Cooperative Student Models

Joseph Beck, Mia Stern, and Beverly Park Woolf

Computer Science Department
University of Massachusetts
Amherst, MA 01003
Phone: (413) 545-0582
Fax: (413) 545-1249
email: {beck, stern, bev}@cs.umass.edu

Abstract. There are benefits for a tutoring system that can tailor its feedback and presentation to meet the needs of individual students. However, there are numerous difficulties with obtaining enough information for an accurate student model. To help remedy this, our tutor prompts the student to provide information that it cannot detect by itself. We call this a cooperative student model as the model is constructed jointly from the student’s and the tutor’s beliefs. There are potential pitfalls to this approach, including the student being either over or under confident, or simply being unable to accurately report his level of ability in the subject matter. We have constructed a framework that accounts for these problems. In addition to simply improving the tutor’s accuracy, this cooperative model provides the user with some means of taking part in deciding how future interactions with the tutor will progress. This system was given an initial and two follow up evaluations in classrooms. The data indicate that “on average” students are poor predictors, but when viewed individually students generally are useful at predicting future performance. Thus one set of model parameters to combine the tutor’s and student’s claims cannot be constructed beforehand; a separate model must be built for each student. We conclude by considering more efficient methods of obtaining confidence estimates from the student and for improving the framework itself to provide more accurate estimates.

1. Introduction

Cooperative student modeling is a move from a system centered to a user centered design. Rather than completely trusting the system to reason correctly about the student, cooperative models allow some method of combining the viewpoints of the tutor and the learner to build a more complete model. Constructing a cooperative model is more complex than constructing a model that does not solicit user input, as there must be a mechanism for combining both sources of input.

Bull and Pain [3] created a system, Mr. Collins, whose student model was constructed in collaboration with each student. The model was viewable, and the system gave a rationalization for its current rating by providing examples of the student’s behavior. If the learner disagreed with the system’s assessment he could challenge it. If the tutor did not agree with the student’s claim, it would challenge him with a question concerning the disputed topic. If the student answered the question correctly, his belief of his ability would be accepted by the system, otherwise the tutor would maintain its prior belief.

In this paper we consider an alternate approach to cooperative models than the one used in Mr. Collins. First we present a case for why cooperative models are useful. We then consider several possible situations such a model must gracefully handle, and propose a statistical framework for combining the input of the system and learner and constructing a student model. Next we present experimental results from classroom use of the system, and evaluate what impact student beliefs have on the model. Finally, we consider how to gather data about student beliefs more easily, and how to better incorporate them into updating the model.
2. Why Have Cooperative Student Models?

Traditionally intelligent tutoring systems (ITS) have tried to maintain a model of the student’s knowledge largely based on implicit actions performed by the student. Self [7] identified several problems with this approach. The main difficulty is that the system cannot necessarily deduce what set of transformations a student applied to arrive at his answer. As a possible solution to this dilemma, Self suggested structuring the system’s interface such that it is obvious what actions the student is performing at each step. Knowing specifically on which step of the problem solving process the student made his mistake is a useful tool for providing feedback [1].

In essence, rather than attempting to recognize the student’s method of solving the problem, the system simply asks that student what he is doing, and consequently determines what mistakes the student is making. However, there are other techniques for eliciting more information from the student. Cooperative modeling is similar in spirit to Self’s recommendation of asking the student rather than relying on complex techniques to reason about his actions and abilities. Also, students are likely to be truthful in describing how they are attempting to solve a problem. However, there is an additional complication in asking students about their level of knowledge: they may exaggerate (or underestimate) their ability. Additionally, it is somewhat odd to ask a novice to rate his knowledge of domain material; he is a novice precisely because he doesn’t understand the domain.

