performance probabilities [16]: (12) the average probability the student knew the skill, (13) the average probability the student would give a correct answer according to the model, (14) fast actions on well-known skills, and (15) slow actions on well-known skills.

The second category of complex features focused on features derived from previous research on meta-cognition and disengagement: (16) help avoidance, the proportion of actions where the skill was not known and help was not sought [cf. 2], (17) the proportion of actions where the skill was known and help was not sought, (18) fast actions not involving gaming the system [using the detector from 6], (19) slow actions not involving off-task behavior [using the detector from 3], (20) the average contextual probability that an error was due to slipping [cf. 4], (21) the average contextual probability of slip among actions with over 50% probability of being a slip (called "certainty of slip") [cf. 5], (22) the average contextual probability that a correct response was a guess [cf. 4], (23) the "certainty of guess" (corresponding to certainty of slip), and (24) the average moment-by-moment learning [cf. 7].

Some of these features relied upon cut-offs; in these cases, an optimized cut-off was chosen using a procedure discussed in the next section.

3.3 Detector Development

We fit detectors of shallowness using step regression models. (Note that step regression is not the same as step-wise regression.) Step regression involves fitting a linear regression model to predict the labels of shallowness using the features of student behavior in the tutor, and then thresholding that model's predictions with a pre-chosen cut-off, in this case 0.5. Within this statistical framework, all students for whom the linear regression predicted values of 0.5 or higher are assessed to have non-shallow learning, whereas all students for whom the linear regression predicted values below 0.5 are assessed to have shallow learning. The choice of 0.5 is an arbitrary standard convention (0.5 is halfway between 0 and 1); so long as the step cut-off is chosen prior to model fitting, equal performance can be achieved for any step cut-off (different step cut-offs are adjusted for by the constant term of the equation). Hence, this framework takes numerical predictions of shallowness and transforms them into a binary prediction of whether the student's learning is shallow or not, which can be compared to the labels initially derived from the two tests.

These detectors of shallowness are assessed using 10-fold student-level cross-validation [18]. In 10-fold cross-validation, the data points are divided into ten groups (in this case divided by students), each of which serves successively as a test set. That is, for each of the ten groups, the other nine groups are used to produce a model, and then the tenth group is used to test that model. Hence, each model's goodness is never

tested on the same students it was trained on, but each model is tested on every student. Because this process does not exclude any data points (or students) from the modeling process, cross-validation is typically preferred to holding out a test set that is entirely excluded from model development. Cross-validated performance assesses the model's predictive performance when applied to new data, an indicator of the model's ability to generalize.

Two metrics were used as the assessment of goodness for each model: (1) A' (also called AUC, for "Area Under [the ROC] Curve") [19], and (2) Cohen's [14] Kappa, or κ. A' is the probability that if the detector is comparing two students, one labeled as having shallow learning and the other one not labeled as having shallow learning, it will correctly identify which clip is which. A' is mathematically equivalent to W, the Wilcoxon statistic [19]. A model with an A' of 0.5 performs at chance, and a model with an A' of 1.0 performs perfectly. In these analyses, A' was computed using the AUC (area under the curve) method. Cohen's Kappa (κ) assesses whether the detector is better than chance at identifying the correct action sequences as involving the category of interest. A Kappa of 0 indicates that the detector performs at chance, and a Kappa of 1 indicates that the detector performs perfectly. A' and Kappa both compensate for the possibility that successful classifications can occur by chance [cf. 10]. A' can be more sensitive to uncertainty in classification than Kappa, because Kappa looks only at the final label whereas A' looks at the classifier's degree of confidence in classifying an instance.

We fit two detectors. The first detector uses only the individual features discussed above in section 3.2. Some of the features, involving proportions of specific types of actions, depend on a threshold parameter (such as how many seconds differentiates a "long pause" from a "short pause"); these parameters were optimized by computing the single-feature step regression model for a range of potential thresholds (see [8] for more details) and selecting the threshold with the best A' value. The second detector also includes multiplicative interactions between the individual features. In order to reduce the potential for over-fitting (where a set of features does not generalize well to data from new students), we reduce the parameter space of both models prior to fitting full models. The individual feature model is limited to considering features for which a single-feature step regression model has a better value for the Akaike information criterion (AIC [1]) than the empty model, reducing the data space from 24 features to 11 features. The multiplicative interaction model only considers the 66 interactions of those 11 features, and furthermore discards features that fail the same Akaike test, resulting in a set of 35 multiplicative interaction features, plus 11 individual features, for a total data space of 46 features.

We used Forward Selection to find the best model for each one of the two feature sets. In Forward Selection, the best single-parameter model 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. In this case, the goodness criterion for model

selection was cross-validated Kappa.

