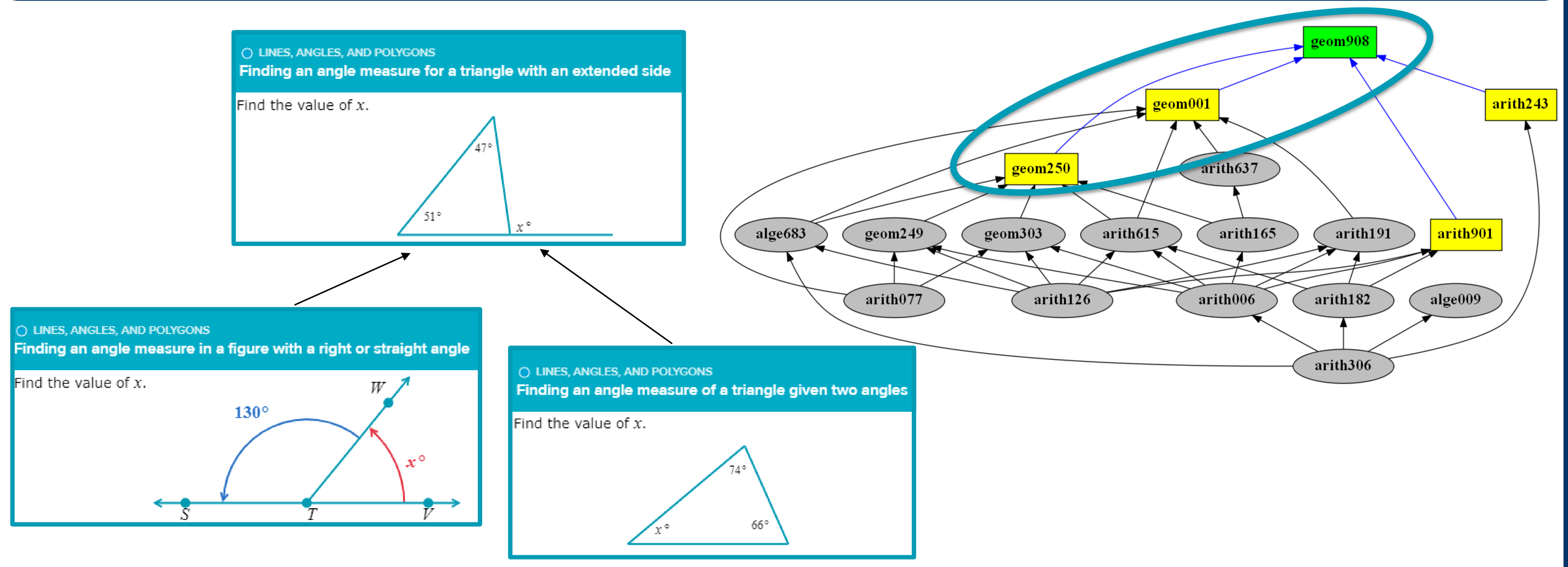
Predicting Skill Level Performance within an Intelligent Tutoring System: An Exploratory Analysis

Christopher G. Lechuga

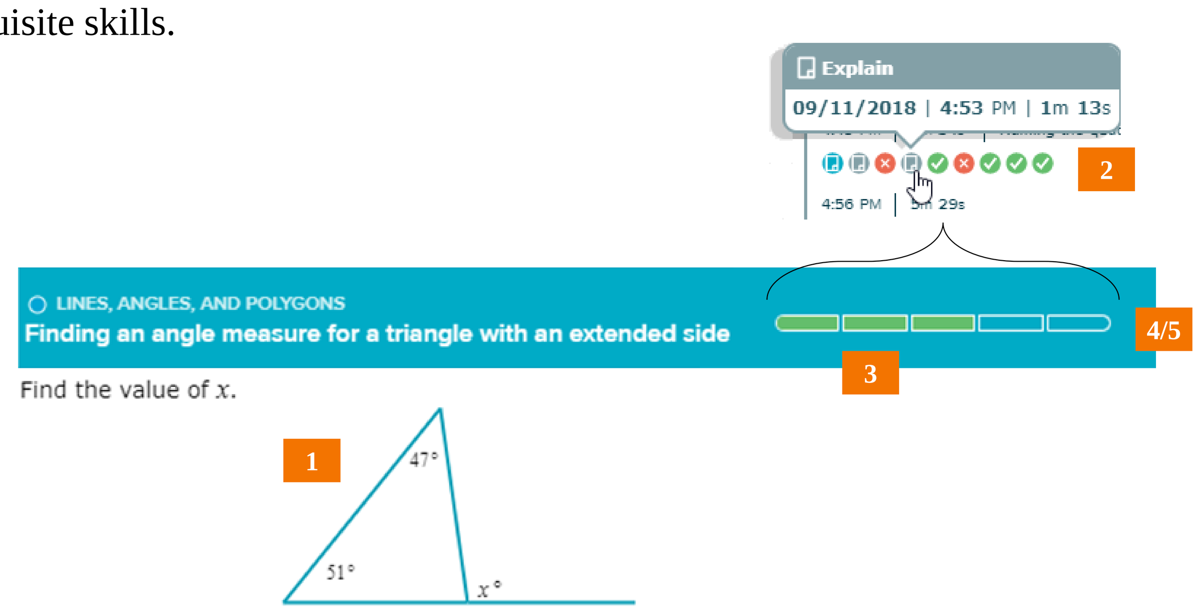
Does student performance on prerequisite skills necessarily predict performance on postrequisite skills?



