Hint Generation Under Uncertainty: The Effect of Hint Quality on Help-Seeking Behavior

Thomas W. Price, Rui Zhi, and Tiffany Barnes

North Carolina State University, Raleigh, NC 27606, USA
{twprice, rzhi, tmbarnes}@ncsu.edu

Abstract. Much research in Intelligent Tutoring Systems has explored how to provide on-demand hints, how they should be used, and what effect they have on student learning and performance. Most of this work relies on hints created by experts and assumes that all help provided by the tutor is correct and of high quality. However, hints may not all be of equal value, especially in open-ended problem solving domains, where context is important. This work argues that hint quality, especially when using data-driven hint generation techniques, is inherently uncertain. We investigate the impact of hint quality on students' help-seeking behavior in an open-ended programming environment with on-demand hints. Our results suggest that the quality of the first few hints on an assignment is positively associated with future hint use on the same assignment. Initial hint quality also correlates with possible help abuse. These results have important implications for hint design and generation.

Keywords: Intelligent Tutoring Systems, Hints, Help-seeking, Programming

1 Introduction and Related Work

A hallmark of Intelligent Tutoring Systems (ITSs) is their ability to provide next-step help to students while they work on problems. A variety of work has explored how such help can be generated [6, 14], how it should be ideally used [2], and its impact on student performance and learning [7, 16]. However, this body of work generally makes the assumption that all help provided by the tutor is equally useful and of high-quality. While some work distinguishes between levels of hints (e.g. pointing, teaching and bottom-out hints), each level represents "correct" advice, and is therefore assumed to be useful.

In this paper, we argue that this assumption is not always valid.Increasingly, ITSs employ data-driven hint generation to reduce the need for expensive expert modelling and to target domains where such modelling is difficult [12, 14]. While data-driven hints can be designed to meet quality criteria, (e.g. leading to a valid solution), this does not necessarily mean that every hint is useful to every student. In fact, a comparison of various hint generators found high variance in their agreement with expert hints [11]. Even hints engineered by experts are still subject to the "expert blindspot," where expert domain knowledge fails to translate into pedagogically useful feedback. Further, domains such as open-ended programming are so complex that even the best hints may not account for the diversity of possible solution paths a student might pursue [12]. Outside of ITSs, researchers have found that feedback has widely varying effects on task performance, often negative, depending on a multitude of factors [9], so we might expect similar variability in ITSs.

Because hint quality in open-ended problem-solving domains can be uncertain, it is important to understand its impact on students. In this work, we investigate the effect of hint quality on students' help-seeking behavior in an open-ended programming environment with on-demand hints. Help-seeking plays an important role in student learning in ITSs, and students frequently fail to make good use of help facilities [2]. We present evidence that hint quality does affect both positive and negative help-seeking behaviors.

1.1 Next-Step Hints in ITSs

ITSs traditionally offer next-step hints as support for individual steps within a problem-solving task. Hints can be provided on-demand or in response to an error, immediately or after a delay. ITSs may present multiple levels of a hint, with increasing specificity, and most include some form of "bottom-out" hint, in which the tutor tells the student exactly how to proceed. Hints in ITSs have been historically grounded in a number of theories. Under the ACT-R model [5], hints can turn an otherwise unsolvable problem into a useful worked example. Under a Vygotskian model, hints play a scaffolding role, bridging the gap between a student's knowledge and the requirements of the problem, and keeping the student in the Zone of Proximal Development (ZPD) [10].

Traditionally, next-step hints are generated using an expert model, such as the Model Tracing found in cognitive tutors (e.g. [7]). By modelling the problem domain, the tutor can identify errors in the student's work and suggest next steps, generally using templated messages. More recently, data-driven hint generation techniques such as the Hint Factory [6] have shown that hints can be generated without an expert model using previous students' attempts at a given problem. Data-driven hint generation was first applied to logic proofs [16] and has since seen a number of applications in open-ended programming [11, 12, 14].

Empirical evaluations of the effect of next-step hints on student learning have shown positive results. Stamper et al. [16] compared students working with and without on-demand hints in the Deep Thought logic tutor across two semesters and found that the hint group completed more of the tutor, had less dropout and had higher final course grades than the control group. Corbett et al. [7] compared a variety of help mechanisms in the ACT Programming Tutor, including on-demand hints, and found that students who received any type of feedback during tutoring completed a subsequent programming assessment in significantly less time, with significantly fewer errors. Piech et al. [11] evaluated a number of hint generation policies in the domain of programming based on their agreement with "gold-standard" hints, authored by human experts, and found that their best hint policy achieved 95.9% and 84.6% agreement with the gold standard hints on two assignments. Aleven et al. [3] point out that some of these studies are not truly experimental, and hints may play a less substantial role in learning than previously thought, especially compared to other interventions, such as support for self-explanation.