4 Results

The best-fitting models for each feature set are as follows:

Table 1: Step regression models with student-level cross-validated A' and Kappa (higher values of model coefficients correspond to non-shallow learners)

Model Type	Model	A'	Kappa
Multiplicative-Interactions	2221 * SlowResponseAfterBugMsg - 0.22 * AverageCertaintyOfSlip * AvgTime + 1.03	0.758	0.389
No-Interactions	34.74 * SlowResponseAfterBugMsg - 1232 * AvgTimeSD + 0.6726	0.767	0.346

As can be seen in Table 1, the multiplicative-interactions model achieves moderately better cross-validated Kappa than the no-interactions model, and slightly worse cross-validated A'. The model with multiplicative interactions achieved an acceptable cross-validated kappa of 0.389 (39% better than chance according to the baseline [cf. 14]). It is worth noting that kappa values typically achieved in data mining are usually lower than kappa values achieved in inter-rater reliability checks among human coders; the standards are different because the goals are different. The agreement between a data-mined model and a construct which is itself noisy will inherently be lower than human agreement on a tightly-defined construct. The A' value for the multiplicative-interactions model is 0.758, which indicates that the model can differentiate a student who performs better on the problem-solving test than the transfer test from a student who does not perform better on the problem-solving test than the transfer test, 75.8% of the time. This level of performance on the A' metric is typically considered to be sufficient to enable fail-soft intervention. This level of performance is significantly better than chance, $Z=-3.56$, $p<0.001$, using the test from [19].

The features that constitute the two models are similar, and both models are quite simple. In both models, the first feature is slow responses after bug messages. The positive coefficient for this feature indicates that students who pause when receiving bug messages are less likely to be shallow learners. Bug messages in this tutor lesson tell students about what order to complete steps in, and which information is necessary to draw valid conclusions. As such, reflective pauses upon receiving these messages may indicate a student trying to understand why certain information is necessary at specific steps in the reasoning process. It seems reasonable that these

reflective pauses would be associated with more robust learning. This feature is also associated with a greater probability of transfer, in the same tutor lesson [8].

The second feature in both models involves average response time. This feature has a negative coefficient in both models, indicating that in general slow response times are associated with shallow learning. Shallow learners are not characterized by fast guesses (which may lead to no learning at all), but just the opposite – they seem rather to be struggling compared to students achieving robust learning. More specifically, average response speed relative to all students enters into the individual feature model. In the multiplicative-interactions model, response speed enters the model as an interaction with the average certainty of slip (the probability of slip among actions that are likely to be slips, an error despite knowing the skill). The average certainty of slip has been previously shown to predict final tutor knowledge, even after controlling for predictions from Bayesian Knowledge Tracing [5]; as such, it makes sense that this feature may be related to the depth of learning. While the more common interpretation of a slip is carelessness, an alternative interpretation is that a slip indicates imperfect acquisition of skill, where a student's skill knowledge works on some problems but not on others [cf. 4]. Such lack of transfer within even basic problem solving would be consistent with shallow learning.

5 Discussion and Conclusions

Within this paper, we have presented models that can distinguish with reasonable accuracy whether a student has acquired shallow learning, operationally defined as performing better on a test of the material learned in the tutor, than on a test of the ability to transfer that skill to related problems. These models are developed in the context of a Cognitive Tutor for Genetics, and cross-validated at the student level; exploring this model's generality to other learning domains and types of educational software is an important area of future work.

The better of these models can distinguish a shallow learner from a non-shallow learner 76% of the time, performing 39% better than chance. These models are based on three features of the student's interactions with the learning software, including two found in both models: the speed of student actions, and the speed of student responses after receiving bug messages. A third feature, probable slips during performance, is only found in the multiplicative-interactions model. As with the previous model of transfer [cf. 8], how students respond to evidence that they do not understand the skill (bug messages) appears to be particularly important for modeling shallow learning. This result is in line with theory that suggests a key role for meta-cognition in robust learning [20]; it also suggests that student responses to bug messages – not currently a key aspect of theoretical models of meta-cognition in intelligent tutoring systems [cf. 2] – deserves a more prominent place in future theoretical models.

Shallowness detectors have considerable potential usefulness for intelligent remediation. Students who have learned the exact skills taught in the tutor but who have not achieved robust learning are a group especially in need of remediation. Traditional student modeling methods are likely to fail to provide them any remediation, as they have learned the skills being taught by the tutor and can demonstrate that skill. A detector of shallow learning can identify these students and offer them remediation specific to their needs, helping a student to build on his or her procedural knowledge to achieve the conceptual understanding necessary for future use of their knowledge. Thus, we view this detector as a second step – building on the first step of transfer and PFL detectors – towards educational software that can predict and respond automatically to differences in the robustness of student learning, an important complement to ongoing research on designing educational software that promotes robust learning [cf. 11, 13, 23, 24, 26].