There are many potential benefits to using a student’s claims of his ability. First, moments of enlightenment (“Aha!” experiences) are difficult to recognize. Allowing the student to adjust his ability permits the system to handle such situations. An opposite case is when a student is unsure of what he is doing, but manages to guess his way through a problem. It is extremely difficult to distinguish this learner from the one who has perfect understanding, but ideally the student model should reflect this difference. Students may find it disconcerting when the system misdiagnoses their mistakes, and begins to “remediate” something they already understood. Giving students some control over the student model can prevent the system from performing such actions.

Cooperative student modeling is a student centered approach: the student perceives that he controls his learning. Four goals of human tutors include enhancing the learner’s confidence, providing a sense of challenge, allowing the learner to feel in control, and eliciting curiosity [6]. Traditionally, ITS’s have done an excellent job at providing a sense of challenge, a fair job at enhancing confidence via individual feedback and attention, but have lagged in giving the user a sense of controlling the interactions. Allowing the learner some say in what the system thinks about his abilities gives the feeling that he can have some effect on how the tutor will proceed.

3. Goals of Our Cooperative Model

Bull [3] answers the question of how to incorporate student claims into the tutor’s model of the student by allowing the system to argue with the student. Her belief is that the process of inspecting the model and understanding the system’s rationale will promote the student to reflect on his knowledge. Questioning the student safeguards against his possibly inaccurate self-perceptions.

Our system does not have an explicit goal of motivating the student to reflect on his knowledge. It is difficult to measure if such a goal has been met, and does not fit within the system’s teaching framework of demonstrating procedural skills and concepts. Therefore, our system is primarily concerned in using the student’s claims to obtain a better representation of his abilities, and providing the learner with a sense of control by allowing him to help control what the machine believes, and consequently how the tutor’s instruction will proceed.
3.1 Framework

Given that the system is concerned with using student input as an additional means of maintaining an accurate representation of his abilities (rather than attempting to promote reflection), a statistical model suffices. Specifically, a two parameter regression model can be used to predict how a student will perform in the future. The regression model is of the form:

\[
\text{future performance} = k_1A + k_2B + \text{y-intercept} \quad (1)
\]

where A is the tutor’s current estimate of the student’s ability and B is the student’s estimate of his ability. No assumptions are made about A and B being either dependent or independent. How the beliefs of the student and system are represented internally by the system is not critical, as long as they can be transformed into a number. This permits the system designer to use any internal student model representation that he wishes. The other parameters (\(k_1\), \(k_2\), and the y-intercept) are determined via standard statistical techniques for finding a regression line. Briefly, this method produces an equation that combines the student’s and system’s claims about the student's ability, and returns an estimate that on average is more accurate than either of them alone.

There has been prior work using regression models to predict a student’s ability. The SMART system [8] used the number of hints required by a student to update the system’s estimate of his proficiency. This system did not take into account the learner’s claims of his ability, but managed to account for an impressive 54% (65% using pretest information that could be incorporated into the system) of the variance in a posttest. This evidence supports further investigation into using regression techniques to update student models.

3.2 System background

This framework is being tested in MFD (Mixed numbers, Fractions, Decimals) [2], a tutor that teaches fraction arithmetic to grade school children. The current system is somewhat different than the previous version [9] for community college students in that it performs more teaching as opposed to primarily supporting problem solving. It retains many of the same design principles of the previous version, including carefully constructing problems of the correct level of difficulty, and providing considerable support to scaffold the student’s problem solving efforts. Topics supported by the system include whole number and fraction arithmetic such as subtracting fractions and multiplying whole numbers. The system tracks a student’s ability on a variety of subskills such as finding the least common multiple, simplifying a fraction, etc. Subskills are component steps that are necessary to solve a problem, but are not tested by themselves.

3.3 Framework goals

As previously mentioned, students may not be the best source for information about their abilities. It is possible the student is overconfident and systematically overestimating his ability or he may lack confidence and therefore consistently underestimate his ability. Girls have been shown to have less confidence in their mathematics ability than boys [4]; therefore one of the goals of MFD is to increase girls’ self-confidence in mathematics. Because improving self confidence is a desired effect of the system, it is necessary to model its effects explicitly and not make any assumptions about a student’s level of confidence.

