The Effect of Feedback Control on Learning to Program with the Lisp Tutor

Albert T. Corbett
John R. Anderson

Department of Psychology
Carnegie-Mellon University

Abstract

Control and content of feedback was manipulated as students practiced coding functions with the Lisp Tutor. Four feedback conditions were employed: (1) immediate error feedback and correction, (2) immediate error flagging but immediate correction not required (3) feedback on demand and (4) no tutorial assistance. The wide range in feedback conditions did not affect mean learning rate as measured by individual production firing, time to complete the exercises or post-test performance. However, post-test results were more highly correlated with student ability as tutorial assistance decreased across conditions. Feedback conditions also affected students' monitoring of the learning process. Across groups, students found the material was easier and believed they had learned it better as assistance decreased across conditions. However, students who received more assistance estimated their mastery of the material more accurately. Finally, students reported relatively little preference for one tutoring condition over the others. Students who could exercise the most control over feedback reacted fairly passively to the tutoring conditions; students in condition 3 tended not to ask for much help and students in condition 2 tended to correct error immediately although it was not required.

Introduction

This study examines four types of feedback in the context of students learning to program with the Lisp Tutor. The four conditions vary from fairly rigid step-by-step feedback to a condition in which no feedback is available. We are interested in several theoretical and practical issues in this research: (1) whether immediate feedback is optimal for learning as predicted by an earlier version of ACT* (2) whether feedback conditions affect students monitoring of how well they have mastered the material and (3) whether students have preferences among the feedback conditions.

The Lisp Tutor is a program that provides assistance to students as they work on lisp coding exercises (Anderson & Reiser, 1985). As the students type their code, the tutor monitors their progress step-by-step, provides feedback on errors and provides the correct next action when a student appears to be floundering. The tutor contains a production system ideal student model and provides feedback by comparing the students responses to correct and buggy productions that could fire at each step. The tutor was developed in large part to test the ACT* model of procedural learning in a "real-life" context and has also been used to explore tutorial variables (Anderson, Conrad & Corbett, 1989)

The standard lisp tutor constrains students to enter their code top-down, depth-first and left-to-right, and presents immediate, atom-by-atom feedback on errors. When an error is made, the tutor interrupts, presents a feedback message, deletes the error and requires the student to try again. An early version of the ACT* model suggested that immediate feedback should facilitate the proceduralization of declarative knowledge, since the formation of productions depends on the working memory trace of the problem solving episode (Anderson, Farrell & Sauers, 1984; Anderson, Boyle, Farrell & Reiser, 1987). However, immediate feedback does not uniformly enhance learning (Kulik & Kulik, 1988) and in earlier studies with the Lisp Tutor, removing feedback content or delaying feedback has not had any impact on degree of learning as assessed in a post-test (Anderson, Conrad & Corbett, 1989; Corbett & Anderson, 1989; Corbett, Anderson & Patterson, in press). Moreover, in some circumstances, immediate feedback can interfere with the students' monitoring of the learning process (Schmidt, Young, Swinnen & Shapiro, 1989). As a result, in this study we have implemented four feedback conditions. Unlike earlier studies, these conditions are implemented in the same program architecture, making it possible to compare production firing times. In addition to this direct test of the theory, we measure post-test performance, time to complete the exercises and have collected questionnaire data.

The four feedback conditions are (1) standard immediate feedback, as described above; (2) error flagging, (3) feedback on demand and (4) no feedback (see Corbett, Anderson & Patterson for implementation details). The three non-standard conditions differ from the standard immediate feedback condition in two ways. First, in these three conditions, students entered their code with a true structured editor. Thus, they could type their code in any order, and edit their code at will. Second, the conditions varied with respect to feedback content and control

In the error flagging condition, the tutor provided immediate feedback, but did not interrupt the students. It flagged any code it did not recognize by displaying it in boldface, but did not display any explanatory text and the students remained in control of their actions. They were free go back and fix the error, ask for a comment on the error (the same message provided automatically in the immediate feedback condition) or to continue coding. The tutor recognizes one or more roughly optimal solutions to the exercises. Errors are flagged on the basis of this knowledge, so, in effect, the tutor is guiding students to an optimal answer. However, all three non-standard versions ultimately will accept any code that works and students knew in this condition that they might have a workable solution, even if some of the code were flagged. If the students completed an exercise with working, but unrecognizable code, the tutor also displayed one of its optimal solutions. There are possible theoretical and practical advantages in this condition. First, students may benefit from generating their own explanations of errors, instead of automatically receiving feedback. Second, in the course of learning, larger productions may be compiled. In that case, atom-by-atom feedback could interrupt the firing of productions. If so, this version of the tutor allows students to complete a production before pausing to deal with an error.