Acknowledgements. The authors thank award #DRL-0910188 from the National Science Foundation, "Empirical Research: Emerging Research: Robust and Efficient Learning: Modeling and Remediating Students' Domain Knowledge".

6 References

1. Akaike. H: A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19 (6), 716-723. (1974)
2. Aleven, V., McLaren, B.M., Roll, I. Koedinger, K.R.: Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. International Journal of Artificial Intelligence and Education, 16, 101-128. (2006)
3. Baker, R.S.J.d.: Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. Proceedings of ACM CHI 2007: Computer-Human Interaction, 1059-1068. (2007)
4. 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. Proceedings of the 9th International Conf. on Intelligent Tutoring Systems, 406-415. (2008)
5. 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. Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization, 52-63. (2010)
6. Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a generalizable system to detect when students game the system. User modeling and User-Adapted Interaction, 18 (3), 287-314. (2008)
7. Baker, R.S.J.d., Goldstein, A.B., Heffernan, N.T.: Detecting the Moment of Learning. Proceedings of the 10th Annual Conference on Intelligent Tutoring Systems, 25-34. (2010)
8. Baker, R.S.J.d., Gowda, S., Corbett, A.T.: Towards predicting future transfer of learning. Proc. of the 15th International Conf. on Artificial Intelligence in Education, 23-30. (2011)

9. Baker, R.S.J.d., Gowda, S., Corbett, A.T.: Automatically Detecting a Student's Preparation for Future Learning: Help Use is Key. Proceedings of the 4th International Conference on Educational Data Mining, 179-188. (2011)
10. Ben-David, A.: About the Relationship between ROC Curves and Cohen's Kappa. Engineering Applications of Artificial Intelligence, 21, 874-882. (2008)
11. Butcher, K.R.: How Diagram Interaction Supports Learning: Evidence from Think Alouds During Intelligent Tutoring. LNCS 6170, 295-297.
12. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. Proc. of the International Conference on Machine Learning, 161-168. (2006)
13. Chin, D.B., Dohmen, I.M., Cheng, B.H., Oppezzo, M.A., Chase, C.C., Schwartz, D.L.: Preparing Students for Future Learning with Teachable Agents. Educational Technology Research and Development, 58 (6), 649-669. (2010)
14. Cohen, J. A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37-46. (1960)
15. Corbett, A.: Cognitive computer tutors: Solving the two-sigma problem. UM2001, User Modeling: Proceedings of the Eighth International Conference, 137–147. (2001)
16. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4, 253-278. (1995)
17. Corbett, A.T., MacLaren, B., Kauffman, L., Wagner, A., Jones, E. A.: Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. Journal of Educational Computing Research, 42 (2), 219-239. (2010)
18. Efron, B. & Gong, G.: A leisurely look at the bootstrap, the jackknife, and cross-validation. American Statistician, 37, 36–48. (1983)
19. Hanley, J., McNeil, B.: The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. Radiology, 143, 29-36. (1982)
20. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The Knowledge-Learning-Instruction (KLI) Framework: Toward Bridging the Science-Practice Chasm to Enhance Robust Student Learning. Cognitive Science. (in press)
21. Martin, J., VanLehn, K.: Student assessment using Bayesian nets. International Journal of Human-Computer Studies, 42, 575-591. (1995)
22. Pavlik, P.I., Cen, H., Koedinger, K.R.: Performance Factors Analysis – A New Alternative to Knowledge Tracing. Proceedings of the 14th International Conference on Artificial Intelligence in Education, 531-540. (2009)
23. Roll, I., Aleven, V., McLaren, B. M., Koedinger, K. R.: Can help seeking be tutored? Searching for the secret sauce of metacongitive tutoring. Proceedings of the 13th International Conference on Artificial Intelligence in Education. (2007)
24. Salden, R. J. C. M., Aleven, V., & Renkl, A., Schwonke, R. Worked Examples and Tutored Problem Solving: Redundant or Synergistic Forms of Support? Proceedings of the 30th Annual Conference of the Cognitive Science Society, 589-594. (2008)
25. Shih, B., Koedinger, K.R., Scheines, R.: A response time model for bottom-out hints as worked examples. Proceedings of the 1st International Conference on Educational Data Mining, 117-126. (2008)
26. Tan, J., Biswas, G.: The Role of Feedback in Preparation for Future Learning: A Case Study in Learning by Teaching Environments. Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 370-381. (2006)