
A Statistical Analysis of Adaptive Hints

Zhen Zhai
UC San Diego

zzhai@eng.ucsd.edu

Yoav Freund
UC San Diego

yfreund@eng.ucsd.edu

Abstract

1 INTRODUCTION

Intelligent Tutoring System(ITS) is well studied and proven to have positive effects on learning[?]. The goal is to have a system replicate the interaction between a human tutor and a student in order to provide instance help to struggling student. In fact, research has suggested that a computer tutor can nearly be as effective as a human tutor[?]. Therefore many ITS have been developed and integrated into classrooms. A few well developed ITS designed for algebra curriculum in high school mathematics has been proven to be effective[?][?].

Homework is an essential part of students' learning process[?]. It not only gives students a chance to practice but also allows them to identify their confusions and resolve them. Many web-based homework systems for college students have been well developed (e.g., WebAssign, WebWork, OWL). The goal is to help instructors to have a better management of a large enrollment class. These homework systems often provide feedback to students and provide analysis of student performance to instructors. Researches have proven the web-based homework systems to be helpful[?][?]

Our goal is to embed ITS with a web-based homework system to give formative feedback to students to help their learning process when doing their homeworks. Well designed feedback, hint, is essential to improve the learning process[?][?]. We target formative feedback as our hints to students. According to Shute, formative feedback needs to be specific, clear and timely[?]. The formative feedback should provide learner with both verification and elaboration[?][?]. Verification, is to provide feedback to learner as to whether the answer is correct or not. Elaboration, is to provide a short elaborate on the topic or discuss the incorrect response. Also, we used delayed feedback supported by Kulhavy and Anderson[?]. We only provide hints to students who have spent some time on the problem. Furthermore, we give on-demand hints. Studies show that sending hint-on-demand allow students to learn more comparing to proactive hints [?].

There is a common concern about students gaming the system, especially on harmful gaming, where gaming the system leads to poor learning outcome[?][?]. In previous studies, students game the system by tricking the tutoring systems into giving out correct answers[?]. As a result, students succeed in homework but fail on learning. We don't have this concern because our hints don't contain correct answers in any form. The adaptive hint system generates short simple questions as hints instead of worked-examples, which is commonly used among human tutors[?]. In fact, it is suggested that worked examples have no significant effects on learning in a web-based intelligent tutoring environment[?]. Therefore, we replaced worked examples with short questions that are designed to promote thinking and guide the learning. This resolve the concern of students gaming the system because our system gives students hint on how to learn, not how to answer a specific problem. The only way for a student to obtain correct answers is by learning how to solve the problem. Therefore, we not only prevent any system gaming, but our hints also help students in long term. Furthermore, our hint system targets math problems, which require students to type in math expressions. Therefore, unlike multiple choice problems, it would be nearly impossible to guess the answer by looking at hints. There is infinity many ways one can type in a math expression and it is

very hard for students to simply guess it correctly with the help of hints. Therefore, we don't have the concern of students gaming the system.

Contribution

We built and implemented an adaptive hint system to help when students found themselves struggling with homework problems. The adaptive hint system will first study a student's incorrect answer to identify the student's confusion. We target math expression problems where most students will have a part of the expression that is incorrect, a subexpression. Therefore the system will generate a message to point out which subexpression is incorrect. This allows students to see clearly why the answer is wrong. Then, based on the identified confusion, the system will give the student a smaller question to solve. This smaller question is specifically designed to target the student's confusion. The smaller question is carefully designed by instructors and teaching assistants (TAs) so that it is simple enough that a student can easily understand and answer. Our goal is to let the student solve the small tasks before moving on to solve the bigger task. This way, the adaptive hint system guides students in their learning process from understanding their confusions to resolve their confusions.

The system is adopted in a statistical classes of 300 students ...

Our statistical result shows that students, with the help of the adaptive hints, learn to solve the problem and can solve similar problems more efficiently afterward.

* Our analysis demonstrates with statistical significant $p =$ that students who view the hints have an improved score on later problem in the same assignment.

* Show complete p-value of number of attempt on table 1. * Show all results from all assignment in table 2. Then talks about it. We need to convince people of our statistical results. * At last, talks about the insignificant final scores.

