Analyzing the Data Collected by Programming Tutors that Provide Post-Practice Reflection

Amruth Kumar¹, Peter Rutigliano²

¹Ramapo College of New Jersey, Mahwah, NJ ²New York University, NY, NY
amruth@ramapo.edu

Abstract

We used multivariate analysis techniques to analyze the data collected by programming tutors in order to find the relationships, if any, between the correctness of a student’s answer, and several independent variables including the number of steps in the correct answer, the time taken by the student to solve each problem, the number of problems solved by the student during the tutoring session, the hardness of the concept underlying each problem, and the number of attempts needed by the student to correctly identify the concept underlying each problem during post-practice reflection. After eliminating outliers determined through combined variance of all the independent variables, we found that correctly solving the problem is closely associated with the number of attempts needed by the learner to identify the underlying concept during post-practice reflection. Therefore, the reflection questions as presented by our tutors correctly assess understanding of the concept underlying each problem.

1. Introduction

We have been developing software tutors, called problets to help students learn C/C++/Java/C# programming language concepts by solving problems. Three such problets cover selection statements (if, if-else), while loops and for loops. These problets present programs to the learner, ask the learner to predict the output of the programs, grade the student’s answer, and provide demand feedback, including explanation of the step-by-step execution of the program that produces the correct output [13]. Figure 1 shows a snapshot of the selection problet with the program in the left panel and the feedback in the right panel.

Each problet generates problems as instances of parameterized problem templates. Each template is associated with one concept in the domain. Some of the concepts in the selection problet include executing a selection statement when the condition is true/false, executing nested if statements, executing if-else statements nested in cascaded/classification style, and executing a program with multiple dependent/independent selection statements. Similarly, some of the concepts covered by the while and for loop problets include nested dependent and independent loops, multiple dependent and independent loops, loops that iterate zero or one time, and loops that update the loop variables multiple times. During data analysis, we will refer to concepts as template category. The templates in each problet are sorted in increasing order of difficulty of the underlying concept. So, the larger the template category, the harder the problem.

The problets adapt the generated problems to the needs of the learner, and present problems on only the concepts that the learner has not yet mastered [14]. Typically, a problet administers a pretest to evaluate the learner’s knowledge. During the pre-test, it does not provide any feedback. After the pre-test, the problet provides tutoring practice, during which, it adaptively generates problems and provides demand feedback. Finally, the problet provides a post-test, during which it provides no feedback. All the data analyzed in this paper was collected during the tutoring practice session.

The selection, while and for loop problets present problems on predicting the output of a program. The output of a program may contain zero, one or several steps, each step corresponding to the result of executing one output statement once, e.g., if a program contains a single output statement in a loop that is iterated 5 times, the correct answer to the problem contains 5 steps. During data analysis, we will refer to the number of steps in the correct answer as actual steps.

The problets use a reified user interface [15] wherein, the learner enters the answer using direct manipulation, i.e., the learner enters the output of the program one
step at a time; and enters each step by first clicking on the line of code that produces the output. Therefore, without a deep understanding of the behavior of the program, a learner cannot correctly solve a problem. The learner may enter any number of steps before submitting her/his answer. During data analysis, we will refer to the number of steps entered by the learner as attempted steps. We will refer to the number of steps in the learner’s answer that are actually correct as student steps. Suppose the correct output of a program is:

5 printed by line 13
7 printed by line 15
9 printed by line 16

The actual steps in the correct answer are 3. Suppose the learner enters the output of the program as:

5 printed by line 13
9 printed by line 15

The attempted steps are 2, and the student steps, i.e., the number of steps correctly entered by the student is 1.

While entering the answer, the learner has the option of clicking on a button to indicate that the code does not produce any output – in this case, the attempted steps are 0. The learner also has the option of clicking on a button to indicate that she/he does not know the answer to the problem. In this case, the learner is marked as not having attempted the problem.