Background

Over the past several decades, educational software known as intelligent tutoring systems (ITS) have attempted to simulate the role of a human tutor with the goal of helping students on a large scale. As common with face-to-face tutoring, many ITS base their pedagogical design on skill building through scaffolding, practice, and mastery. ITS also support educators with dashboards that provide insights on student progress. These dashboards are a resource, which often can alert educators of at-risk learners. However, at the skill level, reporting is captured only after students have attempted to learn the skill in the system.

In this exploratory study, we use data from the web-based system ALEKS to fit a model that predicts when a student will struggle to master a skill before she is presented with the skill. We leverage the ITS's knowledge map of pre- and postrequisite relationships among skills, as well as its domain model[4], which requires students to master prerequisite skills before moving on to postrequisite skills.



Terminology[1]	Descriptions
1. Item	A problem type that covers a discrete unit of an academic course. Each item is comprised of algorithmically generated instances on the same skill on which the student displays mastery through practice.
2. Learning Sequence	A sequence of correct and wrong responses and viewed explanations on a particular item.
3. Score	The learning sequence score that updates after each correct or wrong event on a particular item. Viewed explanations have no effect on the score.
4. Success	A learning sequence that ends by the student achieving a sore of 5 on a particular item.
5. Fail	A learning sequence that ends in 5 wrong responses in a row on a particular item.

Research Questions

- **RQ1**: Can we develop a model to effectively identify when a student will struggle on a postrequisite skill.
- **RQ2:** What features are most important in predicting success or fail on a postrequisite learning sequence?

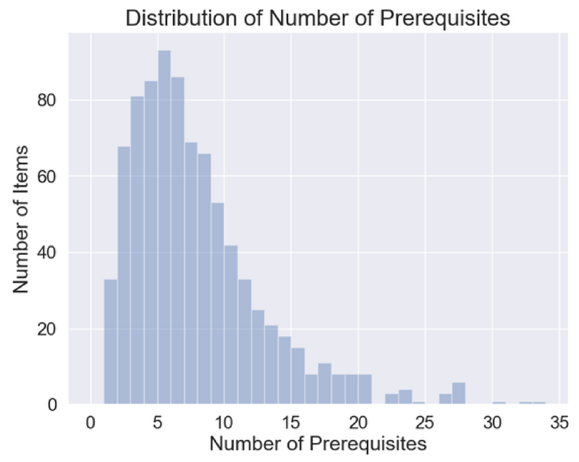
Data and Method

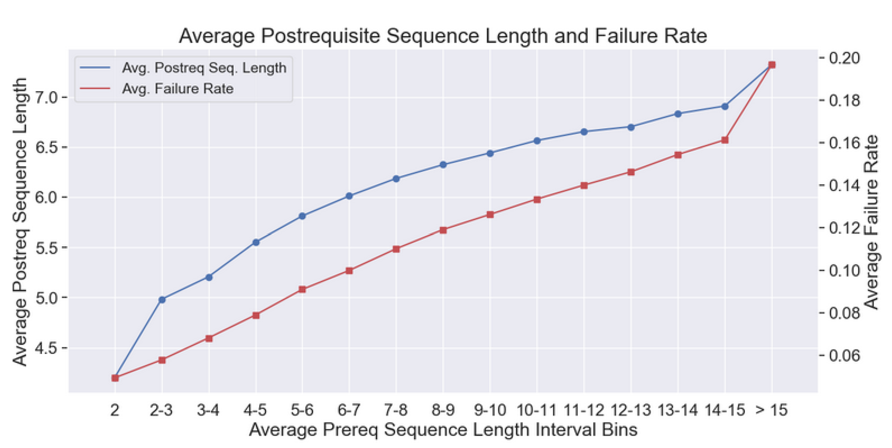
- Learning sequences were gathered on both pre- and postrequisite items from the ALEKS course product Middle School Math Course 1 from 2017 to 2019.

Row taken from original dataset.

