# An Analysis of Response Time Data for Improving Student Performance Prediction

XIAOLU. XIONG, ZACHARY. A. PARDOS AND NEIL. T. HEFFERNAN
Worcester Polytechnic Institute, USA

This paper describes a series of experiment observations and analysis on the response time data in the ASSISTments dataset. The response time data was extracted from different aspects of the dataset, and applied to machine learning algorithms and statistical analysis to investigate the utility of students' response time in performance prediction. Two experiments were conducted. The first one was to gauge whether response time was useful in prediction and to identify how best to use response time. This experiment was run across the entire dataset. The second experiment focused on the practical task of predicting the student's "next" response based on their previous responses. We found that response time provided a small but significant improvement in experiment 1. However, in experiment 2 the results were not significant. We provided a case study of response time distribution graphs for further investigating of our results.

Key Words and Phrases: ASSISTments, feature selection, response time, Stepwise regression, Random forests

## 1. INTRODUCTION

The 2010 Knowledge Discovery and Data Mining Cup competition involved predicting student responses in an Intelligent Tutoring System (ITS) called the Cognitive Tutor. For this competition, information about how long it took a student to answer a question was provided in the training data but was withheld from the test data [Pardos and Heffernan 2010]. Competition organizers cited response time as being too predictive of the target value which was if a student answered the question correctly or incorrectly. Given this suggestion about the importance of response time in the Cognitive Tutor, we studied what importance response time had in a different math ITS called the ASSISTments Platform.

We conducted two experiments to analyze the role of response time in predicting performance. The first experiment separated the dataset into testing and training folds at the action level allowed for the response time of the action being predicted to be known. The second experiment focused on predicting student performance without knowing the response time of the action being predicted [Shih, Koedinger and Scheines 2008]. For this experiment we allowed our models to know the response time and correctness of the first two answers of each student for the top problem sets in ASSISTments [Hershkovitz, Nachmias 2009]. The task was then to predict the correctness of the third response based on information about the first two. We found less evidence of the importance of time in this task and so a case study was also conducted and presented in this paper as a means to further investigate how best to leverage information in student response time to achieve better student modeling and prediction.

## 2. DATASET

The ASSISTments dataset came from Worcester Polytechnic Institute's ASSISTments Platform. It is comprised of student use of the system during the 2009-2010 school year. It contains 1 million rows and was made available for the 2011 Knowledge Discovery in Educational Data workshop. Each row in the dataset corresponds to a student answer which contains 19 columns; it records student's answer correctness, response time,

Authors' addresses: Xiaolu Xiong, Zachary A. Pardos, Neil T. Heffernan, Department of Computer Science, Worcester Polytechnic Institute, 100 Institute Rd., Worcester, MA 01609, USA. E-mail: xxiong@wpi.edu, zpardos@wpi.edu, nth@wpi.edu.

problem information and several other metadata. The student's answer correctness can only be 1 for correct or 0 for incorrect; the response time is a number in milliseconds which shows how much time the student spent on the first attempt. In ASSISTments, there are "Main" and "Scaffolding" types of problems. "Scaffolding" problems are work steps of "Main" problems; students will answer "Scaffolding" problems when they answered "Main" problems incorrectly, or they may choose to see the work steps, in which case the answer of "Main" problem will be marked as incorrect. There are two primary problem set types: "LinearSection" where there is a fixed order of problems in the set and all problems must be completed, and "MasterySection" where problems are given in a random order and students finish the problem set when they have answered three correct in a row. Note that we filtered out some rows which we considered them as unrealistic or gaming response time data [Baker, Corbett and Koedinger 2004; Baker, Corbett, Koedinger and Roll 2006], in the end of this process, 537586 rows of data were left in the dataset

## 3. EXPERIMENT 1: PREDICTING ACTION LEVEL PERFORMANCE WITH AND WITHOUT RESPONSE TIME

We defined a task to investigate if the response time data can help to improve the prediction results. This task is to predict if a student answered a given problem correctly or incorrectly. The Random forests [Breiman 2001] algorithm was used to build the model. An important distinction about this experiment from others is that cross-validation was done at the action level. Meaning, the rows in the dataset, each corresponding to a single student action, were randomly split into bins. Every row being trained and tested on included the millisecond response time the student took to answer the question. The purpose of this experiment was to give us an idea for if the response time feature was helpful at all in performance prediction.