In the feedback on demand condition, the tutor never interrupted the student. However, the student could ask the tutor at any time to check over the code. In this case, the tutor checked the code top-down and as soon as it found an error, provided the same feedback message that the standard tutor would have presented automatically when the error is made. If there were no errors, it informed the subject accordingly. (It should be noted that in all three versions of the tutor that have been discussed so far, the student could also ask for two other types of help: (1) a hint at any goal in the solution (2) the correct action to take at any goal.) If students benefit from detecting as well as correcting their errors, this feedback on demand condition could prove to be optimal.

In the no feedback condition, no tutorial assistance was available to the students as they generated their code. Students in all conditions had access to a lisp environment during coding, so they could load and test their code, but this was the only mechanism available to students in this condition for debugging. In the three non-standard conditions, the students indicated their solutions were complete by pressing a key. In the error flagging condition and feedback on demand conditions, students could not move on to the next exercise until their solution was correct. Students in the no tutor condition were ultimately allowed to move on with incorrect code, but only after trying at least five times to get the correct answer. If the student gave up, the tutor showed them a correct solution.

The Experiment

This experiment was conducted in a class and a total of 55 students participated. Each student worked with one version of the tutor. The four versions of the tutor were implemented for the first two lessons of the lisp tutor. These lessons cover an introduction to function calls and to function definitions, and consist of a total of 20 exercises. While some of these exercises are difficult, most students are ultimately capable of generating solutions even without help from the tutor. After completing these lessons, all students reverted to the standard tutor and so had the same learning environment for the remaining 10 lessons in the course. A paper and pencil post-test was administered after the first two lessons. Four more paper and pencil tests were administered throughout the rest of the course. Since all students had the same learning experiences over the last ten lessons these last four tests will serve as a measure of relative ability. In addition, questionnaires on the tutor were administered after the second and fourth lessons.

Results

Learning. Scores on the post-test following lesson 2 are displayed in Table 1. The effect of tutor type is not significant in an analysis of variance. Because of attrition and absence, the students in the four groups were not well matched. As a result, we used students' mean performance on the remaining tests in the course as a measure of relative ability and removed the effects of this measure in an analysis of covariance. The adjusted means are also displayed in Table 1 and again the effect of tutor type is not significant. Average time to complete each exercise is also displayed in Table 1. Again, there are no significant differences in an analysis of variance or an analysis of covariance that removes the effects of students' relative ability.

Table 1

Mean Post-Test Scores (percent correct) and Mean Exercise Completion Times (minutes) for the Four Versions of the Tutor.

	Immediate Feedback	Error Flagging	Demand Feedback	No Tutor
Post-Test Scores	55%	75%	75%	70%
Adjusted Scores (ANCOVA)	61%	75%	71%	67%
Exercise Times	4.6	3.9	4.5	4.5
djusted Times (ANCOVA)	4.1	4.2	4.5	5.0

There is no evidence in these global measures that the feedback manipulation affected learning. To analyze the effect of feedback on learning more closely we also examined production firing accuracy and firing time in the course of coding. Accuracy is operationally defined as the probability that the student types an atom that corresponds to an appropriate underlying production in the tutor's ideal student model. Production firing time is only measured for correct coding cycles. It is operationally defined as the time from the onset of the prompt at the beginning of a coding cycle until the onset of the prompt for the next cycle. Table 2 displays mean time and accuracy data for the first through fifth opportunities to fire each production. As can be seen accuracy increases and firing time decreases with repeated production firings. Note that there is a speed accuracy tradeoff between the

Table 2

Production Firing Accuracy (percent correct) and Production Firing Time (seconds) in Lesson 2.

