A formative evaluation of a tutor for scatterplot generation: evidence on difficulty factors

Ryan Shaun BAKER, Albert T. CORBETT,

Kenneth R. KOEDINGER, Michael P. SCHNEIDER

Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213 USA

rsbaker,corbett,koedinger,mike4@cmu.edu

We present an intelligent tutoring system for teaching students to generate scatterplots, driven by recent analyses of the difficulty factors in this domain. Prior systems in this domain emphasized point plotting and choosing appropriate scale and bounds. Initial difficulty factors analysis suggested that choosing the correct variables for the graph is the most important difficulty factor. Student performance using the system indicated that errors in point plotting are relatively easy to remediate, but that both scale choice and variable choice are more difficult to remediate. Detailed analysis of the difficulties students had with these skills while using our tutor provides insight into how to remedy them in future systems.

1. Introduction

1.1 Difficulty Factors and Intelligent Tutoring System Design

One of the largest challenges in designing an intelligent tutoring system (or other educational intervention) is tailoring the system to directly address the factors which are most difficult for students. There are at least three challenges involved in designing an educational system with difficulty factors in mind. The first is determining what the most difficult skills are, and what errors occur most often during the execution of those skills. Although this may seem obvious in some cases, our expectations of what will prove difficult to students are often confounded. The second challenge is to determine why those specific skills are difficult and why those errors arise. The third challenge, and perhaps the most difficult one, is how to use this knowledge to help us design effective educational interventions.

We believe that our answers to each of these challenges can and should be refined at each stage of the process of creating an intelligent tutoring system. Conducting a set of difficulty factors assessments (DFA) [6] before even thinking about the system design is often beneficial, as it can close off a huge range of inappropriate designs. However, this should not be the end of our investigation of student difficulty factors. By investigating in detail how students use the educational systems we create, we can learn more about what the important difficulty factors are, and about how to better address them in our designs.

In this paper, we present a case study in designing an intelligent tutor to specifically address student difficulty factors. We begin by looking at the implications of a DFA-driven model of student thinking in a domain, design an intelligent tutoring system to address the difficulty factors identified, and analyze student performance with that system in detail, gaining evidence about which difficulty factors deserve the most attention, and how to better address them. We conclude by discussing implications for re-design.

1.2 A Model of Difficulty Factors in Scatterplot Generation

Data analysis is an important emerging area of middle school education [10]. An important part of the process of data analysis is learning to create graphical representations of data. One key type of graphical representation is the scatterplot, which depicts the relationship between two quantitative variables, using a dot to represent paired values of each variable.

Over the past two years, we have developed cognitive models of the source of two important errors observed in students learning this process: the *variable choice* error and the *nominalization* error [1,2]. When a student makes the variable choice error, she or he places a nominal variable rather than a quantitative variable along one axis. In Figure 1L, for instance, the student has placed the names of different brands of peanut butter rather than their quality ratings along the X axis. Students making the related nominalization error treat a quantitative variable already chosen for one axis as if it were a nominal variable. For instance, in Figure 1R, the student should have plotted the values on the X axis as an interval scale in numerical order: 19, 20, 21, 22, 23, 24, 25. Instead, that student plotted the individual values of the variable in the same order as in the table, with one value (in this case, 23) appearing twice: 22, 20, 23, 25, 24, 19, 23. In both cases, the student is displaying behavior that would be correct if the student was attempting to create a bar graph with a nominal X axis and a quantitative Y axis, but which is inappropriate when generalized to scatterplots [1,2].