Student ID	Initial K	Current K	Item ID	Postreq seq	Start time postreq	End time postreq	Prereq data
9578a4d1-cc9e-4623-9215-9d36482c8762	113	429	geom908	LEWEWEWEWF	2017-12-14 13:47:13	2017-12-14 13:53:02	geom001|LWWLWWLCS|2017-09-08 10:54:56|2017-09-08 13:44:47 geom250|LCWWLCS|2017-09-08 13:38:57|2017-09-08 13:41:57

- Dataset consists of 235,772 students and 11,313,995 rows of data.
- There were 851 items represented in the dataset each having at least one direct prerequisite for which the student worked on.
- Review[2] and various descriptive statistics were computed/plotted to help inform features.





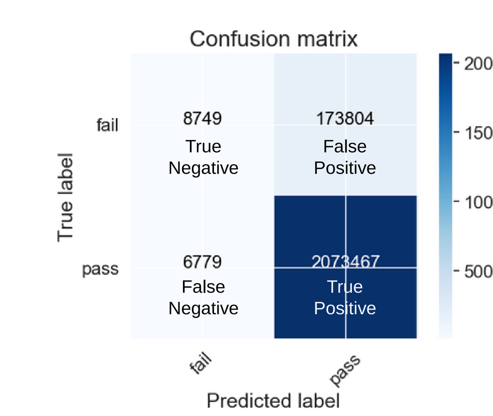
- Our predictive model employed machine learning techniques seeking to predict when a student would produce a failed learning sequence.
- Descriptive statistics and previous work suggested the item and sequence length would be important features to capture.[3]
- In all, 15 features were engineered from the original dataset. One of these is the postrequisite item, which resulted in a feature matrix with shape (11313995, 865).
- The feature matrix was separated into an 80-20 split of training and test sets.
- The training set was trained on a 5-fold cross validation logistic regression using the scikit-learn LogisticRegressionCV classifier.

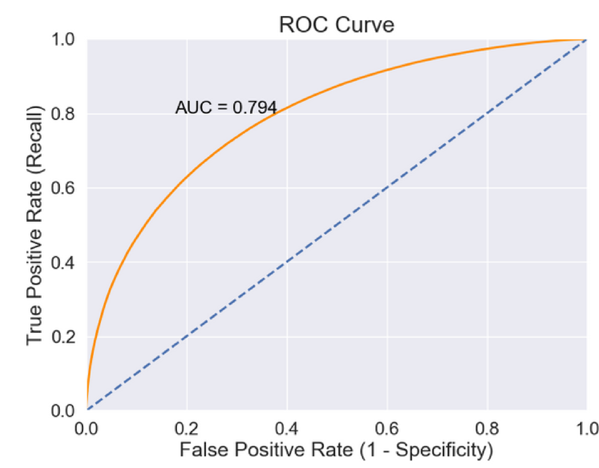
Features

Postrequisite Item	Each item represents a binary feature column (851 total) in the feature matrix.
Initial knowledge	Student initial knowledge as determined by the ALEKS initial assessment.
Current knowledge	Student current knowledge at the time of postrequisite learning sequence.
Pre/post elapse time	Time between end of final prerequisite mastered and start of postrequisite.
Max elapse time	Maximum elapse time needed to complete a prerequisite among all prerequisites.
Proportion of prereqs	Proportion of prerequisites worked on before attempting postrequisite.
Number of fails	Number of fails among prerequisites.
Avg. sequence length	Average sequence length among prerequisites.
Max sequence length	Maximum sequence length among all prerequisite sequences.
Max sequence length (S)	Maximum sequence length among prerequisite sequences resulting in a success.
Total sequence length	Sequence length taken after concatenating all prerequisite sequences on a row.
Number of corrects	Number of corrects in prerequisite data on a given row.
Number of explains	Number of explains in prerequisite data on a given row.
Correct rate	Correct rate computed from prerequisite data on a given row.
Explain rate	Explain rate computed from prerequisite data on a given row.

