The Opportunity Count Model: A Flexible Approach to Modeling Student Performance

Yan Wang
Worcester Polytechnic Institute
Worcester, MA 01609, USA
ywang14@wpi.edu

Seth Adjei
Worcester Polytechnic Institute
Worcester, MA 01609, USA
saadjei@wpi.edu

Korinn Ostrow
Worcester Polytechnic Institute
Worcester, MA 01609, USA
ksostrow@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
Worcester, MA 01609, USA
nth@wpi.edu

L@S 2016, April 25-26, 2016, Edinburgh, Scotland UK
ACM 978-1-4503-3726-7/16/04.
http://dx.doi.org/10.1145/2876034.2893382

Abstract
Detailed performance data can be exploited to achieve stronger student models when predicting next problem correctness (NPC) within intelligent tutoring systems. However, the availability and importance of these details may differ significantly when considering opportunity count (OC), or the compounded sequence of problems a student experiences within a skill. Inspired by this intuition, the present study introduces the Opportunity Count Model (OCM), a unique approach to student modeling in which separate models are built for differing OCs rather than creating a blanket model that encompasses all OCs. We use Random Forest (RF), which can be used to indicate feature importance, to construct the OCM by considering detailed performance data within tutor log files. Results suggest that OC is significant when modeling student performance and that detailed performance data varies across OCs.

Author Keywords
Opportunity Count; Random Forest; Student Modeling; Next Problem Correctness; Intelligent Tutoring System

ACM Classification Keywords
I.6.5. Simulation and modeling: Model Development; K.3.0. Computers and education: General.

Helpful Definitions

Skill Builder: Skill Builders are a type of problem set within ASSISTments based on mastery learning. To master a skill, students must accurately answer three consecutive skill opportunities. High performing students are likely to have fewer OCs, while struggling students are likely to have higher OCs. As OC increases, data points grow scarcer as students master the skill or drop out of the assignment. Each Skill Builder is identified by a unique skill ID.

Random Forest: Random Forest is a proven method for making predictions when considering a variety of features [6]. The method trains regression trees based on decision splits made from a random subset of data features. The resulting output offers a prediction model based on an ensemble of regression trees. Random Forest can also be used to succinctly define the degree of importance a feature holds within a model.

Introduction

Despite the fact that detailed performance data has been shown to enhance student modeling within intelligent tutoring systems [1,2,5], little focus has been given to the critical significance of opportunity count (OC), or the compounded sequence of skill opportunities within a student's learning experience. It seems intuitive that the availability and importance of details within logged tutor data would vary based on OC: modeling certain performance details on a student's 3rd skill opportunity may reveal more about learning than modeling the same details on a student's 7th opportunity. Considering OC could also reduce the noise inherent to modeling few OCs through more flexible modeling of performance details.

The present work introduces the Opportunity Count Model (OCM) and investigates the significance of considering OC in student models. The OCM builds separate models for differing OCs by using Random Forest to determine fluctuations in the importance of student performance details across a dataset stratified by OC. We seek to validate the OCM, by answering the following research questions:

1. Can the accuracy of models predicting next problem correctness (NPC) be enhanced by aggregating separate models for differing OCs when considering additional details of a student's performance?
2. Is there variation in the importance of particular student performance details across OCs?

Dataset and Method

We examined the effectiveness of the OCM using a dataset comprised of student data logged between September 2012 and August 2013 within ASSISTments, a popular intelligent tutoring system for mathematics [3]. The dataset contained performance details for 85,862 problems logged by 3,210 unique students spanning 70 unique Skill Builders (see side panel).

The dataset included OC, student ID (St), skill ID (Sk), correctness (Ct), number of attempts (Att), first response time (FRT), percentage of hints requested (%H), first action (FA), and historical accuracy (% HA) (see final side panel for more detail). Table 1 depicts an example from the dataset. Student 34 has two skill opportunities for 'Skill 5', makes a single attempt on the first and two attempts on the second, and solves both in approximately 30 seconds. She immediately solved the first skill opportunity but required 60% of available hints on the second skill opportunity, requesting a hint before even attempting to solve the problem (FA = 1). Percentage values were discretized by units of 20% to simplify our modeling approach. For example, HA of 64% was discretized to 60%.