For this task, two feature sets with rich data were created. The baseline feature set took every column in the ASSISTments dataset as a feature except the "attempt_count" column since any value greater than one for attempt_count means that the student answered the question wrong on first attempt (this completely determines the target in this case). For the other feature set, we added additional features with Z-scored response time and data point features. The Z-scored response times were calculated by using data selected by a particular feature. The reason for this procedure was to capture the response time of each action relative to some collection of other response times. One hypothesis is that if the student is responding slower than he or she usually does, this will result in an incorrect response [Beck 2005]. This is different than if he or she is responding slower than all the other students or slower than the average for a given problem. By Z-scoring response time with all the features we aimed to identify where the predictive power of response time may come from and where the variance lies. In addition to Z-scores, the number of matched rows for each feature value also became a feature. Z-score and data point number features were calculated for the following columns:

- user_id: the ID of the student,
- problem_id: the ID of the particular problem the student answered,
- sequence_id: the ID of the collection of problems the student answered
- teacher_id: the ID of the student's teacher
- school_id: the ID of the student's school
- tutor_mode: this mode determines if students are told if they are right or wrong after answering and if the student is allowed to get help from the system or not. In most cases the student is allowed help and gets feedback.

- answer_type: this can be multiple choice, fill in or algebra type answer field
- general Z-scores were calculated across all the data as well.

### 3.1 RANDOM FORESTS RESULTS

The Random forests implementation in MATLAB provided us two calculating modes, classification and regression. The regression mode was chosen so the prediction results are values represent possibilities of the binary class. For this task, we set the tree number parameter to 200 [Pardos and Heffernan 2010]. The training set and testing set were randomly chosen by 5-fold cross-validation and average RMSE results were recorded. Table I shows the Random forests regression results.

Table I. RMSE results of Random forests for T1 task

| | **Feature Set** | **RMSE** |
|---|---|---|
| 1 | Original dataset without response time column | 0.4112 |
| 2 | Original dataset with response time column | 0.4038 |
| 3 | With response time Z-scores and data points feature | 0.3985 |

RMSE results showed that adding response time helped improve prediction and adding the additional Z-score and data point features further improved the prediction performance. While the gains are not large, they are statistically reliably different from one another at the $p << 0.01$ level as calculated by a paired t-test on the squared errors of each prediction. The Random forests produced feature importance scores based on the data and feature sets. The feature importance scores were calculated by using out-of-bag permuted delta variable errors. The top 5 ranked features and their scores are shown in Table II.

Table II. Top 5 Feature importance of Random forests for the dataset with and without time information

| | **Feature Set** | |
|---|---|---|
| **Rank** | **Original without response time** | **With response time Z-scores/data points** |
| 1 | Answer_type (8.06) | Sequence_id response time Z-score (7.96) |
| 2 | User_id (6.88) | Problem_id response time Z-score (7.59) |
| 3 | Problem_id (6.44) | User_id (6.93) |
| 4 | Assistment_id (5.04) | Problem_id data point number (6.37) |
| 5 | Sequence_id (4.41) | Answer_type response time Z-score (5.45) |

## 4. EXPERIMENT 2: PREDICTING THE 3RD QUESTION OF A PROBLEM SET

We defined another task to see how the response time data and related features affect the prediction results. The task is to predict the answer correctness of the third problem based on information from the first and second problems. This task was tested by Stepwise regression and Random Forests algorithm.

For this task, we chose five problem sets which have the most number of answers from each of the two problem type sets as training and testing data. Two feature sets for each problem set types were created. One is the baseline feature set which used the following columns:

1. User ID (UID)
2. Problem Set ID (PSID)
3. Correctness of the first answer (1stC)
4. Correctness of the second answer (2ndC)

The other feature set contained all features above, and added additional four columns related to response time data:

5. Response time of the first answer (1stRT)
6. Response time of the second answer (2ndRT)
7. Z-score of the first response time (1stZS)
8. Z-score of the second response time (2ndZS)

The Z-scores were calculated by using mean and standard deviation from all answers of the same problem. A key difference between this experiment and the first was that response time of the action being predicted was not used in the prediction. This is a more practical example of prediction where the student's next action needs to be predicted before the student starts the next problem (no time information is known yet about the next problem).

