Predicting Students' Unproductive Failure on Intelligent Tutors in Adaptive Online Courseware

Seoyeon Park and Noboru Matsuda
Department of Teaching, Learning and Culture
Texas A&M University

INTRODUCTION

The wheel-spinning phenomenon in the current paper refers to students' unproductive failure within a computer-based learning environment using Intelligent Tutoring Systems (ITSs). Beck and Gong (2013) found that students often spend a considerable amount of time practicing a skill in ITSs without making progress. This phenomenon is coined *wheel spinning* because students' learning pattern is like a car stuck in the mud. The wheel-spinning phenomenon has been observed universally on many ITSs (Beck & Gong, 2015). When wheel spinning, students often become frustrated and demotivated to learn (Cen, Koedinger, & Junker, 2007; Baker, Gowda, & Corbett, 2011). Therefore, several studies explored building an effective and reliable wheel-spinning detector to detect the moment of wheel spinning. Beck and Gong (2015) suggested a generic model using logistic regression to predict wheel spinning with three aspects: student's performance on the skill, the seriousness of the learner, and general factors of the learning material such as skill difficulty. Matsuda, Chandrasekaran, and Stamper (2016) built a more simplified wheel-spinning predictor as a combination of the probability of mastery based on Bayesian knowledge tracing, and a neural-network model.

In the current paper, we investigate the wheel-spinning phenomena in the context of adaptive online courseware where many ITSs are embedded into the online courseware. Students are provided with multimedia instruction, including paragraph text instruction, images, videos, and traditional formative assessments such as multiple choice and fill-in-the-blank questions. ITSs are embedded in the courseware as a type of formative assessment as well. In this rich learning environment, we aim to predict the moment of wheel spinning so that the system can provide proactive scaffolding to maintain students' motivation and engagement.

The goal of the current paper is to contribute to the Generalized Intelligent Framework for Tutoring (GIFT) framework by investigating the wheel-spinning phenomena on the adaptive online course platform with many ITSs on which wheel spinning will happen. We discuss the unique nature of the wheel-spinning in this environment and our current progress. The current work is part of our on-going project where we develop evidence-based learning-engineering methods to build adaptive online courseware, called PASTEL (**P**ragmatic methods to develop **A**daptive and **S**calable **T**echnologies for next generation **E**-**L**earning).

The existing models for wheel-spinning detection have some limitations. First, existing models have low recall rates around 0.25-0.50, suggesting that these models are weak and can only detect less than half of actual wheel-spinning cases. Since not catching a moment of wheel spinning would impact students' motivation, we need to develop a model that has a high sensitivity to wheel spinning.

Second, most of the existing models are aimed to *detect* a moment of wheel spinning, instead of *predicting* students who are likely to get stuck. Matsuda et al. (2016) applied a neural-network model to predict wheel spinning at an early stage of learning. However, its prediction power is approximately 0.25, which is still insufficient for practical use. The primary purpose of catching wheel spinning is to maintain students' motivation for learning, it is crucial to *predict* the moment of wheel spinning in advance. With the early prediction, we can provide students with proactive scaffolding that keep those students from experiencing wheel spinning.

Third, existing wheel-spinning detectors/predictors explain wheel spinning on individual skills (the skill-level model), indicating the likelihood of a student to fail to obtain mastery on a particular skill. Historically speaking, this trend has been held because problems on ITSs are broken down into a fine-grained skill set, often called a knowledge component (KC) model (Koedinger, Corbett, & Perfetti, 2012). Taking skills as a unit of analysis works well for a "standalone" ITS (including ITS with "units"). As mentioned above, we target the adaptive online courseware as the platform for wheel-spinning prediction. During our initial trial for creating an instance of adaptive online courseware (called *CyberBook*) with in-service teachers as curriculum consultants, we asked in-service teachers to tag each ITS with the most essential skill that students will learn by solving problems on a corresponding ITS. We observed that in-service teachers often tagged an ITS with a skill that does not appear in any steps on the ITS (as opposed to selecting one of the steps on an ITS as the most essential step hence the most essential skill). For example, an ITS that teaches how to compute the slope of a given linear equation involves steps such as subtracting and dividing terms, but no single step is about "computing the slope." On CyberBook, when a student gets stuck (i.e., wheel spins) on a particular ITS, the system provides the student with proactive scaffolding by showing a link to the related instruction paragraph. A naïve research question therefore is: *Should wheel spinning be predicted on steps within an ITS (hence triggers the proactive scaffolding) or on the ITS as a whole?* Given our observations from in-service teachers tagging ITSs with a skill, we hypothesized that the ITS as a whole should be the unit of analysis for wheel-spinning prediction.