Figure 1 demonstrates possible types of student estimation. The x-axis is the student’s actual level of proficiency, and the y-axis measures the students claimed level of ability. The solid line \(y=x\) represents what the predictions would be if the student could know and report his level of ability with perfect accuracy. Given that this is an unlikely scenario, it is critical to investigate how potential student response patterns will effect the model. Currently we are using a linear regression model, so the mapping from the student’s claims to actual ability must be linear, but need not be any particular line (i.e. a regression model is constructed for each student individually).
In Figure 1a, the student is overconfident and is consistently overestimating how he will perform in the future. In Figure 1b, the student is overly pessimistic and predicting he will perform worse than he actually will. Either pattern appears plausible, and should be related to a student’s confidence in his math abilities: students lacking in confidence are more likely to underestimate their ability [5].

Fortunately, both of these cases are easily handled by a regression model. Consistent over or underestimation can be corrected by adjusting the y-intercept term (from Equation 1). If the student estimate is not off by a constant amount (as in Figure 1b), this is still accounted for in the regression equation. The term $k_2$ can be modified to change the slope of the student’s prediction line to be in agreement with the “correct” slope. The tutor cannot know ahead of time what the correct mapping from student predictions to actual ability is, and it must learn this by comparing his predictions to his future performance.

A final possibility is that the students’ claims are completely unrelated to his future performance. It is conceivable that some students will not be able to accurately judge their level of ability, or will respond in a random manner. If this is the case, the student’s prediction is not useful, and the term $k_2$ would be 0, which results in his input being ignored. These features are an extension of the work done by Bull [3], in that the system, Mr. Collins, was somewhat more fragile in regard to inaccurate student input. If a student consistently overestimated his ability, he would be challenged by only one question each time. For someone who consistently overestimates, a fair number of “false acceptances” may occur: If a student has a 1 in 3 chance of getting a question correct, then there is a 1 in 3 chance his rating will (erroneously) be accepted by the system. This difference in emphasis is largely due to a difference in target audience. Mr. Collins was tested with adult learners while MFD helps fifth graders, who are presumably more likely to try to “beat the system”.

3.4 Potential problems

This method of assigning credit to the tutor’s and student’s predictions based on their ability at predicting future performance is excellent from a standpoint of maintaining an accurate representation. However, from an affective standpoint, there are drawbacks to this approach. If students are not useful at predicting their future performance, the system ignores their input. This may be frustrating from the learner’s perspective: he is supposedly being asked for input about how he is doing and then being ignored by the system. Given that no students became frustrated over the course of the experiment this appears unlikely, but should be watched for.

Statistical regression assumes independence of observations within each group. This means that the student ratings are not correlated with each other, and the tutor’s rating are also not correlated with themselves. This assumption does not hold, but simple regression serves for a first attempt at building a model. More powerful statistical models that do not make this assumption are being considered for future versions of the system.
4. Experimental Design

As part of a test of MFD in Fall 1996, the system was modified to ask the students about their self confidence following each correctly answered question. The system was tested with 19 fifth grade children in a rural school. Students used the tutor for 6 sessions over a period of two weeks. On average, each student completed about 25 problems. The students did not work on the same problems; the system created problems to meet the student’s current level of ability [2].

When the student submits an answer, simply asking for his overall confidence that his answer is correct does not suffice. He may be unconfident because solving the problem required the use of an unfamiliar substep, but in general he could solve problems of this type. What skill should the student’s reported confidence be associated with? We are back to the problem we were trying to avoid, namely having to perform complex machine reasoning with little information with which to work. Instead, the system can periodically query the student about his beliefs of his abilities in different skills. An example of this is shown in Figure 2.

Each time the student answers a question he is quizzed about what he thinks his ability is for one topic. The topic selected is not chosen at random, but is biased towards activities the student has seen recently (or will see soon). Additional types of questionnaires asked students which topics they thought they thought needed more work on, and which topics they might have forgotten.