2 Our System

The adaptive hint system is built on top of Open edX, an open source MOOC platform. We used Open edX as our homework system and embedded our adaptive hint system. The homework system allows students to answer problems by entering math expressions and provides instant feedback. We assign one assignment to students each week, and each assignment contains 10 to 15 math expression problems. These 10 to 15 problems in an assignment all target the same material that is covered in lecture. Students have an unlimited number of attempts for each problem, they can spend as many attempts as they want until they get to the correct answer.

Our adaptive hint system is built by creating hints ahead of time-based on historical data. After we release the homework, the hint system will start to parse student attempts to figure out students' confusion. Then it looks for the appropriate hint in the hint database and sends to students.

Create Hints

Hints are created before the homework problems are released. This allows the system to have a pool of hints to start with at the time the assignment is released. To create useful hints, we need to analyze the incorrect student attempts from previous quarters. We asked TAs to create hints based on the mistakes students made in previous quarters. The hints we created are composed by a short explanation following by a short question for students to answer. For example, if a student type in $3 * 10 - 50$, where the correct answer should be $3 * 10 - 50!$, the system will first look at the last part of the expression and what it stands for. Say this student types in 50 instead of 50! for the number of ways to arrange 50 different poker cards, the hint question will then be "You need to find the number of ways to arrange 50 different poker cards. How many ways can you arrange 3 different poker cards?" This way we can encourage students to think about the simpler question and try to answer it. We don't want to give students the correct answer right away, instead, we want to guide them by giving them short and simple questions to solve. Note that students can choose to answer or not answer these hint questions. If students choose to answer the hint questions, we will provide instant feedback to let the students know whether their answers are correct or not. If students choose not to answer the hint questions, it won't hurt their grade.

Parse Trees

The adaptive hint system needs to parse student attempts. Since we are targeting math expression problems, we can use parse trees to parse the math expressions. The parse tree always has operators as tree parents and operands as children. See Fig. 1. Each tree node is marked by its position in the tree. The root node is node R . The left child node is indicated by 0 and right child is 1. Therefore, the left tree node of node R is $R.0$, and right node is $R.1$.

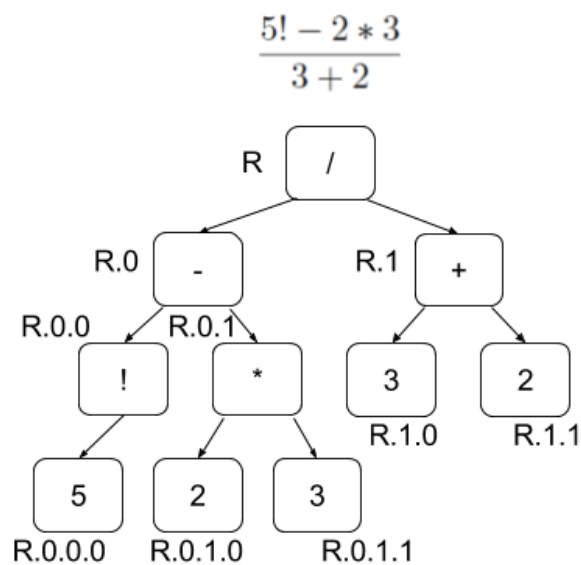


Figure 1: The parse tree mark each tree node by position. "R" stands for root node, "0" stands for left child node and "1" stands for right child node. Therefore, "R.0.1" stands for the right child node of the root's left child.

To figure out the mistake in an attempt, we compare the parse tree of the attempt with the parse tree of the correct answer. The tree comparison can tell us the subexpression that students make mistakes on. We can then classify student inputs based on the incorrect subexpressions and assign hints correspondingly. For example, the student who typed $5! - 2 * 3$ and the student who typed $\frac{5! - 2 * 3}{10}$ for a problem with solution $\frac{5! - 2 * 3}{3 + 2}$, as in Fig. 1, would be classified as the same group. Both of the students will have a mismatch of tree node $R.1$. This tells the system that the subtree of $R.1$ is wrong and the system will look for the corresponding hint.

One problem is that there are many different ways to write the same math expression. For example, $5!$ can also be written as $5 * 4 * 3 * 2 * 1$ or $5 * 4 * 3 * 2$. Therefore, we evaluate each subtree and use it in our comparison. The parse trees are evaluated bottom up. The operators at the bottom of the trees are evaluated first. The root operator is evaluated last. See Fig. 2.