We used MATLAB's implementation of Random Forest (RF, see side panel) to build the OCM and make predictions about student performance [4]. We divided the dataset into training and testing segments, and developed 100 regression trees using the training set. Within this process, subsets of the training data were repeatedly sampled with replacement to construct the trees.

OC	St	Sk	Ct	Att	FRT(10s)	% H	FA	% HA
1	34	5	1	1	3	0	0	0
2	34	5	1	2	3	60	1	100
1	56	5	0	1	2	0	2	0

Table 1: A subset of data with OC and performance details.

Helpful Definitions

Out-of-bag Error [4]: A subset of training data is left out when building each tree, thereby leaving a portion of data "out of the bag." After building a tree, the out of bag subset is applied to the tree and arrives at a prediction. The root mean squared error (RMSE) of prediction for all out-of-bag cases becomes known as the out-of-bag error.

Feature Importance: When assessing the importance of a feature, *m*, values of *m* are randomly permuted in out-of-bag cases. A secondary measure of out-of-bag error is then calculated based on the permuted data. The difference between this secondary out-of-bag error and the original out-of-bag error is regarded as the importance of feature *m*. The larger the difference in error, the more important the feature is to the model. Negative values suggest a feature that is useless or even harmful in prediction.

As RF progresses through the process of building decision trees, subsets of features are chosen randomly to establish node splits. The number of features, *n*, in this subset can be limited to enhance predictive accuracy. For our current work, we explored a wide range of student performance details (or model features), ultimately retaining the *n* with minimum out-of-bag error (see side panel) when applying RF to the test set.

We ran and tested RF for each OC, using segments of the dataset. For comparison, we also constructed a Traditional Model (TM), or a single model encompassing all OCs. Within the TM, it is possible to consider or ignore OC as a feature, resulting in two types of TMs: TM without OC (TMNOC) and TM with OC (TMOC). The code used for this process is available at [7].

RF was used to examine the predictive accuracy of the TMNOC, the TMOC, and the OCM, and to quantify feature importance (see side panel) within each model. Five-fold cross validation was used to assess predictive accuracy. Predictive accuracy is represented here by the root mean squared error (RMSE) of test sets.

Results

Figure 1 shows the RMSE of model predictions for next problem correctness at various OCs. Data points at the 5th OC represent error in predicting correctness on the 6th OC. Feature importance was generated while running RF for each feature within each fold, and ultimately averaged across folds. Table 2 presents relative feature importance within the OCM.

Discussion

The OCM distinctly outperforms both TMs (except when OC = 11). This is not necessarily surprising. The OCM builds a separate model for each OC. In general, it has many more parameters (i.e., degrees of freedom) than TMs. While this observation is somewhat obvious, it suggests that OC is a critical feature to consider when modeling student performance (RQ1). We also observed a trend in the difference of prediction accuracy between models. The OCM is less accurate in later OCs. At the 11^{th} OC, TMs begin to outperform the OCM within our dataset. We suspect that this is caused

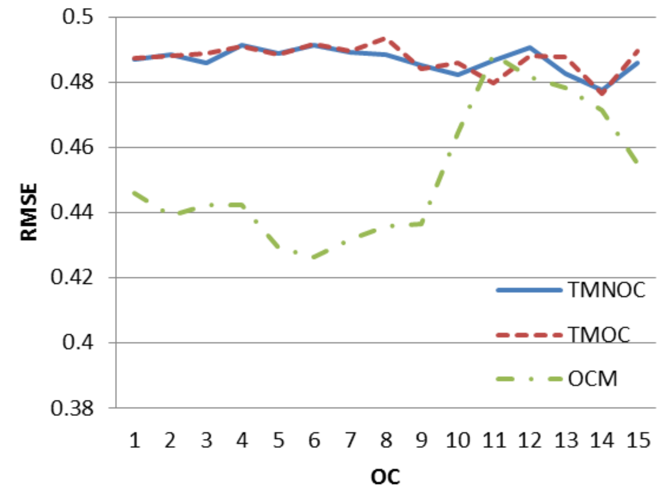

Figure 1: Prediction accuracy of three models. The point at the i^{th} OC shows the prediction accuracy of i^{th} +1 correctness.