	First Occurrence	Second Occurrence	Third Occurrence	Fourth Occurrence	Fifth Occurrence
Immediate Feedback					
Accuracy	83%	90%	95%	95%	91%
Time	18.9	14.5	8.9	8.2	8.9
Error Flagging					
Accuracy	62%	69%	82%	73%	81%
Time	11.5	6.0	5.4	4.1	3.0
Feedback on Demand					
Accuracy	59%	52%	88%	82%	70%
Time	15.7	9.0	6.7	4.9	4.9
No Tutor					
Accuracy	59%	67%	80%	78%	76%
Time	15.0	9.2	7.4	6.7	4.6

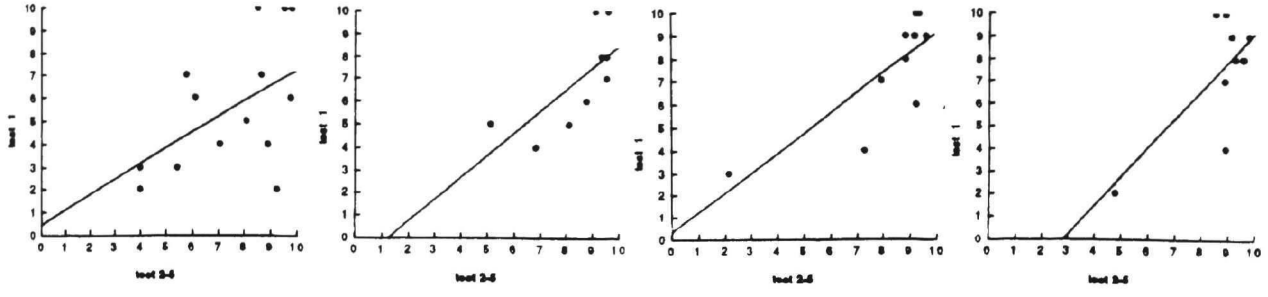
three non-standard conditions and the immediate feedback condition. Students in the immediate feedback condition are responding more accurately at each serial position, but they are also responding more slowly. These students are more cautious in the immediate feedback condition, perhaps because the tutor reacts strongly to each mistake in this condition, but there is no evidence that they are learning the productions better. Thus, it seems that the process of

Conrad & Corbett (1989) and in contrast to the earlier proposals of Anderson, Boyle, Farrell & Reiser (1987). The speed accuracy tradeoff itself is interesting, since there is currently no mechanism in ACT* that allows such a tradeoff in firing productions.

Finally, to assess whether ability interacted with tutoring condition, we we plotted Test 1 accuracy against our ability measure (students' performance on the remaining tests in the course) in Figure 1. The slopes generally grow progressively steeper and the goodness of fit generally increases as the the control exerted by the tutor diminishes. The standard immediate feedback condition has the lowest slope, while the slope for the no tutor condition is almost twice as large. The two other versions fall in between. This result suggests, as might be expected, that when less assistance is provided, student's performance depends more heavily on relative ability.

Figure 1

Test 1 Accuracy Plotted Against Average Accuracy on Remaining Tests



Immediate Feedback	**Error Flagging**	**Feedback on Demand**	**No Tutor**
Slope=0.68	Slope=0.97	Slope=0.88	Slope=1.27
r=0.50, p<.10	r=0.70, p<.05	r=0.80, p<.01	r=0.72, p<.01