The problet keeps track of the time spent by the student on each problem. During data analysis, we will refer to this as time. In addition, the problet reports the ordinality of each problem (e.g., problem #5 or problem #10). During data analysis, we will refer to the ordinality as problem. Finally, the problet reports the number of the template on which each problem is based. We will refer to this as the template number. However, the utility of the template number as a data item is subsumed by that of the template category, described earlier, i.e., our interest lies in tracking the concept on which each problem is based, which is provided by template category. Each problet contains multiple problem templates per concept, and the differences among the different templates (template numbers) is negligible.

2. Reflection

Reflection, i.e., learners reflecting on what they have done during learning and what they have learned, is conducive to learning [4, 7]. Reflection encourages learners to analyze their performance, compare and contrast their actions against those of others, in particular experts, and generalize from the actions they used in similar situations [4]. Therefore, we have been studying the effect of reflection in our problets, and the study is ongoing.

Intelligent tutors have promoted reflection of different aspects of learning. The work of [3] focused on reflection on the learners’ own thought processes and learning and was done in the context of tutors for learning the skills of tennis, problem solving in algebra and geometry, writing, and reading. The Sherlock II tutor [11, 12, 16] encouraged learners to reflect on their approach to solving a problem after they had solved it. Our approach in problets has been to encourage the learner to reflect on the concepts underlying the problems.

Several researchers have studied providing reflection during instructional activities [2, 5, 18]. [10] studied reflection after the instructional activity, and have shown that such Post-Practice Reflection (PPR) can play a significant role in instructing students in the conceptual knowledge underlying tutoring tasks. In the same vein, in problets, we provide an exercise promoting reflection after each problem.

The learning companion Lucy provided with the PROPA ITS to teach explanatory analysis of satellite activity encourages reflection using menu-driven dialog [8]. Several researchers have used natural language dialogs to promote reflection (e.g., [10]).

In problets, we introduced reflection in the form of a multiple-choice question presented after each problem. The question states "This problem illustrates a concept that I picked based on your learning needs. Identify the concept." The learner is provided five choices, each of which is a different concept in the domain. The learner must select the most appropriate concept on which the problem might be based, and cannot go on to the next problem until (s)he selects the most appropriate concept. The problet records the number of unique concepts selected by the learner up to and including the most appropriate one. During data analysis, we will refer to this number as reflection attempts. Ideally, the learner should pick the correct concept on the first attempt. If the learner takes three or more attempts, statistically, the learner is guessing the concept, without reflecting on the problem (s)he has just solved. See Figure 2 for a snapshot of the reflection question presented after a problem in a problet on selection statements.

3. Data Collection and Analysis
As mentioned earlier, we have been studying the effect of post-practice reflection in problets. We conducted controlled tests using the while and for loop tutors in spring 2006 and the selection tutor in fall 2006. For our current analysis, we considered only the data from the experimental groups, all the students in which answered a reflective exercise after each problem. Moreover, we considered only the data from the tutoring practice session, and not from the pre-test or post-test stages during which the students neither received any feedback, nor any reflective exercises.

We collected the data from Computer Science I students from different institutions who used problets. The number of students, the number of institutions from which they were drawn and the number of problems solved by them during the tutoring practice are listed in Table 1 for the three tutors.

<table>
<thead>
<tr>
<th></th>
<th>Students</th>
<th>Inst.</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>While</td>
<td>39</td>
<td>3</td>
<td>500</td>
</tr>
<tr>
<td>For</td>
<td>31</td>
<td>4</td>
<td>402</td>
</tr>
<tr>
<td>selection</td>
<td>50</td>
<td>4</td>
<td>472</td>
</tr>
</tbody>
</table>

**Table 1: Data size**

To recapitulate, the items of data that we collected for each problem were:

1. The **problem** number, i.e., ordinality of the problem solved by the student during the tutoring session;
2. The **template number** used for the problem;
3. The **template category** which represents the concept underlying the problem;
4. The number of **steps attempted** by the learner when answering a problem;
5. The number of **actual steps** in the correct answer of the problem;
6. The number of **student steps** that were correct;
7. The **time** taken by the learner to solve the problem and answer the reflective exercise;
8. The number of **reflection attempts**, i.e., the number of attempts made by the student to correctly identify the concept underlying a problem during the reflection exercise that followed the problem

Our objective in using data mining techniques was to see if we could find any correlations or patterns among the above 8 items of data. In other words, we did not set out to answer specific questions as to find serendipitous answers.