OC	Sk	Ct	Att	FRT (10s)	% H	FA	HA
1	6.31	0.85	0.80	1.16	0.73	0.69	0.00
3	7.12	1.10	0.23	1.36	0.71	0.64	1.86
5	6.06	0.69	0.48	0.91	0.39	0.66	0.42
7	2.87	0.57	-0.05	-0.17	0.33	0.19	0.46
9	3.25	0.42	0.17	0.17	0.18	0.11	0.23

Table 2: Feature importance within the OCM at different OCs.

Helpful Definitions

Correct (Ct): A binary measure of accuracy on first attempt.

Attempts (Att): The number of attempts made before arriving at a correct solution.

First Action (FA): The first thing a student does within a skill opportunity. Students may attempt to solve (0), request a hint (1), or request a scaffold (2).

First Response Time (FRT)(10s): The time between opening a skill opportunity and making a first action. This measure was groomed to remove outliers larger than 400ms (less than 1% data loss) and to simplify the time structure to 10-second increments.

Percent of Hints Used (%H): The percentage of available hints requested by a student within a skill opportunity.

Historical Accuracy (HA): Generated to compile a student's percentage of correctness across prior OCs within a skill.

by the reduction of data points as OC increases. As OC increases, data points grow scarcer as students either master the skill or drop out. Therefore, at later OCs, the OCM has fewer cases on which to train and test.

Findings also suggest that feature importance varies as OC changes, justifying use of the OCM when considering performance details in student modeling (RQ2). Results presented in Table 2 revealed that feature importance differed considerably with increases in OC. When considering an OC of 3, aside from the importance of the skill itself, the most relevant features within the model were first response time and historical accuracy. However, when considering an OC of 7, first response time was no longer important and may have actually hindered prediction accuracy. Examining feature variation in this way allows us to discriminate important factors across different phases of learning. For example, the skill itself is more important with fewer OCs, but with higher OCs, importance shifts to other features. This may suggest that at later modeling stages, researchers should focus less on the refinement of skill content and more on factors pertaining to individual students. While it is difficult to draw any specific conclusions here based on these results, it is worth future investigation.

Contribution

The present work revealed that OC is an important factor for the community to consider when modeling student performance. Further, our finding that the importance of features varies across OCs establishes a call for further examination of the significance of student performance details at different learning phases.

Acknowledgements

We acknowledge funding from the NSF (1316736, 1252297, 1109483, 1031398, 0742503), ONR's "STEM Grand Challenges," and the IES (R305A120125, R305C100024).

References

1. Duong, H.D., Zhu, L., Wang, Y. and Heffernan, N. 2013. A prediction model uses the sequence of attempts and hints to better predict knowledge: better to attempt the problem first, rather than ask for a hint. In *Proc of the 6th Int Conf on EDM*, 316-317.
2. González-Brenes, J.P., Huang, Y. and Brusilovsky, P. 2014. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proc of the 7th Int Conf on EDM*, 84-91.
3. Heffernan, N. and Heffernan, C. 2014. The ASSISTments ecosystem: building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *Int J of AIED*. 24, 4: 470-497.
4. MathWorks. 2015. TreeBagger. Retrieved 10/10/15 http://www.mathworks.com/help/stats/treebagger.html
5. Pardos, Z. A. and Heffernan, N. 2010. Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *J of Machine Learning Research W & CP.*
6. Leo Breiman and Adele Cutler. Random Forests. Retrieved 10/10/15 https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.html.
7. Wang, Y. 2015. Data and code for "The Opportunity Count Model: A flexible approach to modeling student performance." http://tiny.cc/LAS2016-OCM