Questionnaire 1: Monitoring the Learning Process. A questionnaire was administered after students completed the first two lessons with the tutor, but before taking the post-test. The questionnaire contained seven questions to which students responded on a seven point scale (See Table 3). Two of the questions concerned the students perception of the learning process, since to the ability to self-monitor this process is an important skill in students (Chi, Bassok, Lewis, Reimann & Glaser, 1989). These are the first two questions in Table 3. The first question asked how difficult the exercises were and the differences between the groups were marginally significant ($F(3,51) = 2.25$, $p < .10$) As can be seen in Table 3, there is a perfect correlation between degree of tutorial assistance and ratings of difficulty. (Pairwise comparisons between the no feedback and error flagging, and between no feedback and standard tutor were reliable). The paradoxical result is that the more assistance provided by the tutor, the harder the exercises seemed. It appears that perceived difficulty reflects not the degree of effort expended by the student, but rather the degree to which an external source points out the students' errors. We generated scatter plots for this question to examine the relationship of the ratings to student ability. In Figure 2, the seven point rating scale appears on the abscissa and student performance on the post-test appears on the ordinate. As can be seen, the slope of the best fitting straight line is steeper for the standard and error flagging condition, both of which provide immediate feedback, than for the feedback on demand and no tutor conditions. While the immediate feedback groups perceived the questions as more difficult, it appears that the students within the two immediate feedback conditions were more sensitive to the difficulties they experienced. Moreover, in the feedback on demand and no tutor groups the correlation between difficulty estimates and post-test performance is small and nonsignificant.

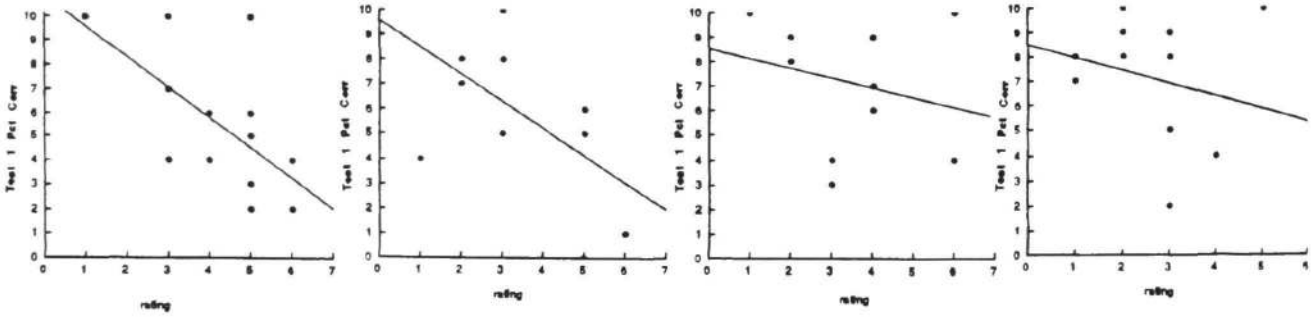
Table 3

Questionnaire 1 Mean Ratings

	Imm Fdbk	Error Flag	Demand Fdbk	No Tutor
1. How difficult were the exercises? (1=Easy, 7=Challenging)	4.1	3.9	3.4	2.8
2. How well did you learn the material? (1=Not Well, 7=Very Well)	5.4	4.6	5.4	5.8
3. How much did you like the tutor? (1=Disliked, 7=Liked)	5.2	4.5	4.8	4.9
4. Did the tutor help you finish? more quickly (1=Slower, 7=Faster)	5.1	4.6	4.7	4.5
5. Did the tutor help you understand better? (1=Interferred, 7=Helped)	5.3	4.9	4.7	4.7
6. Did you like the tutor's assistance? (1=Disliked, 7=Liked)	5.3	5.0	4.7	4.7
7. Would you like more or less assistance? (1=Less, 7=More)	4.3	4.9	4.5	4.6

