Are You Confused? Predicting Student Confusion in Proof Helper

Colleen Lee, Ethan Fast, Jeffrey Chen

December 16, 2011

Abstract

Proof Helper addresses an important area of education, automated evaluation and assistance with proofs. However, one important question is when to provide hints to a student; ideally, we'd like to do this when the student is judged as being sufficiently confused. We ran a study using the oDesk service to collect data on how submission histories can be used as predictors for confusion. Though we were not able to reliably predict confusion given the amount of data we collected and the features we examined, we examine our results and analyze them to guess how we could make a similar future experiment more successful.

1 Introduction

Scalability is a major problem facing today's education system. To be effective, university-level education must include methods for evaluating students and assisting them as they try to learn new concepts. However, as we have seen from CS 229's overcrowded office hours, one-on-one guidance is difficult to achieve, even with traditional classroom-based education. Thus, enabling students to learn without the assistance of another human being is crucial for improving the scalability of university-level education. Furthermore, we would ultimately like to be capable of teaching anyone who is interested in learning a subject, not just those fortunate enough to be able to enroll in a certain class. The recent launch of three online courses by Stanford's computer science department is a major step in this direction, relying on online videos to teach computer science to over 160,000 students around the world, but significant problems still remain. While disseminating information is easy today, an effective educational experience must also provide students with both a way to verify that they truly understand the material and some form of guidance in solving problems. Then, in order to teach a large number of people at once, this evaluation and guidance must come through an automated tool.

Proof Helper is the work-in-progress that attempts to fulfill this need, allowing students to enter proofs in a formalized way and providing automated verification of these proofs. However, in light of the need for interaction between the student and a more "knowledgeable" entity, it will ultimately also be necessary for Proof Helper to determine when to provide a student with a hint, and what kind of hint to provide to the student. In our study, we attempt to address the first question, by formulating a method for predicting whether or not a student is confused, based on characteristics of the student's submission history.

2 Related Work

Although languages like Prolog and Metamath have allowed the programmatic representation of formal proofs, these efforts occupy a significantly different space than Proof Helper. Unlike these

languages, Proof Helper is not specific to any one domain, and its primary purpose is not only to provide a mechanism for specifying verifiable proofs, but also to educate and evaluate a student in an automated way.

3 Methodology

We collected data using oDesk, a website which allows us to hire individuals to do online tasks. For the purposes of this experiment, we added an "I am confused" button to the application, which the user was instructed to click whenever they are unsure about how to proceed with the proof.