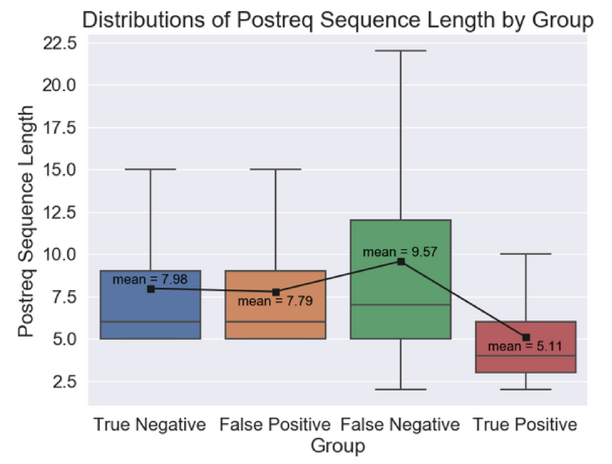
Results

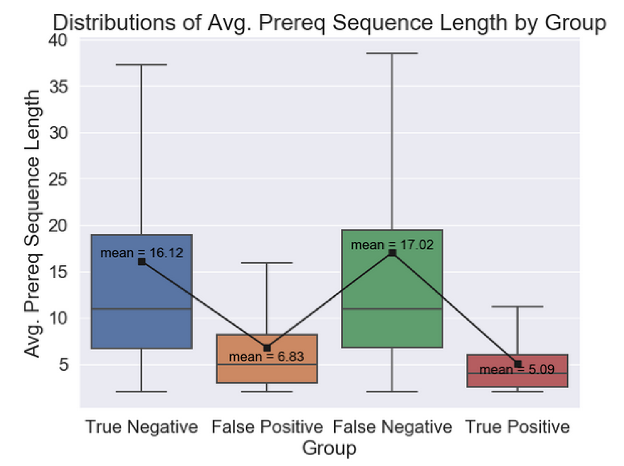
- Performance was analyzed on the held out test set consisting of 20% of the overall dataset.
- The area under the receiver operating characteristic (ROC) curve was 0.794 > 0.5, which meant that the model performed better than having no ability to distinguish pass vs. fail.
- The model showed difficulty identifying fails, as observed by the false positive group.





- We considered a more broad approach to see if our model was performing reasonably.
  a. False Positive: Why are these predicted to pass when in fact fail?
  b. False Negative: Why are these predicted to fail? Were they struggling to pass?





- On average, the model tends to predicts fail for large prerequisite sequences and pass for small prerequisite sequences for each of the four groups.
- **Re RQ1:** Focusing on the misclassified (false) groups, this trend reveals that performance on the prerequisite skill does not necessarily translate to the postrequisite skill, at least not with enough consistency for the model to predict when a struggling sequence will occur.
- **Re RQ2:** For model training efficiency, the feature data was stored as a sparse matrix, which makes it difficult to interpret the model's coefficients. We can get a sense of the relative importance of each feature by using a technique called permutation feature importance[3], which randomly shuffles the values of a feature across all data points in the test set. This is done to one feature at a time leaving all other features untouched. The idea is, feature(s) that produce the largest change in the AUC are the most important. (These are shown in bold.)

Permuted Feature	AUC (Change)	Permuted Feature	AUC (Change)
Original Model	0.794	Avg. sequence length	0.792 (–0.002)
Postrequisite Item	**0.582 (–0.212)**	Max sequence length	0.794 (~0.000)
Initial knowledge	0.789 (–0.005)	Max sequence length (S)	0.794 (~0.000)
Current knowledge	**0.733 (–0.061)**	Total sequence length	**0.772 (–0.022)**
Pre/post elapse time	0.794 (~0.000)	Number of corrects	0.789 (–0.005)
Max elapse time	0.792 (–0.002)	Number of explains	0.793 (–0.001)
Proportion of prereqs	0.794 (~0.000)	Correct rate	**0.776 (–0.018)**
Number of fails	0.794 (~0.000)	Explain rate	0.794 (~0.000)

Discussion

- Since performance on prerequisites does not predict performance on postrequisites with enough consistency, future work could investigate when it does and when it doesn't. In the case of the false positives, perhaps there is a missing prerequisite skill needed, which could inform curriculum developers to add additional skills to the knowledge map.
- A limitation of the model is that the student is not a feature. Having a student feature could improve results considering it's reasonable to suspect some students are more likely to struggle than others. While there is potential upside, this is likely difficult to implement in practice, as a model would have to update as the student continued to work in the system.

References

1. ALEKS: New ALEKS student module: Reference guide (2016), https://www.aleks.com/resources/New_Student_Module_Ref_Guide.pdf
2. Adjei, S. A., Botelho, A. F., & Heffernan, N. T. (2016, April). Predicting student performance on post-requisite skills using prerequisite skill data: an alternative method for refining prerequisite skill structures. In *Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 469-473).
3. Matayoshi, J., Uzun, H., & Cosyn, E. (2019, June). Deep (Un) Learning: Using Neural Networks to Model Retention and Forgetting in an Adaptive Learning System. In *International Conference on Artificial Intelligence in Education* (pp. 258-269). Springer, Cham.
4. Sottilare, R. A., Graesser, A. C., Hu, X., Olney, A., Nye, B., & Sinatra, A. M. (Eds.). (2016). *Design Recommendations for Intelligent Tutoring Systems: Volume 4-Domain Modeling* (Vol. 4). US Army Research Laboratory.

Acknowledgements

I would like to acknowledge and thank Jeffery Matayoshi for his collaboration on this study and Shayan Doroudi for his feedback and guidance. I also want to thank Hasan Uzun for providing a de-identified dataset required for this study.

UCI University of California, Irvine