This was a pilot study to examine the impact of using student input as part of the tutor’s model. Therefore, the confidence measures were simply time-stamped and stored, but were not used by the system. Some of the students were puzzled by this. There were two major flaws with this design. First, these dialogs appeared every time the student solved a problem. As a result, students became frustrated at the constant interruptions. Second, the button “Need more work on this” was selected by default. This led to many students simply clicking “OK” (as they were already frustrated by this process). For the first trial any responses where the student simply selected the default answer were discarded, which understates the results as one-fifth of the inputs to the regression equation were discarded.

The second and third studies were modified to account for these problems. First, questionnaires to solicit the student’s opinion were only asked every fifth question, but such dialogs asked about multiple skills (from one to eight). This resulted in less frequent interruptions, and students seemed to enjoy this more than the initial design. Additionally, there was no default button checked on the dialogs, so students had to actively make a selection, and were not permitted to submit the questionnaire until all skills had been evaluated. These larger questionnaires had one copy of Figure 2 for each skill about which it was asking the student.

The tutor’s estimate of the student’s ability in a skill is computed by considering a history of the hints the student required to solve a problem using that skill [2]. The student’s most recent estimate of his ability is used as the confidence measure. To solve equation (1) is some metric of the student’s performance is needed. Initially the goal was to use how many hints students required to solve a problem [8]. However students tended to work out problems carefully on paper, and although they made mistakes, they did not make a sufficient number to produce much variance in the results. Therefore we considered using the time required to solve a problem, but this metric has too much variance. We decided to smooth the results, so that the tutor is trying to predict the average of the next (for our analysis) three target values. This analysis was conducted for both time to solve the problem, and the tutor’s internal estimate of the student’s ability.

Whenever a student completes a problem, the tutor stores:

- Tutor’s estimate of student’s ability (Parameter A from equation 1)
- Student’s estimate of his ability (Parameter B from equation 1)
- Type of problem (Fractions or whole number)
- Time to solve the problem

The system is attempting to predict future proficiency and time to solve a problem, therefore two equations were created: one for whole number problems and another for fractions. Students took approximately 30 seconds to solve whole number problems while fraction problems average 217 seconds to solve. Given this large difference in times, having separate predictions for each type of problem is clearly beneficial.
An interesting question is how many prediction equations to create. One possibility is to have a separate prediction equation for each topic within the system. However for each individual topic there would not be sufficient datapoints to construct a model. Additionally, it is beneficial for the system to transfer its knowledge about the student across topics. That is, if a student tends to overestimate the time he takes to solve adding whole number problems that knowledge should be used when evaluating his predictions about problems involving subtracting whole numbers. For the second and third studies we constructed only one regression equation, but used the problem type (fraction or whole number) as a predictor variable. This allows for more data to be used in constructing the model, but can only represent a constant difference in problem solving times between fractions and whole numbers. That is, a student will always take (for example) 45 seconds more on a fraction problem than one with whole numbers. This is clearly an oversimplification, but splitting the domain in this manner is a compromise between the model’s flexibility and its power. As stated previously, the model was not used by the system, the data were gathered on-line and the model constructed afterwards.

5. Results and Discussion

There were three studies conducted to determine the applicability of our regression model. The first study was a pilot, and the second two were improved based on what was learned in the pilot. The second and third studies varied in the audience using the system. Studies 1 and 2 were conducted at a school in a rural location, with primarily middle-class students, while study 3 was conducted in an urban setting with a more diverse population.

5.1 Study 1

For the first test, the problem type, and tutor’s and student’s estimates accounted for 23% of the variance in problem solving time across all students, with the type of problem accounting for the majority of this. Using only the tutor’s and student’s estimate of performance less than 4% of the variance was accounted for. However, this ignores two critical points: 1) there are certainly individual differences in the time students require to solve a problem, even if the tutor believes they are equally skilled (time to solve problem is not used as a criteria for updating student performance in the current version of MFD), and 2) if some students are underestimating their performance and others are overestimating it is possible these effects will cancel out, resulting in
little net effect from student predictions. This strongly suggests that trying to determine in advance how to incorporate a student’s input is not productive.