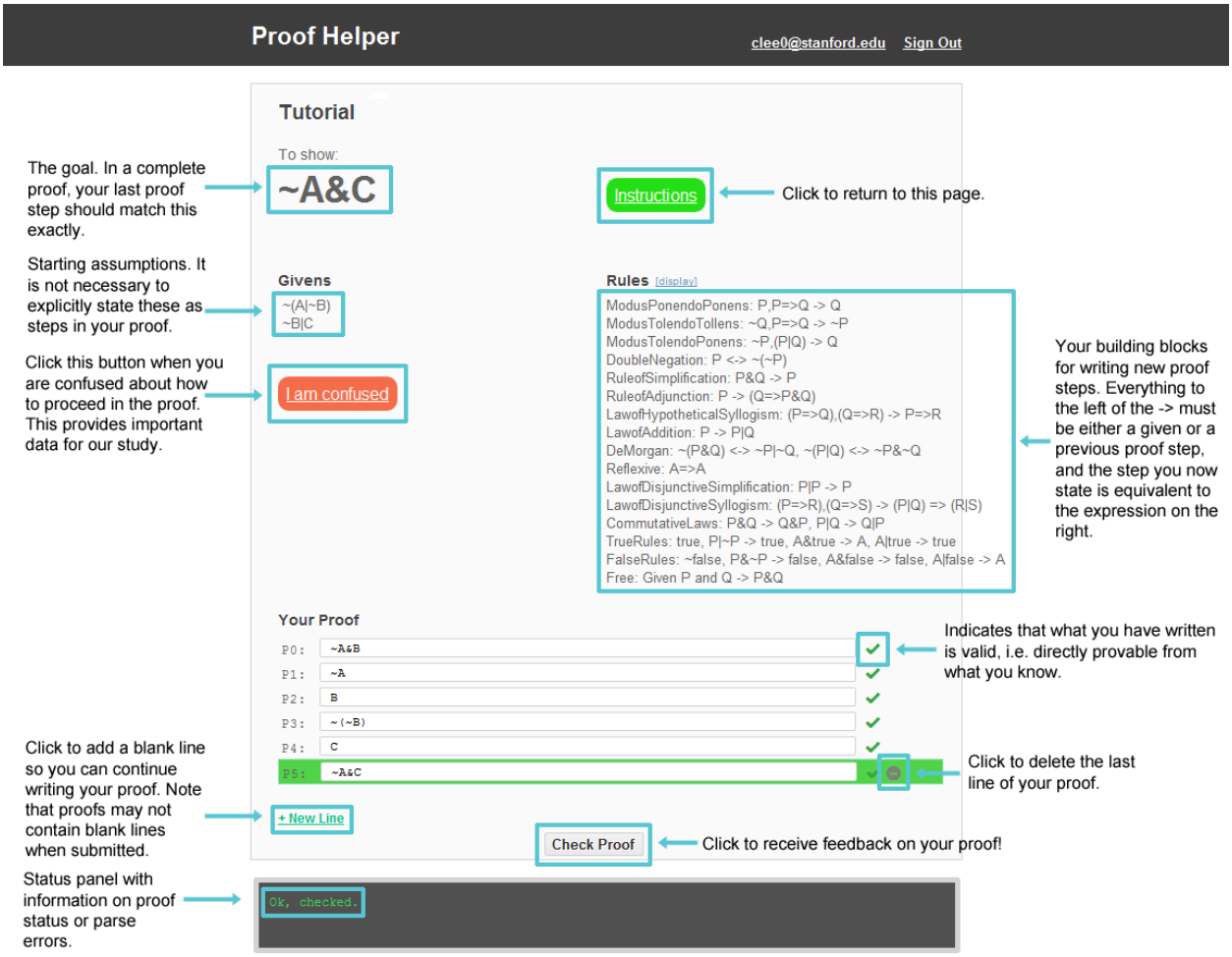

Figure 1: The tutorial screenshot for Proof Helper.

To better understand how a user's confusion level varies as he or she progresses through attempting to solve the problem, we record data whenever the user submits a proof by clicking the "Check Proof" button, or clicks the "I am confused" button. Based on these proof submissions, we can construct the following features to use for supervised learning:

1. Total number of proofs submitted
2. Number of syntactically incorrect statements

3. Number of syntactically correct statements that do not follow from current knowledge
4. Number of syntactically correct statements that do follow from current knowledge
5. Number of successful proofs (proving the goal)

These features correspond to a count of events falling within a certain timespan, separated by user and by proof. Each of these time blocks represents one data point; additionally, the target variable (confusion) is 1 if and only if the "I am confused" button is clicked at least once during this timespan. Last, to improve the quality of our data, we also:

1. Include a "filter" question to ensure that users are sufficiently qualified. We did this in the form of a very simple logic proof.
2. Emphasize that the "I am confused" button is *not* a functional hint button, i.e. clicking it will not allow the user to complete the task more quickly, and that the sole purpose of this button is to provide the necessary information for this study.
3. Randomize the order of the problems presented, so that certain problems will not systematically influence performance on other problems.

4 Conducting the Experiment

Before collecting data, we ran a short pilot study to see what roadblocks new users to Proof Helper would encounter.

4.1 Pilot Study

The five users who participated in the pilot provided useful feedback on the tool, revealing some bugs within the assignments used as well as in the parser. Another crucial lesson was that individuals who were seeing Proof Helper for the first time often had no idea how to go about writing proofs. As a result, we recorded a tutorial video walking a user through a sample proof with difficulty on par with the proofs provided, which our users said was very helpful for seeing how to begin. We also made the tutorial proof itself and an alternate solution to the tutorial proof available.

4.2 The Experiment

Though we began the pilot study on schedule, making the necessary updates to Proof Helper in response to the pilot resulted in a late start to real data collection. Though many users were able to complete the proofs quickly, often on the same day as accepting the job and within a three-hour timespan, fewer users applied to our posted job than we expected, which required us to consider shorter timespans. Ultimately, we collected data from ten users, in addition to the five who participated in the pilot.

The users who participated in our study were mostly employed in computer science or had graduated with computer science degrees, but they also included a couple of math majors and one individual in education. Most had seen propositional logic through computer science courses, and a few expressed interest in the tool because they wanted to review material from their classes.

5 Results & Error Analysis

We tried several algorithms on our data, including SVM, softmax, batch logistic regression, and stochastic logistic regression. Our most successful results came through using batch logistic regression, where the data used time blocks of length 10 minutes. We also considered using the fraction of each category of proof as opposed to the absolute counts of each kind of proof, i.e. what proportion of proofs submitted during a timespan were syntactically incorrect instead of the number of such proofs submitted, but this did not improve our results.