The goal of the current study is to develop a wheel-spinning predictor, which can distinguish students who have a high possibility to wheel-spin as quickly as possible, at the problem level. The specific research questions are as follows:

1. How accurately can we predict wheel-spinning at the problem level?
2. How early can we detect wheel-spinning at the problem level?

To build a wheel-spinning prediction model that can find wheel-spinning cases with high accuracy and speed, we propose to use four general factors, students' performance, hint usage, the sum of response time, and difficulty of each problem type. These factors are generally available on most ITSs and are known to be effective in predicting students' academic performance. We have previously built a wheel-spinning predictor at the step-level (Park & Matsuda, under review). In the current paper, to understand whether the problem-level prediction is any better than step-level prediction, we apply logistic regression and an ensemble modeling to predict wheel-spinning cases at the problem-level.

DATA PREPROCESSING

We used an existing dataset from DataShop, entitled 'Cog Model Discovery Experiment Spring 2010' in the 'Geometry Cognitive Model Discovery Closing-the-Loop study' project. There were 49 skills forming 45,597 observations done by 123 students in the 'KTracedSkills' model in this data. This dataset contained 5,279 student-skill pairs. The DataShop data uses fine-grained skills that are decomposed by Learning Factor Analysis. In order to predict wheel-spinning at the problem level, we needed to create 'problem type' as a different dimension of measuring wheel-spinning. We used a text-mining technology named SMART to create 'problem type'. SMART is an AI technology that can compute the similarity among words within the text and extract a keyword. We input hint message of each intelligent tutor and set an arbitrary k number; k=25, 50, 75, 100. After SMART generates problem types, those problem type models were validated with the DataShop knowledge component model. Table 1 shows the result of comparing SMART generated problem type models.

Table 1. Comparison of SMART generated problem type models

Model name	Problem types	Observations with Problem types	AIC	BIC	RMSE (student stratified)	RMSE (item stratified)
SMART k=25	17	85,115	46,986.00	48,454.30	0.273130	0.271673
SMART k=50	28	85,115	46,787.87	48,461.83	0.272680	0.271298
SMART k=75	40	85,115	47,114.50	49,012.91	0.274457	0.272230
SMART k=100	39	85,115	47,145.30	49,025.00	0.273595	0.272066
KTracedSkills	49	41,756	29,096.28	31,005.13	0.333781	0.324864

KTracedSkills row is the baseline when comparing other SMART generated problem type models. We chose to use the problem type model named 'SMART k=50' because this model shows the lowest root mean squared error (RMSE). Comparing to KTracedSkills model, 'SMART k=50' has bigger AIC and BIC, but these figures are affected by the number of observations. Considering that the number of observations of our SMART generated problem type models is more than twofold, the AIC and BIC figures make sense.

We employed the 'SMART k=50' problem type model and did data preprocessing. There were 28 problem types and we created 1,889 student-problem type pairs. Mastery in this study is defined as three consecutive correct responses on one's first attempt within 10 practice opportunities (Beck and Gong, 2013) on a problem level. We filtered out "indeterminate" students, who did not practice on enough opportunities, which was 10 opportunities in this study, for us to define their mastery (Beck & Gong, 2015). After removing indeterminate student-problem type pairs, this dataset came to contain 1,794 student-problem type pairs and 31,801 observations with 123 students. The dependent variable is whether a student shows mastery (M) or wheel-spinning (W) on a problem type within 10 opportunities, based on the response sequences of each student-problem type pair. In order to see how early we can predict wheel-spinning on a problem type, we made subset at each practice opportunity from the third opportunity to the ninth opportunity.