We used Simca-P 10.5 to analyze the data. We performed a Projection to Latent Structures by means of Partial Least Squares Discriminant Analysis (PLS-DA) on the data. Independent variables were **problem, template category** (TmplCat in figures 3 and up), **attempted steps** (Attempted), **actual steps** (ActSteps), **student steps** (StdSteps), **time, reflection attempts** (ReflAttempts) and the specific tutor section used, i.e., selection, while or for loops (Tutor). The dependent variable was the percentage correctness of the student’s answer, calculated as the ratio of student steps / actual steps, dichotomized into two levels: correct if the ratio was 1.0 corresponding to a completely correct answer, and incorrect if the ratio was less than 1.0, corresponding to both partial and incorrect answers.

### 3.1. Statistical Techniques

It was determined we would use Partial Least Squares Regression analysis using SIMCA Pro+ 10.5 (www.umetrics.com). Partial Least Squares Regression (PLS) is a means of predicting the dependent variable from several independent variables while also describing the relationship between the variables. Unlike traditional regression, however, PLS combines features of principal component analysis which investigates coherent subsets of the data [1].

Simca offers several benefits over more traditional analysis packages. Specifically, Simca provides the ability to determine combined variable outliers. Traditional analysis can only look at individual outliers. For instance, if subject X has an extreme score on the number of reflection attempts, we would exclude this subject. However, with Simca, we can see combined outliers; subjects whose individual variables may be within acceptable range, but the variance of a number of variables combined together may not be consistent with the rest of the subject pool. A second benefit of Simca PLS is that Simca does not have the same requirements of assumptions of the data as with traditional analysis [6]. For instance, Simca PLS remains robust despite issues with multicollinearity, normality, noise and missing data. Therefore, our confidence in the final outcome is higher than with traditional analysis.

Although our current dataset does not contain a large number of variables, PLS analysis using Simca provided two advantages: determining outliers based on combined variance and a more robust statistic test of the data.

### 3.2. Initial Analysis

The variables showed a moderately strong accounting of the variance (cumulative $r^2 = 0.57$; cumulative $q^2$ (cross validated $r^2$) = 0.35). As
indicated in the scatter plot of the combined variance of the independent variables in Figure 3, there was a clear separation between the two groups – the problems solved correctly and those solved incorrectly.

In the loading plot in Figure 4, we see that the number of problems (Proble) and the number of student steps (StdSteps) differentiate the dependent variable the most; the actual number of steps (ActSteps) and the number of reflection attempts (ReflAttemp) show moderate influence and the rest of the variables show little to no influence at all (red diamonds indicate the 2 levels of the dependent variable, and black squares indicate each independent variable separately). The problem number (problem) is most associated with not correctly solving a problem (M2.DA in the figure). This can be attributed to the fact that problets present problems in increasing order of difficulty of the underlying concepts. Therefore, learners are more likely to incorrectly solve problems that appear later during the practice session. Actual steps (ActSteps) is moderately associated with incorrectly solving a problem. This seems logical – the more the number of steps in the correct answer, the harder the problem and/or the more likely that a student would make a mistake when solving the problem. Reflection attempts (ReflAttemp) is also moderately related to incorrectly solving a problem. Again, this is logical – if a student does not solve a problem correctly, (s)he may not understand the underlying concept, and cannot identify the underlying concept during the subsequent reflection exercise. Finally, student steps (StdSteps) is strongly associated with correctly completing the item (M2.DA2 in the figure). This was expected as well, since the absolute correctness of the student’s answer is directly proportional to the number of correct steps in the student’s answer.