### 4.1 STEPWISE REGRESSION RESULTS

Stepwise regression was the first algorithm we tried. It is a good sequential feature selection technique, especially it could remove features that have been added or add features that have been removed. The regression RMSE results are shown in table III. Feature names refer to the feature numbers in the last section, similarly hereinafter. With the additional response time features, the Stepwise regression did achieve better results compare to the baseline performance as well. However, it is important to note that the Stepwise regression did not select all the features. So besides the two feature sets we tested, we also tried to force the Stepwise regression take response time and Z-score features (feature number: 5, 6, 7, 8) as initial features (feature numbers in parentheses). Final features included by Stepwise regression are shown in table IV.

Table III. RMSE results of Stepwise regression

**MasterySection Problem Sets**

| | **Feature Set** | RMSE |
|---|---|---|
| 1 | 1 2 3 4 | 0.4730 |
| 2 | 1 2 3 4 5 6 7 8 | 0.4713 |
| 3 | 1 2 3 4 (5 6 7 8) | 0.4713 |

**LinearSection Problem Sets**

| | **Feature Set** | RMSE |
|---|---|---|
| 1 | 1 2 3 4 | 0.4533 |
| 2 | 1 2 3 4 5 6 7 8 | 0.4533 |
| 3 | 1 2 3 4 (5 6 7 8) | 0.4527 |

Table IV. Final features included in Stepwise regression

**MasterySection Problem Sets**

| | **Feature Set** | **Features** |
|---|---|---|
| 1 | 1 2 3 4 | 3 4 |
| 2 | 1 2 3 4 5 6 7 8 | 3 4 6 |
| 3 | 1 2 3 4 (5 6 7 8) | 3 4 6 |

**LinearSection Problem Sets**

| | **Feature Set** | **Features** |
|---|---|---|
| 1 | 1 2 3 4 | 2 3 4 |
| 2 | 1 2 3 4 5 6 7 8 | 2 3 4 |
| 3 | 1 2 3 4 (5 6 7 8) | 2 3 4 5 6 |

### 4.2 RANDOM FORESTS RESULTS

For the Random forests model we set the tree number parameter in this task to 500. A higher tree count was used in this experiment because of the smaller dataset size than experiment 1. Similarly, 5-fold cross-validation was used to generate training and testing sets. Cross-validation was done at the student level per problem set in this experiment. Table V shows the results.

Table V. RMSE results of Random forests for Experiment 2

**MasterySection Problem Sets**

| | **Feature Set** | RMSE |
|---|---|---|
| 1 | 1 2 3 4 | 0.4704 |
| 2 | 1 2 3 4 5 6 7 8 | 0.4632 |

**LinearSection Problem Sets**

| | **Feature Set** | RMSE |
|---|---|---|
| 1 | 1 2 3 4 | 0.4537 |
| 2 | 1 2 3 4 5 6 7 8 | 0.4524 |

Similar to the results from Stepwise regression, the above tables show that adding response time and Z-score (feature number 3, 4 and 5, 6) features improved RMSE results, but still not very dramatically. Feature ranking and scores are shown in table VI.

Table VI. Feature ranking and scores in Random forests results

**MasterySection Problem Sets**

| | **Feature Set** | **Feature ranking and scores** |
|---|---|---|
| 1 | 1 2 3 4 | 1: 69.9795<br>4: 64.3492<br>3: 62.5137<br>2: 11.4880 |
| 2 | 1 2 3 4 5 6 7 8 | 6: 47.8563<br>8: 44.8881<br>1: 41.5196<br>7: 40.7067 |

**LinearSection Problem Sets**

| | **Feature Set** | **Feature ranking and scores** |
|---|---|---|
| 1 | 1 2 3 4 | 1: 47.3939<br>2: 47.0197<br>3: 22.6413<br>4: 19.1135 |
| 2 | 1 2 3 4 5 6 7 8 | 6: 54.2491<br>4: 52.2612<br>8: 51.5196<br>1: 50.7067 |