The next analysis examined individual students. It was not possible to partition the predictions by problem type (whole number or fraction), as this would have resulted in too few datapoints for some students which would have artificially inflated the variance accounted for. Instead, the problem type was ignored in the analysis for the first experiment. Given the large disparity in times for the two problem types, and the large variance accounted for by this factor, this is a large handicap imposed on the model. In spite of this, 32% of the variance was accounted for using just the tutor’s and student’s estimate of ability. This improvement in prediction, coupled with ignoring a large source of variance lends support to considering students individually rather than making broad statements about how to generally incorporate students’ input.

More variance has been accounted for, but is this due to the tutor’s or the student’s performance estimate? Regression has a statistic called Beta that can be used to determine how much influence (and the direction of the influence) each predictor has on the target variable. The magnitude of Beta reflects how much influence the variable has, and the sign of Beta indicates its direction. $R^2$ denotes how much variance is accounted for by the predictor variables, with 1.00 being perfect and 0.00 meaning the predictor variables are useless.

<table>
<thead>
<tr>
<th></th>
<th>Tutor</th>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>-1.23</td>
<td>-0.53</td>
</tr>
<tr>
<td>Maximum</td>
<td>-0.05</td>
<td>0.45</td>
</tr>
<tr>
<td>Mean</td>
<td>-0.42</td>
<td>0.08</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.34</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 1: Beta values for Tutor and Student predictions

Based on these data, it appears that the tutor has a higher absolute Beta (the magnitude of Beta indicates its strength of influence, the sign indicates the direction), and thus contributes more information than the student’s prediction. In fact, on average, the student predictions were nearly worthless (average Beta of 0.08). However, the tutor Betas are all less than 0 while the student Betas are distributed around 0, so the positive values are canceling out the negative ones. The average of the absolute values of the Betas for the student predictions is 0.32, which is almost as useful as the tutor’s predictions. A personalized model can learn how useful a student’s prediction really is (its Beta value), rather than being forced to guess ahead of time, and be forced to select the mean (which is a non-impressive 0.08).

Given that on average student predictions are worthless (averaged across the entire student population), there is no way to determine a priori how to combine the student claims with the tutor’s model. Instead, the claims must be handled on a case by case basis. This is to be expected given the large differences in student confidence and its relation to over and underestimating performance, and in student’s ability to accurately measure their own knowledge.

5.2 Studies 2 and 3

The second study used the same target population as the first, but used the improved version of the system, and was used with more students (N=50). Study 3 was done in an urban area (N=16). A similar methodology was conducted, except that both a regression model for time to solve the problem and the future estimated proficiency were constructed. Table 2 details the average Beta value (for the cases where the aggregate values are being computed), and the average absolute Beta value (for the second and forth cases, where the parameters are being estimated individually).

As can be seen from Table 2, the students’ predictions are useless when considered in the aggregate. The best estimate of their worth is the average of -0.12 and 0.13, which is 0.005, or 0.07.
Table 2: Prediction Values for time to solve a problem

Taking the weighted average. In this case the sign of the Beta value is important, as when constructing a model one must know which direction the variable exerts its influence. If the students are examined individually, the direction Beta exerts an influence can be computed, and a more meaningful value can be computed (0.45, or 0.54 with weighted average). Table 3 examines the accuracy of predicting future values of the proficiency variable. These results are not as clear cut as those in Table 2, but again the value of the student’s predictions (and the overall prediction accuracy) rise noticeably.

Table 3: Prediction Values for problem solving accuracy

It is interesting to note that the urban students were not as familiar initially with the content of MFD. It would seem to follow that their estimates of their ability would not be as accurate as the rural students, who had some familiarity with the material. The fact that this is not the case indicates the relationship may be more complex than initially thought, and a larger study should be conducted to determine what is happening.