FEATURES

We used four features that are all general factors in any dataset of ITSs. This is because first, we want to show that predicting wheel-spinning at the problem level can be generalized among any ITS construct, and second, we want to build a more simple and scalable wheel-spinning model.

Student's performance on each problem type

The first feature we used is how well a student did on a problem type. This represents a student's ability to solve a certain type of problem. This is calculated as the average probabilities of correct first attempts per each student-problem type pair.

Problem type difficulty

The second feature is the difficulty of each problem type. We calculated this variable by getting the average correct response rate of each problem type across all students who practiced the problem type.

Max_hint

Hint usage is regarded as one of the important factors in explaining students' learning (Feng, 2009; Rivers, 2017). Thus, we used the maximum number of hint usage of students on each problem type.

Sum_duration

Response time is one of the key features in a wheel-spinning model (Beck and Gong, 2015). Each problem type has several steps, so we added step duration of constituent steps to get the response time of a student on each problem type.

PREDICTION MODELS AND RESULTS

A basic model for wheel-spinning prediction at the problem level

With the combination of features above, we trained a logistic regression to build a basic model for wheel-spinning prediction at the problem type level with ten-fold cross validation. The coefficients would not be suggested due to the limit of space. This basic model for wheel-spinning prediction shows high accuracy throughout practice opportunities in Table 2. The overall accuracy in percent correct is 92.75% and overall AUC is 0.916. Considering the accuracy of the generic wheel-spinning model in a skill level (Beck and Gong, 2015), which was less than 90% in percent correct and 0.9 in AUC, this basic model shows a sufficient performance with even using the smaller number of features.

Table 2. Accuracy of a basic model per practice opportunity

	opp3	opp4	opp5	opp6	opp7	opp8	opp9
Percent correct	0.908	0.91	0.917	0.923	0.932	0.949	0.950
AUC	0.856	0.863	0.894	0.939	0.938	0.949	0.975

We not only need to see the accuracy of this model but also precision and recall rates in order to have an insight into its classification. Table 3 shows the precision and recall rate of this model at each opportunity. Both rates are increasing by each opportunity. However, the precision rate is 60% and recall rate is 33.65% on average across the third through ninth opportunity. These figures are relatively low comparing to those of existing wheel-spinning models (around 70% in precision rate and 25~50% in recall rate). Moreover, using this basic model, we cannot predict wheel-spinning on a problem type level as early as possible due to its weak recall rate in every practice opportunity.

Table 3. Precision and Recall rates of a basic model per practice opportunity

	opp3	opp4	opp5	opp6	opp7	opp8	opp9
Precision	0.358	0.440	0.551	0.619	0.666	0.755	0.814
Recall	0.0798	0.176	0.230	0.285	0.417	0.612	0.554

The upgraded model for wheel-spinning prediction at the problem level using gradient boosted decision tree model

We found that the basic model has some limitations in terms of its precision and recall rates. Thus, other data mining techniques were explored to find a better prediction model. Especially, we focused on getting a higher recall rate in the early phases so that we can predict wheel-spinning on a problem level as quickly as possible. We discovered that the gradient boosted decision tree model using the same combination of features shows much better performance in accuracy, precision, and recall rate. Gradient boosted trees is an ensemble of multiple tree models to create a powerful prediction model for classification. This algorithm generates a series of trees where trees are made by correcting poor predicted examples of the previous trees in the series. We trained this model with a ten-fold cross validation by each practice opportunity. The overall accuracy of the upgraded model is 96.90% and 0.97 in AUC. Table 4 shows that this model shows higher accuracy throughout opportunities.

Table 4. Accuracy of the upgraded model per practice opportunity

	opp3	opp4	opp5	opp6	opp7	opp8	opp9
Percent correct	0.953	0.955	0.961	0.972	0.974	0.985	0.981
AUC	0.942	0.96	0.974	0.985	0.987	0.991	0.997

This model also has a much higher performance on both precision and recall rates than those of our basic model. Overall, the precision rate is 87% and recall rate is 75% across the third through ninth opportunity. These figures are showing that this upgraded model has greater wheel-spinning prediction power than other existing models. Applying this model, we can predict wheel-spinning on a problem type on students' fifth opportunity with 65% accuracy and over 80% accuracy on the sixth opportunity.