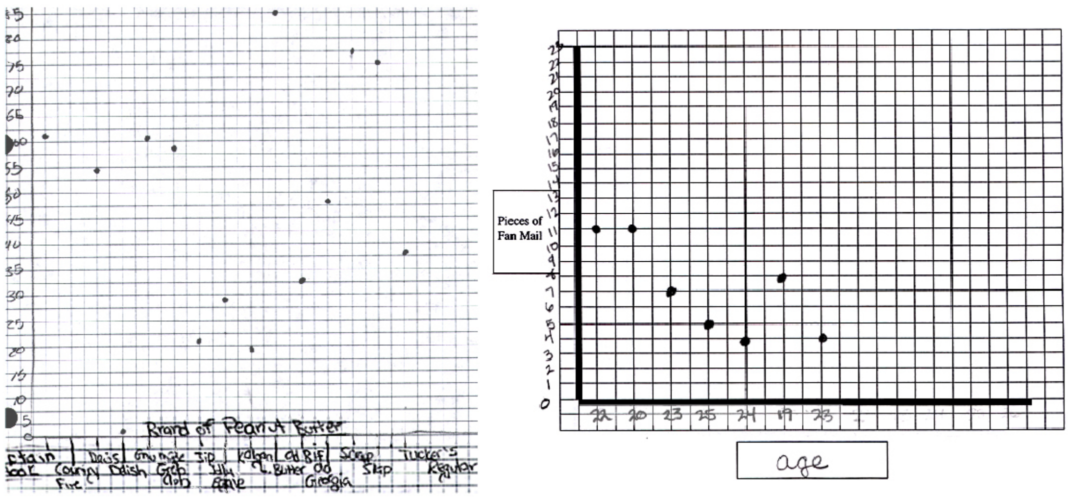

Figure 1: The variable choice (L) and nominalization (R) errors.

1.3 Prior Educational Systems

Although scatterplot generation is a relatively new area of emphasis in most curricula, there have been a number of intelligent tutoring systems (ITS) and computer-aided instruction (CAI) systems teaching similar skills, as well as paper-based curricula focusing on these skills. Many of these systems scaffold or support the process of generating a correct graph, but do not directly address the difficulty factors discussed above. We briefly discuss two of the most widely used systems here.

Tabletop [5] is probably the most-used system for teaching students to analyze data with graphs. Tabletop allows students to explore problems in data analysis, and scaffolds them in creating a number of different representations quickly and easily. When creating a graph, the student chooses the variables, and an appropriate way of representing them on the axes is immediately generated by Tabletop – although students are allowed to place the data points themselves. Ensuring that the student chooses the correct variables or type of representation is left to teacher intervention – resulting in the observation that students using Tabletop frequently choose inappropriate types of representations to answer specific questions [5]. Tabletop also bypasses the step of labeling values along the axes. In Tabletop, for example, when the student chooses an axis variable, appropriate upper and lower bounds and scale are automatically chosen for the axis, and values are automatically written along the axis. This allows the student to move straight to placing points, but gives no opportunity to remediate the nominalization error or any other difficulties students might have with these skills.

Cognitive Tutor Algebra [8] is another widely-used educational system which deals with the generation of this sort of graphical representation, though students graph points from a function rather than a scattered data set. Like Tabletop, Cognitive Tutor Algebra does not provide support for students who would, on their own, make the variable choice error. In this case, though, it is because the student never has additional variables (quantitative or nominal) they can choose. Hence students with the misconception that they should choose a nominal variable are never given a chance to express it and receive feedback. Similarly, students are not given the chance to express the nominalization error: The student chooses each axis's lower bound, upper bound, and interval between labels, and the system then labels numeric values along the axis for them, preventing the nominalization error.

2. Experimental Design

2.1 Tutor Design

As part of a broader effort to create an intelligent tutor-based curricula for middle school mathematics, we developed a cognitive tutor [8] to assist students in learning to generate scatterplots. We created two variations of this tutor, which differed in whether they allowed the nominalization error.

In the tutor, students were given a set of problems with a set of variables (including the variables to use, and both quantitative and nominal distractor variables), and a question to answer. Their task was to generate a scatterplot that could be used to answer the given question. Immediate feedback was given for student errors; an error would turn red, and if the student's action indicated a known bug, then the tutor popped up a remedial explanation of why the behavior was incorrect (but did not give the student the correct answer).