Both the parse tree and the evaluation tree are used when we compare the attempt to the correct answer. We identify a subtree as incorrect only when the evaluation and the subtree itself don't match. Therefore, $5!$ and $5 * 4 * 3 * 2 * 1$ will be identified as the same subtree because it evaluates to the same result.

Assign Hints

When TAs create hints, they are also asked to specify rules. Hint rules tell the system when to send certain hints and who the hints will go to. One example of hint rule for expression in Fig. 1 could be "R.0.0 is wrong" and the corresponding hint could be "You need to find the number of ways to arrange 5 poker cards. How many ways can you arrange three different cards?". When the hint system captures an attempt with an incorrect $R.0.0$ subtree, this hint will be sent. Such attempts

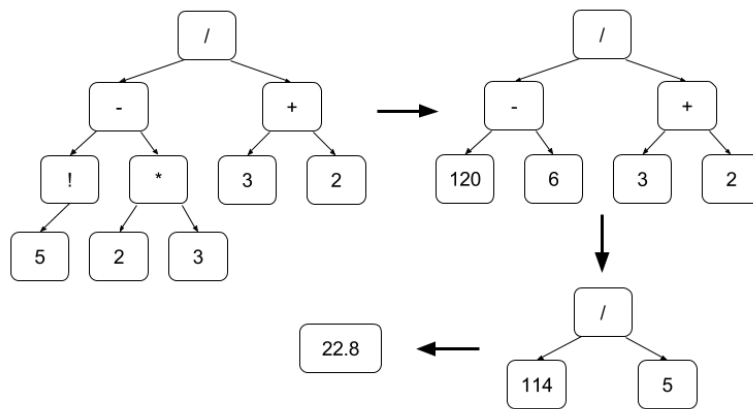


Figure 2: The parse tree is evaluated bottom up. We first evaluate the lowest level of the tree then pass the evaluated results to the level above. Then use the lower level results to evaluate the upper level of the tree until we reach the top node of the tree, the root, which will give us the final numerical result of the expression.

could be $\frac{4!-2*3}{3+2}$ or $\frac{30-2*3}{3+2}$. The adaptive hint system will assign hints automatically based on the hint rules. This allows our hints to be sent automatically.

Note that the system doesn't send hints to every student. We only send hints to students who have been working on the problem for more than 5 minutes or students who did more than three attempts. We think these students are the ones who are working on the problem actively and are struggling. Furthermore, students need to demand for hints by clicking a "Show Hints" button. Once a hint is assigned to a student, the student can choose to click or not click the "Show Hint" button. One can ignore the assigned hint by not clicking the "Show Hint" button. If a student wants to see the assigned hint, he/she needs to click the "Show Hint" button to see the hint. See Fig. 3 In other words, we only show hints to students who ask for hints. Students who don't click the "Show Hints" button will not see the assigned hints. This makes sure we give hints to students who need helps and actively ask for help.

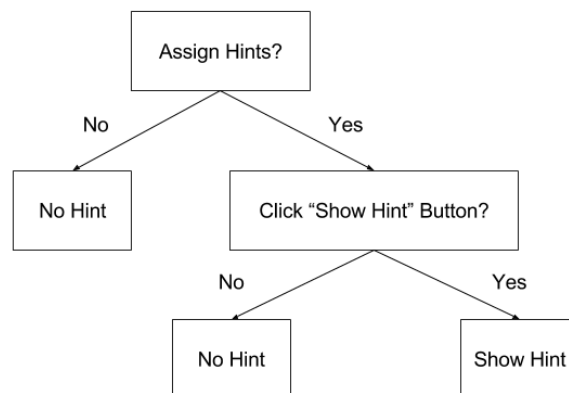


Figure 3: First the system decides whether to assign hints to a student. Then the student gets to decide whether to see the assigned hint. A student will see a hint only if the system decides to assign the hint and also the student decides to see the hint.

3 Statistical Study on Effectiveness of hints

The hint system has recorded all the student attempts and the timestamps of each attempt. This allows us to extract useful features, including the number of attempts on each problem, time spent on each problem, etc. These features are used to study what effects hints make on students behavior.