Table 5. Precision and Recall rates of the upgraded model per practice opportunity

	opp3	opp4	opp5	opp6	opp7	opp8	opp9
Precision	0.792	0.834	0.867	0.843	0.840	0.958	0.963
Recall	0.616	0.606	0.651	0.829	0.865	0.864	0.813

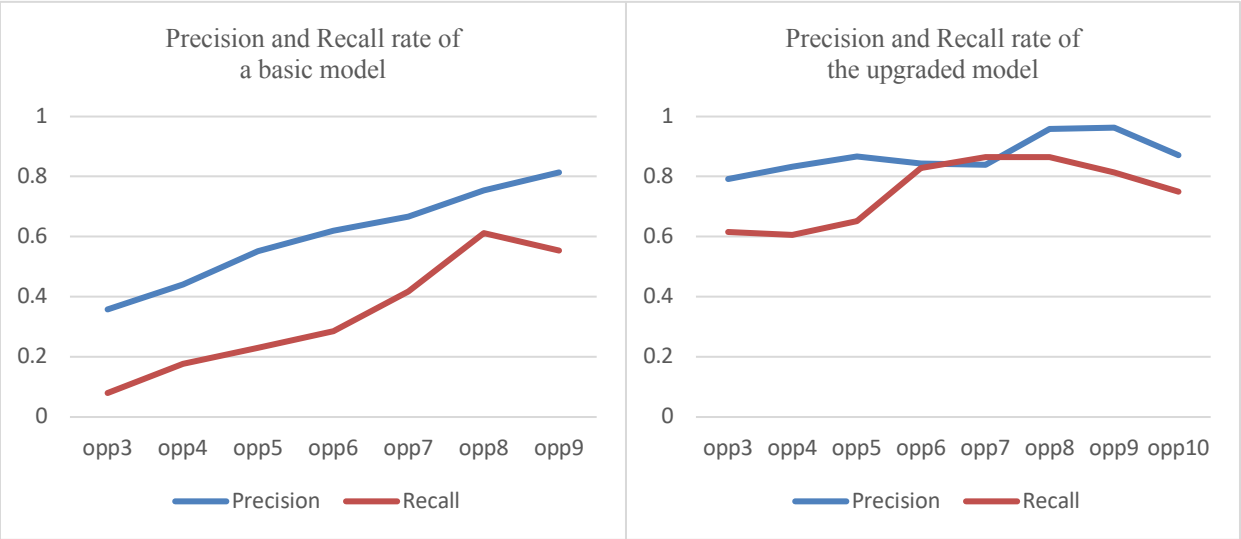

Figure 1. Precision and Recall rate of two models

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The goal of the current study is to seek the way in which we can predict wheel-spinning at the problem level (i.e., an individual ITS as opposed to a step in an ITS) with high accuracy, prediction power, and speed. We have some important findings in this work. First, we found that the four general variables (i.e., students' performance, hint usage, the sum of response time, and difficulty of each problem type) that are available for most ITSs can sufficiently build a prediction model for wheel spinning at the problem level. Our basic model with four general variables shows similar performance with existing models in its accuracy (average percent correct is 0.93 and overall AUC is 0.92). Its recall rate (0.34) is higher than that of the other wheel-spinning prediction model (Matsuda, Chandrasekaran, and Stamper, 2016).

Second, we explored other machine learning techniques to improve the accuracy of wheel-spinning prediction. Our upgraded model with gradient boosted decision tree algorithm shows enhanced precision and recall rate with an average recall rate of 0.75. A pragmatic merit of this upgraded model is its speed—the recall rate on the sixth practice opportunity is around 0.83. This would expand our chance to promote students' efficient learning in ITSs by keeping them from wheel spinning in advance.

As for the contribution to the Generalized Intelligent Framework for Tutoring (GIFT), the current study demonstrated a generic technique to predict students' unproductive failure (wheel spinning) on an ITS embedded into adaptive online courseware. The adaptive online courseware with embedded intelligent tutors has a tremendous potential for future online learning hence investigating fundamental techniques such as the wheel-spinning prediction plays an important role. We also demonstrated an importance of building the wheel-spinning predictor at the different level of granularity of the skill model.