What is most interesting about this data, however, are the factors that do not appear to relate strongly with correctly solving a problem. Specifically, attempted steps, i.e., the number of steps attempted by the learner was not related to either correctly or incorrectly solving the problems. This may be due to the fact that problets use negative grading, e.g., if a problem has 4 actual steps, and the attempted steps by the student is 6, and the first four of these 6 steps are correct, the student identified all the output of the program, but proceeded to identify two non-existent outputs. The problem allows 2 student steps in this example, penalizing the learner 2 correct steps for the 2 non-existent steps. Similarly, time was not related to either correctly or incorrectly solving the problems – under-prepared students may take a long time to solve a problem correctly or incorrectly; well-prepared students may solve a problem correctly in a short amount of time, and disengaged students may “solve” a problem incorrectly in a short amount of time. Finally, the template category and the tutor used were not associated with correctly solving the problem.

3.3. Analysis Without Outliers

The Distance to Model Plot in Figure 5 shows the combined variance of the independent variables away from the corrected model for all the data points, and thus specifies several outlier data points. After removing outliers using a cut off score of 2.5, a different clinical picture emerged. Total explained variance of independent variables on dependent variable remains high (cumulative $r^2 = 0.49$; cumulative $q^2 = 0.72$). Only the independent variable reflection attempts (ReflAttemp) shows a distinct relationship to correctly solving the problems in the loading plot in Figure 6. This suggests that the number of attempts necessary to reach the correct answer in the reflection question is related to correctly solving the problems.

The new scatter plot of reflection attempts in Figure 7 obtained by eliminating the outliers shows a clear distinction between the problems correctly solved by the learners versus the problems incorrectly solved by the learners. Furthermore, the points are clustered into five groups. When we color-code the levels of the ReflAttemp variable as in the scatter plot in Figure 8, it is obvious that these five groups correspond to the five possible values for the number of reflection attempts (1-5). Comparing Figures 7 and 8, it is clear that an increase in the number of attempts on the reflection question correctly predicts incorrectly solving the problems and vice versa.

4. Discussion

Given the close association between the number of attempts to answer the reflection question and solving the problem correctly/incorrectly, we can conclude that the reflection questions as presented by problets correctly assess understanding of the concept underlying each problem.

The problems presented by the tutors in this study are about predicting the output of programs. The reflection question is a multiple-choice question on identifying the concept underlying a problem, premium being placed on how few attempts a learner needs to identify the most appropriate concept. Given the close association between the two, we conclude that a learner who can predict the output of a program can
also identify the concept illustrated by the program. In other words, problems on predicting the output of programs are adequate for assessing whether the learner understands programming concepts. These problems could complement problems on writing programs on assessment instruments for introductory programming classes.

Researchers have studied the effect of reflection on learning by providing reflection through self-explanation (e.g., [2]), and through inspection of open student model (e.g., [9]). In contrast, we studied the effect of reflection provided in the form of a multiple-choice question where the choices are abstract concepts. Research with the Lisp Tutor has shown that “reflection on problem solutions that focus on understanding the abstractions underlying programs or that focus on understanding how programs work seems to be related to improved learning” [17]. Our work confirms this result, albeit for imperative programming in C/C++/Java/C#.

Acknowledgements

Partial support for this work was provided by the National Science Foundation’s Educational Innovation Program under grant CNS-042602. This work was supported in part by a grant from the Ramapo College Foundation.

5. References


Figure 1: Snapshot of the problem on selection statements – program in the left panel, explanation of the step-by-step execution in the right panel.
Figure 2: Snapshot of the reflection question in the right panel.

Figure 3: Plot of Incorrect (Class 1) versus correct (Class 2) answers to problems.
Figure 4: Loading Plot of Correct vs Non-Correct answers to problems according to independent variables. DA2 = correctly completed the problem; DA = did not correctly complete the problem.

Figure 5: Distance to Model – combined distance to model of all the independent variables on correctly solving problems.
Fig 6: Loading Plot of Correct vs Non-correct answers to problems according to independent variables without outliers. DA2 = correctly completed the problem; DA1 = did not correctly complete the problem.

Fig 7: Plot of Reflection Attempts without outliers: Class 1 (black) = incorrectly or partially solved the problem; Class 2 (red) = correctly solved the problem.
Fig 8: Color-Coded plot of Reflection Attempts without outliers: 1 attempt in red, 2 attempts in yellow, 3 attempts in green, 4 attempts in light blue and 5 attempts in dark blue