To study the effectiveness of the hint system, we randomly split students into two groups for each assignment, one group with hints(the hint group) and one group without hint(the no-hint group). Each student has 50% chance of getting a hint on each assignment. If a student is chosen to receive hints on one assignment, he/she will be able to receive hints on all the problems in that assignment. Note that students can still choose not to see the hint on certain problems by not clicking the "Show Hint" button. Therefore, for the students in the hint group, we have two different types of problems. One is **problems with hints**, these are the problems that students decided to click "Show Hint" to see the hint content. We also have **problems without hint**, these are problems that students decided not to click the "Show Hint" button. Also, for each problem, we have three groups of students. We first have **no-hint group**, the group of students receives no hint on the problem. We also have **hint group**, the group of students receive hints on the problem and clicked "Show Hint" to see the hint content. Finally, we have **no-show group** the group of students receive hints on the problem but decided not to see the hints by not clicking the "Show Hint" button.

3.1 Effect on Problems with Hints

We first look at problems with hints. These problems have at least one hint sent out to at least one student. We extracted the number of attempts students spent on these problems. We calculated the number of attempts for each problem average over students in the no-hint group, and also the number of attempts for each problem average over students in the hint group. We compared these two numbers for each problem and found that students without hint spend fewer attempts on a problem compared to students with hints. See the left graph in Fig. 4. This shows that students in the hint group are students who struggle with these problems.

We also extracted the time students spent on each problem. We measured the time differences between each attempt and filtered those that are longer than 10 minutes. We consider the time gaps larger than 10 minutes as students taking a break. We then sum all the time gaps and consider the sum as the total time students spent on the homework problems. We get a similar result comparing to the number of attempts experiment. See the graph on the right of Fig. 4. The students with hints spent more time working on problems than the students with no hint. The result is consistent with the number of attempts measurement.

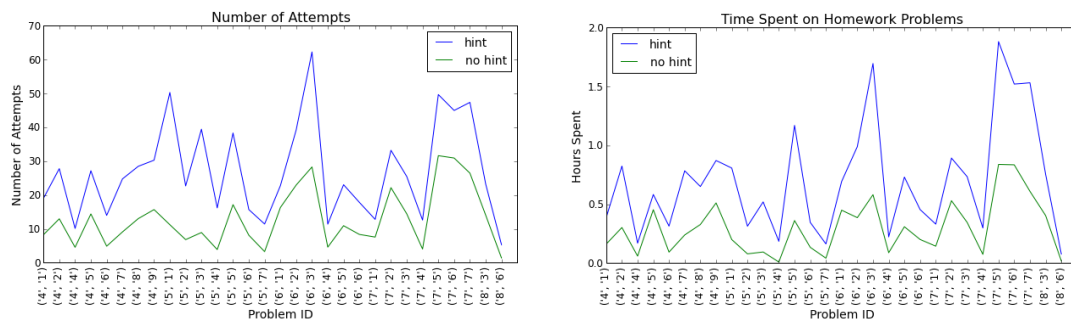


Figure 4: On the left, the graph shows number of attempts for each problem average over all students. These problems are problems that at least one student received hints on. It shows that students who ask for hints spent more attempts on these problems than the students in the no-hint group. On the right is the graph of time spent on each problem average over all students in hint group vs. students in no-hint group. Students with hints spend more time on problems than the students in no-hint group. It is consistent with the graph of number of attempts.

Furthermore, we want to examine whether these two groups of students are significantly different. We performed a two-tailed t-test on each problem for both the number of attempts and the time spent.

Problem ID	P-Value	
	time spent	number of attempts
(4, 5)	0.099	0.000
(4, 6)	0.064	0.008
(4, 8)	0.246	0.073

Table 1: P-Value of two tailed t-test for time spent on problems and number of attempts spent on problems. The null hypothesis is that students in hint group and students in no-hint group spend the same amount of time on each problem. Here we listed 3 out of 30 problems we examined. The other 27 problems all have p-value smaller than 5%.

We found that 23 out of 30 problems have p-value smaller than 0.1%. See Table 1. Therefore, we conclude that students who ask for hints are students who struggle. They spend more time and more attempts on these problems before they reach the correct solution.

Note that students with hints are likely to spend more time on these problems than the students with no hint because all our hints are short questions designed for students to answer. Students will spend time to do the hint questions in between the attempts on the homework problems.