1.2 Help-Seeking in ITSs

Help-seeking is a self-regulatory skill that plays an important role in learning. It pertains to a student's ability to appropriately seek out and use available help resources when problem solving. In ITSs, researchers have noted that students generally display poor help-seeking behaviors [4]. For example, Aleven and Koedinger [1] found that students using the PACT Geometry Tutor focused primarily on bottom-out hints, skipping through other hint levels on 82%-89% of steps. Wood and Wood [18] studied students working in the QUADRATIC learning environment and found that those with lower prior knowledge were more likely to seek help and benefit from it. However, in a later study of a different system in which they used a pretest to select challenging problems for each student, they found these differences disappeared [17], suggesting that the effect is due to subjective problem difficulty, rather than prior knowledge alone.

Guided by theoretical perspectives and empirical evidence, Aleven et al. developed a model of desired student help-seeking behavior in a cognitive tutor and implemented the Help Tutor, itself a cognitive tutor, to teach the skill of help-seeking [2]. Their model also defined help-seeking bugs, including Help Abuse, in which students misuse or overuse help, and Help Avoidance, in which a student could benefit from help but chooses not to request it. In an empirical evaluation, they found that help-seeking errors negatively correlated with domain learning, that the Help Tutor reduced the incidence of some of these errors, but that it had no impact on domain learning [15]. Aleven et al. also offer useful reviews of help-seeking research in interactive learning environments [2–4].

2 iSnap

The dataset analyzed in this paper comes from students using iSnap [13], an extension of Snap*!* [8], a block-based programming environment for novices. iSnap augments Snap*!* with many features of an ITS, including on-demand, data-driven hints, generated using the CTD algorithm [12]. Students request hints by clicking a Help button, after which their code is annotated with multiple hint buttons, each corresponding to a contextual hint generated based on their current code. Hovering over a button highlights the code to which the hint applies.

When a hint button is clicked, all buttons disappear and the student is shown a next-step hint window, which generally suggests a single edit to the student's code, such as inserting, deleting or reordering a code element. The hints are presented visually, as shown in Figure 1, demonstrating the edit to make. These are similar to "bottom-out" hints, in that they tell the student exactly what to do, but they do omit some details, such as variable names and literal values.

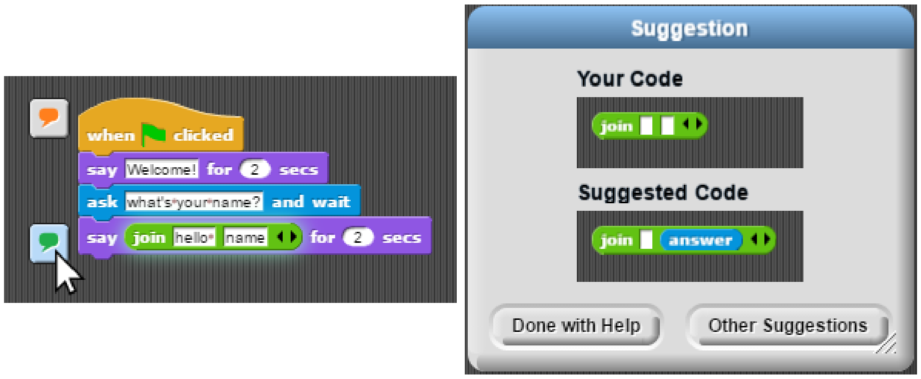

Fig. 1. A student's code is annotated with hint buttons (left). Upon clicking a button, a hint window suggests a change to the student's code (right).

Students can continue to work with the hint window showing, or they can dismiss it, optionally choosing to re-show the hint buttons. Students can click through multiple hints this way, searching for one that suits their needs.

3 Methods

This study was conducted during an introductory computing course for non-majors, consisting of 68 students, held at a research university during the Fall 2016 semester. During the first half of the course, the lab sections taught the Snap*!* programming language through a curriculum based on the Beauty and Joy of Computing (BJC) [8]. The course included three in-lab programming assignments, which were completed in class with help available from teaching assistants, interleaved with three homework assignments, which were completed independently. Students completed all work in iSnap, and data-driven hints were available on the 2nd, 3rd, 4th and 5th assignments, shown in Table 1. Hints were generated using data from the Spring 2016 semester.