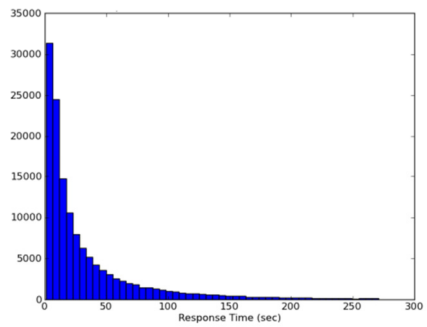
Unlike the feature selection in Stepwise regression, response time and Z-scores played much more crucial roles in the Random forests, especially the response time performance of the second problem. Also note that the correctness of answers became the least important features in the Random forests.

## 5. DISTRIBUTIONS OF RESPONSE TIME

After conducting experiment 2, we did not see significant improvement on prediction results, so we investigated some essential distributions of response time. Response time data was extracted from the filtered dataset. For each problem set type, we plotted the response time distributions for correct and incorrect answers. We found that correct and incorrect response time distributions were very similar regardless problem type sets, and the difference between correct and incorrect distributions was not enough to help the prediction. Figure 1 shows the response time distributions for correct and incorrect answers in LinearSection.

**LinearSection response time distributions**

Correct Answers




Incorrect Answers

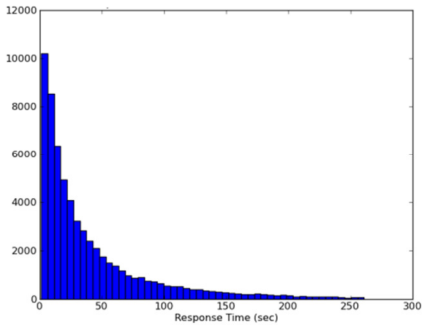


Fig.1. Response time distributions for correct and incorrect answers in LinearSection

We also studied response time distribution plots for five individual problems and five students; the five problem sets have the most number of answers and the five students answered the most number of problems. Although distributions followed a similar pattern, they still did not present a clear trend to fit in a standard curve function.

## 6. CONCLUSION

We have established that adding response time and related features lead to small yet reliable improvement for student performance prediction when response time of the predicted action was known; however, when predicting the next action of a student without knowing their response time, time information on past problems did not provide a reliable increase in prediction performance. An analysis of feature importance suggested that response time relative to the problem and sequence is more predictive than relative to a student's own past response times; in addition, a case study of student and problem response time showed that correlation between response time and correctness did not present a clear enough trend to fit a standard curve function. In order to further benefit from response time data for improving prediction accuracy, further study is needed. If response time vs. correctness was examined with consideration of student skills, the curve might show some more generalizable patterns, which may be worth to exploit.

## ACKNOWLEDGEMENTS

This research was supported by the National Science foundation via grant "Graduates in K-12 Education" (GK-12) Fellowship, award number DGE0742503 and Neil Heffernan's CAREER grant. We also acknowledge the many additional funders of ASSISTments Platform found here: http://www.webcitation.org/5ym157Yfr.

## REFERENCES

BAKER, R. S., CORBETT, A. T., KOEDINGER, K. R. 2004. Detecting student misuse of intelligent tutoring systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 531–540

BAKER, R. S., CORBETT, A. T., KOEDINGER, K. R., and ROLL, I. 2006. Generalizing detection of gaming the system across a tutoring curriculum. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 402-411

BECK, J. 2004. Using response times to model student disengagement. *Proceedings of ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*. Maceio, Brazil

BECK, J. 2005. Engagement tracing: using response times to model student disengagement. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005),* 88-95.

BREIMAN, L. 2001. Random forests. *Machine Learning, 45(1): 5–32.*

HERSHKOVITZ, A., NACHMIAS, R. 2009. Consistency of Students' Pace in Online Learning. In *2nd International Conference on Educational Data Mining*, 71-80.

Pardos, Z.A., Heffernan, N. T. In Press. Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. The Journal of Machine Learning Research W & CP.

SHIH, B., KOEDINGER, K., and SCHEINES, R. 2008. A Response Time Model for Bottom-Out Hints as Worked Examples. *Proceedings of the First International Conference on Educational Data Mining*, 117-126.