3.2 Effect on Problems without Hint

We then look at problems without hints. Our goal is to select the problems with no hint and examine whether adaptive hints help students learn to solve similar problems when no hint is present.

We filtered out the problems that students attempted before the first hint in each homework assignment. We make sure to only look at problems that students solve after they see their first hint. This makes sure students received at least one hint before solving these problems. For example, student A does an assignment of 9 problems in the following order $\{1, 4, 3, 2, 6, 5, 7, 8, 9\}$. Student A receive hints on problem $\{3, 5, 7\}$. We will then only look at problem $\{2, 6, 8, 9\}$ in this assignment. We made sure that student A does problem $\{2, 6, 8, 9\}$ after receiving at least one hint in this assignment.

To measure the difference between the hint group and the no-hint group, we first take all the students in the no-hint group and calculate the average number of attempts t_i on each problem i . We use t_i as our baseline for each problem i . Then for each student in the hint group, we find the problems without hints and sum up all the attempts s_i on these problems. For previous example of student A, we would have $S_A = s_2 + s_6 + s_8 + s_9$. We would also have the sum of number of attempts from our baseline $T_A = t_2 + t_6 + t_8 + t_9$. We then compare S_A and T_A to see whether they are the same. See Figure. 5.

Average Number of Attempts of Student A	Baseline Attempts	Problems (Order in Student A's attempts)
15	18	1
25	22	4
20	17	3 ← hint
23	29	2
30	35	6
18	13	5 ← hint
15	14	7 ← hint
23	24	8
9	11	9
$22 + 30 + 23 + 9 = 84$	$29 + 35 + 24 + 11 = 99$	Sum of Attempts on Problems w/o Hint

Figure 5: We filter out the problems that come before the first problem with hints. We only look at problems that come after problem 3 in the above example. We then sum up the attempts and compare the sum of attempts for student A and our baseline.

Assignment	Number of attempts of (no-hint group) - (hint group)	P-Value
4	3.626	0.414
5	6.541	0.237
6	8.695	0.019
7	8.906	0.085
All Assignments	11.697	0.056

Table 2: The table listed the difference of average number of attempts between hint group and no-hint group. We first compare the attempts for each assignment, then we look at all assignments at the end. We also perform a two tailed t-test with the null hypothesis that students in no-hint group do the same amount of attempts on problems as students in the hint group. The p-value of the two tailed t-test is small enough that we can reject the null hypothesis.

Our data shows that students in the hint group spend an average of 11.697 fewer attempts than the students in the no-hint group over all assignments. We again perform a two-tailed t-test. Our null hypothesis of our t-test is that students in the hint group should have the similar sum of attempts with students in the no-hint group. The p-value of our t-test is 5.64%. Therefore, we reject the null hypothesis and conclude that student performance in the hint group and the no-hint group are significantly different. Therefore, we concluded that the adaptive hint system helps students to learn to solve similar problems that come later in the assignment. See Table 2.

This result shows us that the adaptive hints teach students how to solve problems instead of giving out students the solution to a specific problem. Students who learn from hints perform better on similar problems comparing to students without hints.

4 Conclusion

5 Statistical Analysis

5.1 Effect on Final Score

Final scores play an important role as to evaluate the learning outcome of students. We examine the relationship between the number of hints one receives and their final score.

The problems in the final exam aim to examine students understanding of different topics learned throughout the college quarter. Each of these topics have a collection of homework problems for students to practice. Therefore, we can find a set of homework problems correspond to each of the problem in final exam. We then created a mapping show in Table 3.

Final Problem	Homework Problem IDs
3	(4,1), (4,2)
4	(5,1), (5,2), (5,3), (5,4)
5	(6,4)
6	(6,6)
7	(6,2), (6,3)
8	(8,3), (8,4), (8,5), (8,6)

Table 3: Each final problem on the left has a set of corresponding homework problems on the right.

We first look at students in hint group vs. students in no-hint group based on the problem mapping. For each problem, students who receive no hint in any of the corresponding homework problems belong to the no-hint group. Students who receive at least one hint in any of the problem belong to hint group. We compare students' final score for each final problem. See Figure 6. We also compare students' final performance for each homework assignment. We reverse the mapping in table 3 and look at each homework assignment and the corresponding final problems. For example, homework assignment 6 will have corresponding final problems {5, 6, 7}. We will therefore look at the average final scores for all these three problems. See Figure 6.