Before the first hint-enabled assignment, a researcher briefly introduced the hint interface, explaining that the students were encouraged to use hints without any penalty to their grades and that the hints may not be perfect. iSnap reminded students that help was available at the start of each assignment. There was no limit on the number of hints a student could request. iSnap recorded complete logs of hint requests and snapshots of students' code after each edit. Students did not login to use iSnap, so unfortunately we cannot analyze a given student's work over multiple assignments due the anonymous logs.

Using the log data, we identified 642 hint requests (when a hint button was clicked, as in Figure 1) across the four hint-enabled assignments. For each hint, we calculated how long the hint window was viewed and how long after viewing it the student made their next code edit. Because each hint suggested a specific edit to the student's code, we were also able to label each hint as "followed" if the

Table 1. For each assignment with hints available, the number of attempts submitted (N), the number of attempts with one or more hint requests (H), the mean grade with SD, and the agreement (κ) for the graders.

Assignment	**Type**	**N**	**H**	**Mean Grade (SD)**	**Grader κ**
Polygon Maker	In-lab	65	11 (17%)	97.5% (9.0%)	0.62
Squiral	HW	60	18 (30%)	78.6% (23.0%)	0.79
Guessing Game 1	In-lab	66	13 (20%)	95.2% (10.8%)	0.81
Guessing Game 2	HW	59	13 (22%)	89.8% (17.8%)	0.63

student's code reflected the suggestion within their next 5 edits. Some students viewed a single hint multiple times in a short time span. For example, a student might browse through multiple hints and then return to one to implement it. We considered two hints duplicates for a given student if they suggested the exact same edit, and the student viewed the two hints within 60 seconds, or without changing their code in between. We merged duplicate hint requests to produce 542 final hint requests, which we used in our analysis. We considered these merged hints to be "followed" if any of their component hints were followed.

3.1 Submission Grading and Hint Rating

For each assignment, we identified 5-6 assignment objectives based on the instructions and the course instructor's grading rubric. Two researchers independently graded each submission, marking each objective as complete or incomplete. The graders discussed disagreements to produce a final grade for each submission, calculated as the percent of objectives completed. Interrater reliability is given for each assignment by Cohen's kappa (κ) in Table 1.

We also developed a detailed hint rating rubric to quantify the quality of iSnap hints.[1] The hint rubric has 3 attributes, each rated 1, 2 or 3, with higher scores being better: *Relevance*, how likely the hint is to address one of the student's current goals; *Progress*, how well the hint moves the student's current code to a correct solution without removing useful code; and *Interpretability*, how easily a novice could understand the intention and value of the hint.

Across all four assignments, a total of 55 assignment attempts included at least one hint request. These may include attempts from the same student on multiple assignments. Of these 55 attempts, 39 (70.9%) included a second hint request, and 29 (52.7%) included a third request. We limited our analysis to the first 2 hints in an attempt, since barely half of the relevant attempts included a third hint request. This resulted in a subset of 94 hints that were selected for rating. Two authors independently rated these hints on each attribute. The raters had access to a snapshot of the student's code when the hint was shown. Disagreements were discussed and resolved to produce final ratings. The hint

[1] Complete rubric and dataset available at: http://go.ncsu.edu/aied2017-rubric.

raters achieved squared-weighted Cohen's kappas of 0.857, 0.756 and 0.685 for Relevance, Progress and Interpretability, respectively. This indicates substantial agreement, and while we make no claims that the ratings are objectively "correct," the multiple raters helped to ensure consistent ratings across hints.

4 Analysis

We structured our analysis around the following four research questions, which are addressed in the following subsections:

RQ1 How frequently did students use and follow hints in iSnap?

RQ2 How did hint use relate to student performance on assignments?

RQ3 How did a given hint's quality affect whether it was followed?

RQ4 How did the quality of students' first hints affect their future help use?

4.1 Hint Use in iSnap

With 542 unique hint requests over 250 assignment attempts, students received on average 2.2 hints per assignment. This average is misleading, however, since the number of hints requested per student varied widely. The percentage of students who requested at least one hint on a given assignment ranged from 16.9% to 30.0%, with somewhat higher usage on homework assignments. Despite being introduced to the hint system in class and receiving reminders from iSnap before starting work, the majority of students never used the help on a given assignment. This is consistent with studies of help-seeking in other domains, where hint-usage rates were low. For example, Aleven and Koedinger found hints were used on 22-29% of steps in the PACT Geometry Tutor [1].