The process required to generate the scatterplot was as follows: first, the student had

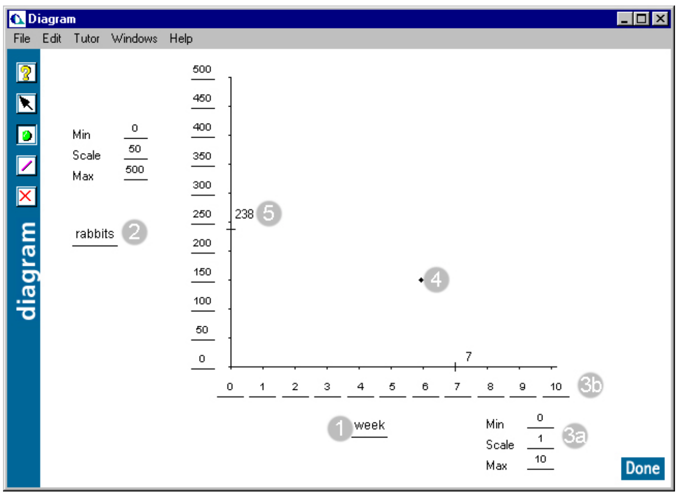

FIGURE 2: The Tutor User Interface

to choose and label the variables on the X and Y axis, by typing a variable name into a blank along each axis (labeled 1 and 2 in Figure 2). A student who chose a nominal variable was given immediate feedback – therefore, we predicted that the frequency of that error would decrease from pre-test to post-test.

Next, the student had to choose the axis's bounds and scale; in the tutor variation where nominalization was allowed, the student determined the min, max, and scale implicitly, by typing values into slots below the axis (labeled 3b in Figure 2); this allowed the student to make the nominalization error during this process. In the other variation, the nominalization error was prevented with a scaffold which had the student explicitly choose values for the min, max, and scale (labeled 3a in Figure 2) – afterwards, the system labeled the values along the axis.

Finally, the student plotted points on the graph by clicking on the point tool and then clicking on the graph where they wished to place the point. (for example at label 4 in Figure 2) A hashmark on each axis (labeled 5 in Figure 2) indicated the mouse's current location along the axis, to prevent the student from making errors due to not being able to visually translate the cursor's location across the screen. If a student plotted a point incorrectly, then the point would turn red, and could either be deleted with the delete tool or moved to another location via clicking-and-dragging.

2.2 Study Design

Forty-six students at three middle schools in urban and suburban Pittsburgh participated in our study. All of the students had just completed a unit of declarative instruction on data representation in class. Each student was given a pre-test, a post-test, and completed at least four exercises in the tutor. Every student used the tutor for the same amount of time, so some students were able to complete more problems than others – the problems were given in the same order for each student. Before students started with the tutor, they were given a practice problem, during which the user interface was explained to them. After this, they worked on the tutor on their own.

Both the pre-test and the post-test consisted of one of two nearly isomorphic problems, counterbalanced between the pre-test and post-test. In each problem, students were given a data table with two quantitative variables and one distractor nominal variable, and were asked to generate a scatterplot to show the relationship between the two quantitative variables. The scatterplots given could reasonably be drawn with a scale of 1,2, or 3. By the nature of the problems, an error at one step precluded completely correct performance on later steps – thus, each student was counted as having at most one error.

3. Student Performance

3.1 Pre-test and Post-test

Overall, the tutor was successful at improving students' ability to generate scatterplots. Table 2 shows the overall pattern of errors on the pre-test and post-test. The percentage of

Table 2: Percentage of students making each error, on pre-test and post-test

Error	Pre-Test	Post-Test
Variable Choice Error	17%	13%
Nominalization Error	0%	0%
Point Plotting Error (>1 point incorrect)	9%	0%
Scaling Errors	17%	15%
General Miscomprehension Errors	7%	0%

students who made any errors decreased significantly from 50% at pre-test to 28% at post-test, $\chi^2(1, N=92)=4.56$, $p<0.01$. Students started out fairly good at point plotting, with only 9% plotting more than one point incorrectly on the pre-test, but were almost perfect by the post-test, with no student plotting more than one point incorrectly. This was a significant gain in skill, $\chi^2(1, N=92)=4.18$, $p=0.05$. The percentage of students making errors stemming from generally misunderstanding scatterplots also reduced marginally significantly, going from 7% on pre-test to 0%, $\chi^2(1, N=92)=3.10$, $p=0.08$. Examples of this include combining two points into one point when they had the same X value, drawing a frequency histogram of one variable, and connecting the dots of the graph into shapes.