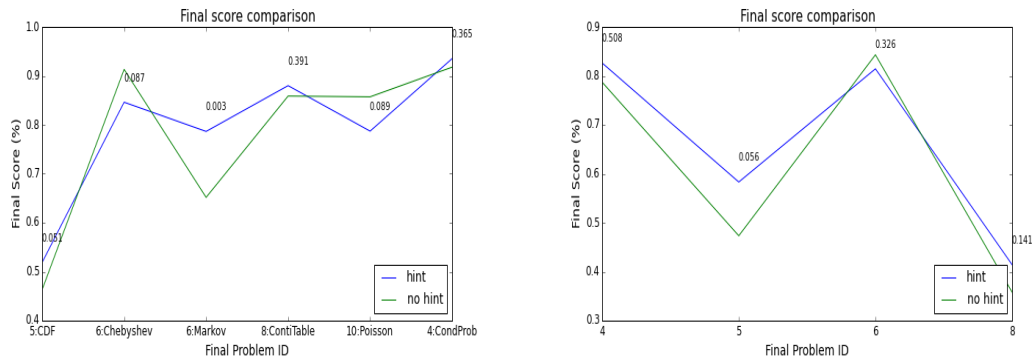


Figure 6: The graph on the left is the plot of students' final scores of each final problem. The graph on the right is the plot of students' average final scores of each homework assignment. The data points are labeled with p-value of a two-tailed t-test.

We then split students in to two groups, strong group and weak group. Students in strong group are students who have final grade above average. Students in the weak group are students with final grade below average. We first look at each final problem. We compare the final score for each problem between students in hint group and students in no-hint group. Table 4 shows the number of students in each group. We then compare final grades in Figure 7.

We also look at each assignment. We compare the final score for each homework assignment between students in hint group and students in no-hint group. Table 5 shows the number of students in each group. We then compare final grades in Figure 8.

Group(Problem)	No hint assigned	See hints	Ignore hints
Strong(P1)	72	11	54
Strong(P2)	16	107	14
Strong(P3)	66	36	35
Strong(P4)	66	15	56
Strong(P5)	31	70	36
Strong(P6)	41	61	35
Weak(P1)	41	28	39
Weak(P2)	22	73	13
Weak(P3)	60	24	24
Weak(P4)	60	16	32
Weak(P5)	31	58	19
Weak(P6)	43	35	30

Table 4: The table listed the number of students in strong group and weak group on each problem in final. It listed number of students received no hint, number of students received and saw hints, and number of students received but didn't see any hint.

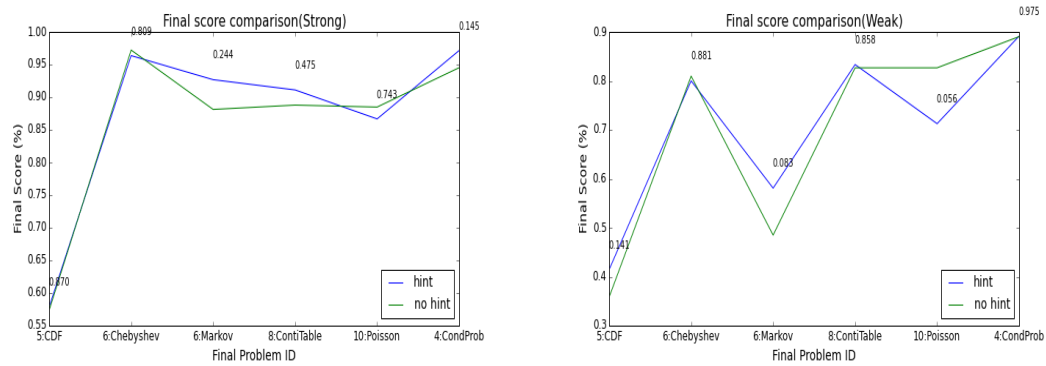


Figure 7: We compare the average of students' final scores of each final problem in strong and weak group. Each data point is labeled with the corresponding p-value of a two-tailed t-test.