Finally, a negative Beta value indicates that the student’s prediction should be treated as the opposite of what the student is claiming. That is, if the student claims he is performing quite well, and has a Beta of -0.5, the system should guess that the student is actually doing poorly. That this occurs is very counterintuitive, but seems to consistent across all studies as a fair percentage of students actually predict the opposite of what they should. Again, this is an area that should be further investigated.

6. Conclusions and Future Work

If a system considers how to use a student’s opinion, then there are clear benefits from asking him about his level of knowledge. In MFD students are almost as useful as the system at predicting their future performance. However, system designers cannot decide ahead of time how to incorporate the student’s input into the model. Rather, the usefulness will vary on a student by student basis. Fortunately, statistical techniques can be used to account for many of the potential difficulties that may arise.

Using a more complete set of target variables (i.e. student performance metrics) is also desirable. Generally there is tradeoff between speed and accuracy during problem solving, but most tutoring systems look at only one or the other, and as a result much information is lost. Ideally, both of these factors should used along with the student’s estimate in updating the internal representation of the student’s ability. We are considering how these factors should be combined when evaluating students in our system. Our goal is not to pressure students into solving problems quickly, but to recognize that solving a problem in 20 seconds indicates a higher degree of knowledge than solving it in 70 seconds. Obviously how this is achieved will vary on a system by system basis, as educational goals vary widely.

We are still looking for less obtrusive methods of gathering information from the students. An obvious area of improvement is being more selective about which skills the system asks the student. Currently a set of heuristics are used that guess what skills a student will be using soon.
This can be improved upon by being made more accurate, and determining what skills will actually provide useful information to the system.

The current implementation does a better job of acquiring information from students than the initial version. However, there is still much that can be done. Currently there is no way for students to volunteer information, so the “Aha!” experience mentioned earlier can only be detected in the system happens to ask about that particular skill just after the revelation occurs. We are searching for mechanisms that allows students to easily make claims, but does not interfere with the tutor.

One of the research goals of the system is to improve girls self concept in mathematics. While using the system it was found that girls had lower self confidence than boys on their ability to solve whole number problems, but equal confidence on fraction problems. Since whole numbers are introduced first, and fractions appear later, this is preliminary evidence that girls’ confidence is increasing as they use the system. This lends support to our goal of increasing girls self confidence, but presents a difficulty: students may not consistently over or underestimate; their model parameters may change over time. Therefore an additional goal is to examine how stable the parameters are, and how to value student estimates when constructing the regression equation. If the student’s self-concept is changing in a regular manner, more recent estimates should be given higher weight in determining how to use the student’s current claim of confidence. If overall the student his underestimated his performance, but recently has been quite accurate, the system may do better to consider his future estimates be fairly accurate rather than averaging in the earlier data where the student was underestimating.

Also, adding data about a student’s confidence in his math ability should result in the model attaining a good fit more quickly. Such data can quickly be gathered from a survey, or as part of a brief questionnaire administered by the tutor. Students with low confidence in their math abilities are likely to underestimate their performance. If the tutor knew this from the start, it could begin by biasing the student’s estimates upwards when it used them to predict future performance. This would enable it to more quickly adapt its instruction to the student. We are currently investigating whether this is a useful feature to add the system.

Finally, the regression model used assumes that each observation of each variable is independent. That is, student and tutor confidence measures do not correlate with themselves. This clearly does not hold, and we are considering more powerful statistical methods rather than simple regression for the next implementation.

In conclusion, using student data as an additional predictor has merit as students are able to provide useful information about their state of knowledge. They may not give an accurate portrayal, but this can be accounted for via basic statistical techniques. Incorporating student input not only makes the student model more accurate, it also allows the learner to feel that he is taking part in grading himself, and has some say in how he will progress through the curriculum.

References