On the other hand, our tutor was less effective at remedying errors in two key skills. There was a positive trend with the variable choice error, from 17% to 13%, but this effect was not significant, $\chi^2(1, N=92)=0.34$, $p\sim0.85$. The tutor was also not effective in reducing the incidence of errors in choosing a scale, which went from 17% at pre-test (which was much higher than we had anticipated) to 15% at post-test. As was observed in [7], there were several different types of scale choice errors that made it difficult for students to plot all of the points. Some students chose scales too small, making it impossible to put all of the points in the graph. Other students chose scales too large, compressing all of the points into a very small area of the graph.

One additional surprise was that the nominalization error was not observed on the pre-test in either condition – we neither expected nor saw other differences between the two versions of the tutor.

Hence, the tutor's overall effects on student performance were positive but left considerable room for improvement. In the following sections, we will consider evidence from log-files of tutor usage, and see what they can suggest both for our future attempts in this domain, and possibly for other educational or tutoring efforts in this domain.

3.2 Tutor Time Usage

Analysis of the time students spent exercising different skills, as in [9], provided interesting evidence about how time could better be allocated in future tutors for this domain. In the original design of our tutor, we elected to have the student complete the entire process of drawing a graph, as they would on paper. However, this over-emphasized some skills at the expense of others. Because each problem had at least eight points, students spent more time (212 seconds per problem, 48% of total time) plotting points than on any other part of the process of creating the scatterplot.

The set of skills which took the next most time (132 seconds per problem) was setting up bounds and scale for each axis. This varied less than 5 seconds between the two versions of the tutor. Students took the least time (96 seconds) to choose variables.

This demonstrates the necessity of careful attention not only to including the relevant skills but also to the time spent on each skill. Simply giving each skill a frequency of application similar to that required in the external use of the skill means that students spend the most time on the skills which are naturally most time-consuming, rather than the skills which are most important or most difficult. In this case, students spent the most time plotting points, even though it was not the most difficult part of the task. However, it is uncertain whether the amount of time spent plotting points was appropriate to guarantee mastery, or whether it was over-kill. To determine this, we investigated the pattern of student performance of this skill within the tutor.

3.3 Point plotting in the tutor

We analyzed the pattern and frequency of point plotting errors within the tutor. Students had the most difficulty on the first opportunity to plot a point in the tutor, plotting 41% of points incorrectly. This level of difficulty was not observed again within the students' usage of the tutor – students averaged 15% error on the first point of subsequent problems, and 9% error on all other points. However, student performance at point plotting had considerable variation between points. Two sources of difficulty seemed to account for a substantial portion of the variance in student performance. ($r^2 = 0.287$) First, as described above, students were more likely to get the first point of each problem incorrect. This stemmed from a specific error: inverting the point's X and Y axes. After students committed this error once on any given problem (which occurred an average of 11% of the time on the first point of the problem), they committed it only 2% of the time on subsequent points within the same problem – but often committed it on the next problem. The second source of difficulty was plotting points in problems that required a larger scale, possibly because the screen space for each point was smaller. Students did not improve at plotting points on graphs of large scale over time, suggesting that this difficulty factor may result in more frequent slips rather than an increase in one specific error.

There were two remaining difficulty factors, both of which caused a spike in errors the first time they occurred but were not subsequently a source of error: using a value from the wrong variable when there were more than two quantitative variables, and plotting a point which had 0 for its value along one of the axes. These errors' transitory nature suggests that they required only minor refinement to knowledge.