Figure 2

Test 1 Accuracy Plotted Against Ratings of the Difficulty of the Lessons



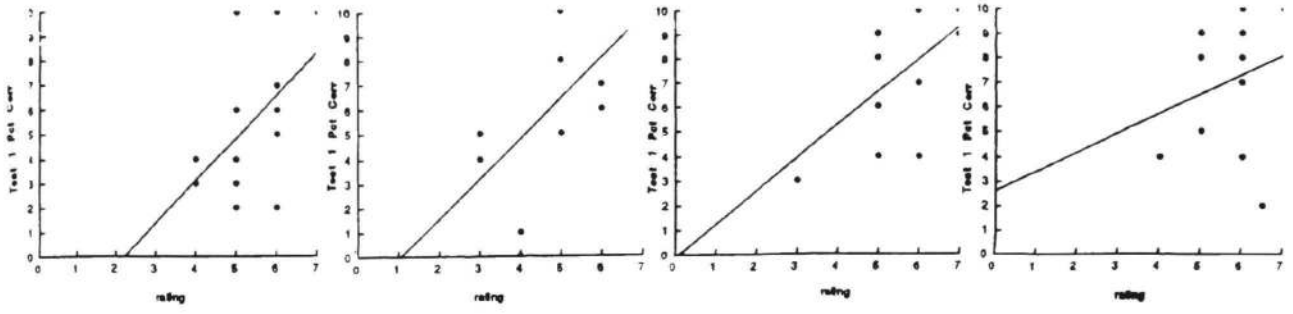
Immediate Feedback	**Error Flagging**	**Feedback on Demand**	**No Tutor**
Slope=-1.27	Slope=-1.10	Slope=-0.40	Slope=0.52
r=0.62, p<.05	r=0.53, p<.10	r=0.26, ns.	r=0.24, ns.

The second question asked how well the student had learned the material. Again, there was a significant effect of tutor type, ($F(3,51)=3.57$, $p < .05$). (Pairwise comparisons between the error flagging and no tutor conditions and between the error flagging and standard condition were significant). The results of this question are similar to what would be predicted from the prior question; generally the groups that thought the material was easy also thought they had learned it better. The exception is that the standard immediate feedback condition has shifted upward.

This may reflect the fact that this is the only group that received a written explanation from the tutor each time they made a mistake. Scatter plots for this question are displayed in Figure 3. Again, the slope is higher for the two groups receiving immediate feedback, suggesting, that the subjects in those groups are more sensitive to how well they'd learned the material.

Figure 3

Test 1 Accuracy Plotted Against
Ratings of How Well the Material was Learned.



Immediate Feedback	**Error Flagging**	**Feedback on Demand**	**No Tutor**
Slope=1.75	Slope=1.64	Slope=1.34	Slope=0.79
r=0.52, p<.05	r=0.51, p<.10	r=0.59, p<.10	r=0.24, ns.

Questionnaire 1: Tutoring Preferences. The remaining five questions concerned the assistance provided by the tutor. The surprising result is that despite the wide range of control exerted by the tutor across the four conditions, students did not display any preferences among the tutoring conditions. There were no significant differences among the four groups in rating (1) how much they liked working with the tutor, (2) how much help the tutor was in completing the exercises, (3) how well they liked the tutor's assistance and (4) whether they would prefer more or less assistance.

An interesting aspect of this issue is that the students in the two groups who were able to exercise control, the error flag and feedback on demand groups, gravitated toward the two extremes. In the case of the feedback on demand tutor, only 3 of 11 subjects ever asked the tutor to check over code (only 2 asked more than once), 2 asked for goal hints and 4 asked for any explanations. In the case of the error flagging condition, subjects on average corrected 58% of their errors immediately and 76% after no more than one intervening operation. Thus, when given control, students respond fairly passively. If the tutor does not volunteer help, the students tend not to ask for it. If the tutor flags errors, students tend to respond immediately, even though they are free to do otherwise.

Questionnaire 2. After students had all worked with the standard immediate feedback tutor for two lessons, we asked them to fill out a second questionnaire comparing the two tutors they had used. This questionnaire contained four questions, displayed in Table 4, that required a rating on a seven point scale. As can be seen in question 1, all four groups preferred the first version of the tutor. This may seem anomalous for the group that used the standard tutor in the first place. However, beginning in the third lesson, students no longer had access to a lisp environment while doing exercises, and as can be seen in question 4, all four groups liked having the Lisp environment available. The second question asked about the editing facilities and there was a marginally significant effect of tutor type here, $F(3,48)=2.19$, $p=.10$. In general, students who had a true structured editor preferred that interface to the more constrained interface in the standard condition. Finally, the third question asked about the feedback and help properties of the tutor and, surprisingly all four groups rated themselves as indifferent between the tutor they used for the first two lessons and the standard immediate feedback tutor used subsequently.