Error (min.)	Correct	Correct ($y = 1$)	Correct ($y = 0$)	Precision	Recall	F1 score
Train (10)	109/174 (62.6%)	14/19 (73.7%)	95/155 (61.3%)	0.189	0.737	0.301
Test (10)	24/37 (64.9%)	2/4 (50.0%)	22/33 (66.7%)	0.153	0.500	0.235
Train (2)	457/481 (95.0%)	4/21 (19.0%)	453/460 (98.5%)	0.363	0.190	0.250
Test (2)	107/110 (97.3%)	1/4 (25.0%)	106/106 (100.0%)	1.000	0.250	0.400
Train (30)	26/86 (30.2%)	19/20 (95.0%)	7/66 (10.6%)	0.244	0.950	0.388
Test (30)	4/18 (22.2%)	3/3 (100.0%)	1/15 (6.67%)	0.176	1.000	0.300

Above, we display information on the quality of results using timespans of 2, 10, and 30 minutes. The above training error is calculated by running the algorithm on a randomly selected 80% of the data set, and testing it on the remaining 20%. We note that as the size of our timespan decreases, the proportion of data points with $y = 1$ (confused) also decreases, which encourages our algorithm to simply predict that the user is not confused most of the time. On the other hand, we also note that logically, confusion should vary drastically over a 30-minute span, and hence total counts over 30-minute timespans should not be strong predictors of confusion.

Additionally, in a system that wishes to produce student confusion, we believe that false negatives are worse than false positives; if the system believes a student is confused, it can always ask the student if he or she wants a hint, while if the system does not believe a student is confused when the student is in fact completely lost, this will result in a very negative learning experience for the student.

One significant source of error is the small quantity of data, and in particular the lack of data points indicating confusion. Aside from decreasing the accuracy of our learning algorithm, this meant that we also had to increase the proportion of our data reserved for testing when calculating test error, or we would run the risk of having *no* points indicating confusion in the test set. As is, we can see that there are still dangerously few points indicating confusion in the test data set, which leads to high variability in test errors, since the data to be reserved for testing is selected randomly.

We also note that the data from different users varied significantly, a problem which is exacerbated by the small number of participants. For example, some users were more prone to clicking the "I am confused" button, and some users were more likely to submit rapidly over a short time interval, while others appeared to take more time to consider the problems outside of the tool.

Last but not least, visually plotting confusion versus each feature reveals no visual correlation between most of the features and confusion. Intuitively, while we might expect that a user is unlikely to both indicate confusion and complete a proof within the same timeblock, this certainly does occur, even over 2-minute timespans.

6 Conclusion & Future Work

Conducting this experiment has certainly yielded interesting results. In the process, we have discovered various factors that make it easier for new users to use Proof Helper, and learned about concerns that should be addressed for this tool in the future. Certainly, the ability to predict student confusion would still be very valuable, not only because such algorithms will allow the system to decide when to offer hints to the student, but perhaps even to decide what kinds of hints to provide, depending on which features contribute most to the estimation that the student is confused.

If we were to run this study again, there are several changes we would consider. First, a major hindrance to our analysis is the lack of data collected; specifically, the lack of data where students indicate they are confused. Aside from collecting data over a longer period of time, we could also actively and regularly poll the user for confusion, or require that the user indicate some measure of confusion with every proof submission. In hindsight, it was not clear how often the user should click the "I am confused" button, and given that the button serves no purpose in helping them complete the task at hand, it is likely users may have simply forgotten to do so.

Additionally, there exist other features that are potentially good predictors of student confusion that we were not able to consider given the format of the data we collected. For instance, proof history, where we consider dependencies from previous time frames, might correlate strongly with student confusion. If a student tends to resubmit identical proofs several times, this may indicate that the student believes the proof to be correct, and is confused because the system repeatedly rejects the proof. In a future study, we would format the data to make it more conducive to considering the sequential progress the student has made, including observing repeated steps.

Another feature to consider is the usefulness or ease of steps. Steps that are successfully proved may be en route to a valid proof, or they may simply be gropes in the dark. Using this as a data point requires designing some metric for estimating this usefulness. On the other hand, the ability to estimate the ease of steps and to use these as features can be beneficial because a user may be confused after proving many easy steps (making progress), then abruptly being confronted with a difficult step (and having that progress halted).

With the knowledge gleaned from this experiment, we believe that the above changes would greatly increase the probability of success for a future attempt to estimate student confusion using Proof Helper.

7 Acknowledgments

We would like to thank Professor Alex Aiken, who provided funding and a great deal of helpful advice and guidance for Proof Helper and this project; Daphne Koller and Andrew Ng, who provided early feedback on Proof Helper; Jeffrey Wang and David Kamm, who provided helpful discussions and advice on Proof Helper and data analysis; and CS 229, which gave us the impetus to carry out this project.