After observing the large role of slips and transitory errors in student performance with this tutor, it was unsurprising to see that slips played a substantial role in the best-fitting version of the two-state Bayesian model of learning used in cognitive tutors [3]. This model estimated that students had a probability of 0.67 of knowing how to plot points before starting the tutor, a probability of 0.86 of learning the skill at each opportunity to plot a point, explaining why performance improved substantially after the first point, a probability of 0.09 of slipping and getting an incorrect answer despite understanding how to plot points, and a probability of 0.01 of guessing and plotting a correct point despite not knowing how to. The model estimated that 100% of the students in our study understood how to plot points at the conclusion of using our tutor, but that only 91% of points would be plotted correctly. Given the nearly perfect performance on the post-test, this suggests that specifics of the tutor's interface may have made point plotting more slip-prone in the tutor than on paper.

Most importantly, the model estimated that after plotting just four points, every student in our study had a probability of 0.99 or higher of knowing how to plot points. This suggests that, indeed, the students using our tutor spent substantially more time than necessary plotting points, and that (with attention to the specific difficulty factors) this skill's emphasis could be substantially reduced in general in future tutors for this domain.

3.4 Choosing variables in the tutor

Another skill which students appeared to learn within the tutor was the skill of choosing quantitative variables. On the first tutor problem which included a distractor nominal variable in the data table, 24% of the students attempted to place the nominal variable on at least one axis, with (somewhat surprisingly) more attempting to place a nominal only on the Y (13%) than only on the X (7%). This behavior was very quickly extinguished by the tutor's feedback – no more than one student ever attempted to place a nominal variable on either axis on any subsequent problem, a significant reduction in error, χ^2 (1, N=92)=9.58, p<0.01. Based on this performance, the best-fitting version of the two-state Bayesian model of learning used in intelligent tutoring systems [3] estimated that 99% of students would be able to correctly choose two quantitative variables on the post-test. This is a substantial overestimation, and suggests that students are not properly generalizing the knowledge they have learned about choosing variables in the tutor. Therefore, the next version of this tutor should focus on helping students put the knowledge gained about variable choice in context, helping students understand the differences between scatterplots and bar graphs and the functional purpose underlying those differences. The key to defeating this misconception may arise from creating new conceptions which help students anchor the skills they learn in the tutor in a deeper conceptual understanding. [4,12]

3.5 Choosing bounds and scale in the tutor

Student performance at choosing the bounds and scale for a given axis did not improve substantially during their use of the tutor, which is consistent with their pre to post test performance. The percentage of students who failed to select an appropriate scale (somewhat arbitrarily defined as being 1, 2, or a multiple of 5, and spreading the values of the data set into 5-10 segments) did not change substantially – for instance, the percent of students who chose an incorrect scale went from 52% on the second problem to 45% on the last problem that at least half of the students completed, χ^2 (1, N=92)=0.39, p~0.6. The rate of errors in choosing a lower bound for the X axis also did not change substantially, going from 24% to 26%. In both cases, a similar pattern was observed on the Y axis. One specific and frequent error was when students chose the maximum value of the data set as the upper bound even when this value was not tractable, given their choice of lower bound and scale. 61% of students made that error on the first problem where it was possible, 17% committed it on the second problem, and 0% on all subsequent problems.

With the exception of this specific error, performance did not substantially improve at bounds and scale selection in our tutor, suggesting that more support might be necessary. One potential improvement might be to create a scaffold that explicitly leads students through the process of choosing correct values for these quantities. Another might be to give more explicit feedback to students who choose inappropriate values, showing them that certain points can not be accommodated by their choice of bounds or scale.

4. Conclusions

In this paper, we have presented a tutor for scatterplot generation in the early stages of its evolution. The tutor we produced was successful at improving student performance; however, it was more successful at remediating some errors than others. There is substantial evidence that it, like previous efforts, overemphasized the skill of point plotting; however, there still is some need to address this skill, since 9% of students plotted two or more points incorrectly on the pre-test. Our tutor correctly focused on variable choice, but needs to improve in how it addresses this skill, and to focus more on scale/bounds selection.