For future study, one intriguing topic would be to find what we should do once we predict wheel-spinning cases. What would be an effective intervention for those who are predicted to wheel spin on a problem? Another suggestion is to explore other machine learning techniques to improve the current wheel-spinning prediction model. This study used logistic regression and gradient boosted decision tree. Our upgraded model using gradient boosted decision tree shows significant improvement in predicting wheel-spinning,

however, a drawback of using this technique is that it is hard to interpret the model itself. Finally, it would also be an interesting idea to extend the research regarding why students show unproductive failure in learning by using ITSs.

ACKNOWLEDGEMENT

This study is supported by NSF grants No. 1623702 and 1644430.

REFERENCES

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. American Psychologist, 51(4), 355.

Baker, R. S., Gowda, S. M., & Corbett, A. T. (2011). Towards predicting future transfer of learning. In International Conference on Artificial Intelligence in Education. Springer, Berlin, Heidelberg, 23-30.

Beck, J. E., & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. In International Conference on Artificial Intelligence in Education. Springer, Berlin, Heidelberg, 431-440.

Beck, J., Ostrow, K. & Wang, Y. (2016). Students vs. Skills: Partitioning Variance Explained in Learner Models. In The 9th International Conference on Educational Data Mining. ACM

Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis–a general method for cognitive model evaluation and improvement. In International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, 164-175.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In Icml. 96, 148-156.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

Gong, Y., & Beck, J. E. (2015). Towards detecting wheel-spinning: Future failure in mastery learning. In Proceedings of the Second (2015) ACM Conference on Learning@ Scale. ACM, 67-74.

Gong, Y., Wang, Y., & Beck, J. (2016). How long must we spin our wheels? Analysis of student time and classifier inaccuracy. Student modeling from different aspects, 32-38.

Heffernan, N., Heffernan, C., & Ostrow, K. (2018). The Assessments underlying ASSISTments. In Proceedings of the AERA 2018. NY.

Koedinger, K.R., Corbett, A.C., & Perfetti, C. (2012). The Knowledge-Learning-Instruction (KLI) framework: Bridging the science-practice chasm to enhance robust student learning. Cognitive Science, 36 (5), 757-798.

Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. AAAI/IAAI, 546-551.

Matsuda, N., Chandrasekaran, S., & Stamper, J. C. (2016). How quickly can wheel spinning be detected?. In EDM. 607-608.

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by doing. Instructional-design theories and models: A new paradigm of instructional theory, 2, 161-181.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In Nonlinear estimation and classification. Springer, New York, NY, 149-171.

Seymour, E., & Hewitt, N. M. (1997). Talking about leaving: Why undergraduates leave the sciences. Boulder, CO: Westview.

Stamper, J., & Ritter, S. (2010). Cog Model Discovery Experiment Spring 2010. Dataset 392 in DataShop. Retrieved from https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=392.

Park, S., & Matsuda, N. (under review). Early wheel-spinning detection in student performance on cognitive tutors using an ensemble model. Journal of Educational Data Mining.

Ritter, S., Anderson, J.R., Koedinger, K.R., & Corbett, A. (2007). The Cognitive Tutor: Applied research in mathematics education. Psychonomics Bulletin & Review, 14(2), 249-255.

Watkins, J., & Mazur, E. (2013). Retaining students in science, technology, engineering, and mathematics (STEM) majors. Journal of College Science Teaching, 42(5), 36-41.

ABOUT THE AUTHORS

Seoyeon Park *is a Ph.D. student at Texas A&M University and a research associate in the Innovative Educational Computing Laboratory, whose research interest is educational data mining and stem education with adaptive scaffolding in computer-based learning environments.*

Dr. Noboru Matsuda *is an Associate Professor of Cyber STEM Education at the Department of Teaching, Learning, and Culture; and the director of the Innovative Educational Computing Laboratory. He leads the NSF funded project on the data-driven learning engineering methods to build adaptive online courseware where the research team investigates scalable AI techniques to evidence-basely build adaptive online courseware.*