Table 4

Questionnaire 2 Mean Ratings

	Imm Fdbk	Error Flag	Demand Fdbk	No Tutor
1. Which tutor do you like better? (1=Version 1, 7=Standard Tutor)	2.9	2.0	2.6	2.7
2. Which editing facilities did you prefer? (1=Version 1, 7=Standard Tutor)	3.4	2.0	3.0	2.3
3. Which feedback and help facilities did you prefer? (1=Version 1, 7=Standard Tutor)	4.1	3.6	3.1	3.9
4. Did you like having the Lisp environment? (1=Yes, 7=No)	2.5	2.2	2.1	2.1

In summary, six questions on tutorial assistance did not reveal any preferences. There are two points that qualify this conclusion. First, when students are given a structured editor to enter their code, they almost always conform to the top-down, left-to-right, depth-first constraint imposed by the tutor. They only deviate to go back and fix errors. This suggests that the preference for the true structured editor expressed in question 2 actually reflects a preference for detecting and fixing their own errors. Second, an earlier study (Corbett & Anderson, 1989) compared just the standard tutor to the error flagging tutor and asked specifically how much students liked the *immediate* feedback presented by the tutor. In that study, students in the error flagging condition gave a more positive rating than those in the standard condition. Thus, there is some evidence that students do not like being interrupted when they make mistakes.

Conclusions

In summary a wide range of tutor vs. student controlled feedback led to no differences in how quickly nor how well students learned how to write Lisp code. This is consistent with our current view (Anderson, Conrad & Corbett, 1989) that learning is a function of the product of a problem solving episode and not the process. As long as the students produce the same code and come to the same understanding of the code, it does not matter what cognitive trajectory they took to produce the code. As our tutorial manipulation is extended to more complex functions, we expect that the feedback manipulation will begin to affect time to complete the exercises. However, we expect that production formation and post-test performance will not be affected.

While the manipulation of feedback seemed not to affect learning, it did influence the students self-perception. Students who received less assistance from the tutor seemed more confident of the skill, however that confidence tended to be unrelated to performance on the post-test. Students who received immediate feedback seemed to assess their knowledge more realistically.

Concerning students' attitudes, there is a general preference for an enhanced feeling of control, even though students don't take much advantage of the added flexibility. When given a structured editor they still tend to type their code top-down, left-to-right and depth-first. When provided a Lisp environment less than half the students use it, except in the no tutor condition. Finally, students apparently prefer to find and fix their own errors and if errors are pointed out, they prefer not being interrupted. However, none of the feedback manipulations affected how well the material was learned.

References

Anderson, J.R., Boyle, C.F., Farell, R. and Reiser, B.J. (1987). Cognitive principles in the design of computer tutors. In P. Morris (Ed.) Modelling Cognition (pp. 93-134). New York: Wiley.

Anderson, J.R., Conrad, F. and Corbett, A.T. (1989). Skill acquisition and the LISP Tutor. Cognitive Science, 13, 467-505.

Anderson, J.R., Farrell, R. and Sauers, R. (1984). Learning to program in LISP. Cognitive Science, 8, 87-129.

Anderson, J.R., and Reiser, B.J. (1985). The LISP Tutor. Byte, 10, 159-175.

Chi, M.T.H., Bassok, M., Lewis, M.W., Reimann, P., Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, 13, 145-182.

Corbett, A.T. and Anderson, J.R. (1989). Feedback timing and student control in the Lisp Intelligent Tutoring System. The Proceedings of the Fourth International Conference on AI and Education., Amsterdam, Netherlands.

Corbett, A.T., Anderson, J.R. and Patterson, E.G. (in press). Student modeling and tutoring flexibility in the Lisp Intelligent Tutoring System. In C. Frasson and G Gauthier (eds.) Intelligent tutoring systems: At the crossroads of artificial intelligence and education. Norwood, NJ: Ablex.

Kulik, J.A. & Kulik, C.C. (1988). Timing of feedback and verbal learning. Review of Educational Research, 58, 79-97.

Schmidt, R.A., Young, D.E., Swinnen, S., & Shapiro, D.C. (1989). Summary knowledge of results for skill acquisition: Support for the guidance hypothesis. Journal of Experimental Psychology: Learning, Memory and Cognition, 15, 352-359.