One of the challenges in designing a tutor for a multi-step process, such as generating a scatterplot, is figuring out how to emphasize one step without ignoring other steps. Simply eliminating point plotting would probably be undesirable for helping students learn and connect together the entire process of scatterplot generation; even using mastery learning to remove completely learned parts of the process might hinder learning of the whole. One potential solution is to integrate problem-solving and worked example as in

[11]. This would give the student the opportunity to execute the skills they do not yet have within the full context of the problem-solving process. Bayesian knowledge tracing, supported by evidence on student learning rates from this study, will help us determine for each student what parts of the problem-solving process need the most practice.

On the whole, this study illustrates the value of continuing research on the errors students make even when using a fairly successful tutoring system. Even when we have a system where student performance improves substantially from pre-test to post-test (in this case, with errors decreasing almost by half), there is often substantial room for improving the system. It is very important to have knowledge about difficulty factors before the design process even begins; but knowledge gained from formative evaluation of a tutoring system is of additional value, both for determining which difficulty factors should be emphasized, and to improve how those difficulty factors are addressed. Knowledge about the pattern of student performance during tutor usage, including the time spent on each skill [9] and when specific errors occur, can be used in redesign and interface refinement – helping us, in the long-term, to develop even more effective educational interventions.

Acknowledgements

We would like to thank Angela Wagner, Geoff Kaufman, Jay Raspat, Megan Naim, John Argentieri, and Katy Getman for assistance in conducting the studies at the schools, Sara Stille, Jane Kamneva and Heather Frantz for help in data entry, and Stephen Blessing, Suzanne Charman, Santosh Mathan, Steve Ritter, Bethany Rittle-Johnson, and Eli Silk for helpful discussions and suggestions. This work was funded by an NDSEG (National Defense Science and Engineering Graduate) Fellowship, and by NSF grant 9720359 to "CIRCLE: Center for Interdisciplinary Research in Constructive Learning Environments".

References

[1] Baker, R.S., Corbett, A.T., and Koedinger, K.R. (2001) Toward a Model of Learning Data Representations. *Proceedings of the Cognitive Science Society Conference*, 45-50

[2] Baker, R.S., Corbett, A.T., and Koedinger, K.R. (2002) The Resilience of Overgeneralization of Knowledge about Data Representations. *Presented at American Educational Research Association Conf.*

[3] Corbett, A.T. & Anderson, J.R. (1992) Student Modeling and Mastery Learning in a Computer-Based Programming Tutor. *Second International Conference on Intelligent Tutoring Systems*, 413-420.

[4] Corbett, A.T. and Trask, H. (2000). Instructional interventions in computer-based tutoring: Differential impact on learning time and accuracy. *Proceedings of ACM CHI'2000 Conference on Human Factors in Computing Systems*, 97-104.

[5] Hancock, C., Kaput, J.J., & Goldsmith, L.T. (1992) Authentic Inquiry With Data: Critical Barriers to Classroom Implementation. *Educational Psychologist*, 27(3), 337-364.

[6] Heffernan, N. & Koedinger, K. R. (1998). A Developmental Model For Algebra Symbolization: The Results of a Difficulty Factors Assessment. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, (pp. 484-489). Mahwah,NJ: Erlbaum.

[7] Kerslake (1981) Graphs. In K. M. Hart (Ed.), *Children's understanding of mathematics*: 11-16 (pp. 120-136). London: John Murray.

[8] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education*, 8, 30-43.

[9] Mostow, J., Aist, G., Beck, J., Chalasani, R., Cuneo, A., Jia, P., & Kadaru, K. (2002) A La Recherche du Temps Perdu , or As Time Goes By: Where does the time go in a Reading Tutor that listens? *Sixth International Conference on Intelligent Tutoring Systems*, Biarritz, France.

[10] National Council of Teachers of Mathematics. (2000) *Principles and Standards for School Mathematics*. Reston, VA: National Council of Teachers of Mathematics.

[11] Renkl, A., Atkinson, R.K. & Maier, U.H. (2000) From studying examples to solving problems: Fading worked-out solution steps helps learning. *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, 393-398. Mahwah, NJ: Erlbaum.

[12] Strom, D., Kemeny, V., Lehrer, R., Forman, E. (2001) Visualizing the emergent structure of children's mathematical argument. *Cognitive Science*, 25 (5), 733-773.