References

- [1] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–207, 1995.
- [2] R. K. Atkinson, S. J. Derry, A. Renkl, and D. Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70(2):181–214, 2000.
- [3] R. Azevedo and R. M. Bernard. A meta-analysis of the effects of feedback in computer-based instruction. *Journal of Educational Computing Research*, 13(2):111–127, 1995.
- [4] R. S. Baker, A. T. Corbett, and K. R. Koedinger. *Detecting Student Misuse of Intelligent Tutoring Systems*, pages 531–540. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [5] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: When students "game the system". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 383–390, New York, NY, USA, 2004. ACM.
- [6] R. S. Baker, I. Roll, A. T. Corbett, and K. R. Koedinger. Do performance goals lead students to game the system? In *Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology*, pages 57–64, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.
- [7] R. L. Bangert-Drowns, C.-L. C. Kulik, J. A. Kulik, and M. Morgan. The instructional effect of feedback in test-like events. *Review of Educational Research*, 61(2):213–238, 1991.

Group(Assignment)	No hint assigned	See hints	Ignore hints
Strong(4)	31	57	20
Strong(5)	20	81	7
Strong(6)	31	65	12
Strong(8)	43	35	30
Weak(4)	45	63	29
Weak(5)	16	110	11
Weak(6)	31	85	21
Weak(8)	41	61	35

Table 5: The table listed the number of students in strong group and weak group on each homework assignment. It listed number of students received no hint, number of students received and saw hints, and number of students received but didn't see any hint.

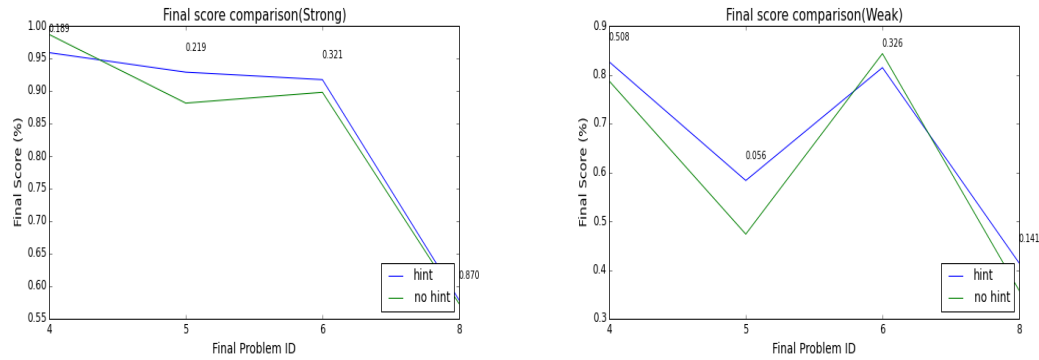


Figure 8: We compare the average of students' final scores of each homework assignment in strong and weak group. Each data point is labeled with the corresponding p-value of a two-tailed t-test.

- [8] H. Cooper, J. C. Robinson, and E. A. Patall. Does homework improve academic achievement? a synthesis of research, 19872003. *Review of Educational Research*, 76(1):1–62, 2006.
- [9] B. J. Mason and R. Bruning. Providing feedback in computer-based instruction: What the research tells us. 15, 01 2001.
- [10] K. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. 8:30–43, 02 1997.
- [11] R. Kulhavy and R. Anderson. Delay-retention effect with multiple-choice tests. *Journal of Educational Psychology*, 63(5):505–512, 10 1972.
- [12] B. M. McLaren, S.-J. Lim, F. Gagnon, D. Yaron, and K. R. Koedinger. *Studying the Effects of Personalized Language and Worked Examples in the Context of a Web-Based Intelligent Tutor*, pages 318–328. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [13] J. Mestre, D. M. Hart, K. A. Rath, and R. Dufresne. The effect of web-based homework on test performance in large enrollment introductory physics courses. *Journal of Computers in Mathematics and Science Teaching*, 21(3):229–251, 2002.
- [14] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis*, 36(2):127–144, 2014.
- [15] L. Razzaq and N. T. Heffernan. *Hints: Is It Better to Give or Wait to Be Asked?*, pages 349–358. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [16] V. J. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008.
- [17] K. Vanlehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. 46:197–221, 10 2011.
- [18] K. Vanlehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The andes physics tutoring system: Lessons learned. *Int. J. Artif. Intell. Ed.*, 15(3):147–204, Aug. 2005.