Across assignments, 55 attempts (22%) included at least one hint request. Many of these attempts included only 1-2 hint requests (47.3%), but those with 3 or more requests had a median of 16 hint requests per attempt, with a maximum of 68. Of all hints requested, 41.3% were followed. An additional 16.4% of hints were unfollowed but occurred within 1 minute of a later hint that *was* followed, indicating that a majority (57.5%) of help requests were closely followed by some resolution. For attempts on any assignment which included at least 1 hint request, we compared the number of hints requested with the percent of hints followed. There is a significant, positive Spearman correlation between the two values ($\rho = 0.552$; $S = 12415$; $p < 0.001$), indicating that students who asked for more hints on a given assignment were also more likely to have followed them.

When a student chose to follow a hint, they took a median 9 seconds after viewing the hint before making their next edit, compared with 19 seconds for unfollowed hints. A Mann-Whitney U test[2] showed the difference was significant ($U = 18164$; $p < 0.001$). Students also viewed the hint window for hints they ended up following for less time (Med=5) than those they did not (Med=6), and the difference was significant ($U = 31550$; $p = 0.026$). This suggests that students needed less time to process a hint when they ultimately followed it.

[2] The Mann-Whitney U test was used were data were not normally distributed.

4.2 Help Use and Performance

The course from which we collected our data did not include a pre- or post-test on Snap*!* programming, so we have no measure of learning. Instead, we focus on programming performance on the assignments to evaluate the impact of hints. While this is not a substitute for learning, it does provide insight into how well the hints played their intended role of scaffolding students during programming. Due to the presence of TAs for the in-lab assignments and the high grades students achieved (see Table 1), we focus on the two homework assignments. We defined meaningful hint usage as requesting and following at least one hint, and we labeled attempts with meaningful hint usage as F1, and those without as F0.

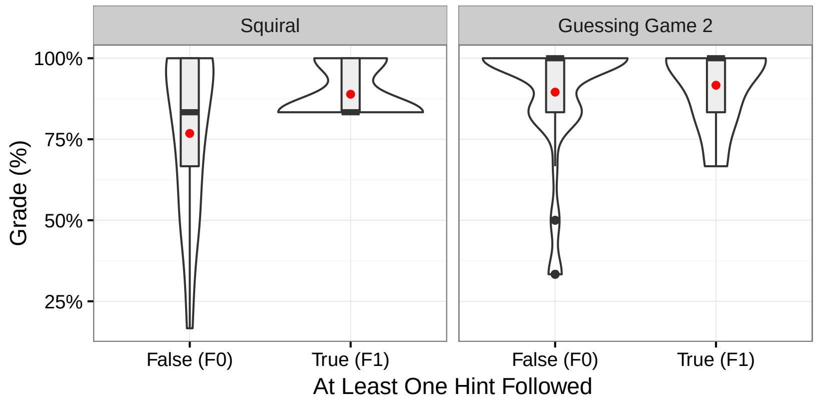

Fig. 2. Violin plots comparing grade distributions for F0 and F1 attempts at both the Squiral (left) and Guessing Game 2 (right) homeworks. Inside each plot is a boxplot, and a red dot indicating average grade.

Figure 2 shows violin plots of the grade distributions for both homework assignments, comparing F0 and F1 attempts. We can see that F1 attempts perform at least as well as their counterparts, with equal median grades and higher average grades on both assignments. This was not a controlled study, so we can make no claims about the direct effect of hints on performance. However, previous studies have found that students with lower prior knowledge are more likely to request help [18], so if the hints had no effect, we would expect those using them to perform worse, rather than at least as well.

One of the justifications for including hints on a homework assignment is that they allow students to progress when they are "stuck" and do not know how to proceed, allowing them perform adequately on an assignment. We define adequate performance as missing no more than 1 assignment objective (out of 5-6 total). By this definition, 100.0% (9/9) of F1 students performed adequately on Squiral, compared 60.8% (31/51) of F0 students. On Guessing Game 2, 87.5%

(7/8) of F1 students performed adequately, as did 88.2% (45/51) of F0 students. Put another way, homework attempts with meaningful hint usage (F1) almost always achieved adequate grades (16/17), and many of the remaining homework attempts (F0) did not achieve adequate grades (26/102). However, different homework assignments do show different impacts of hint use.

The prevalence of students with poor performance and no meaningful hint usage indicates Help Avoidance [2], where students could benefit from using help but choose not to. While there are many things we could do to encourage students to start using the help system, the remainder of this section focuses on how hint quality affects hint use, once students have started using the help.

4.3 Hint Quality and Immediate Hint Use

Our dataset of rated hints consists of 94 hints, the first and second hints requested for each assignment attempt. These were rated on three attributes from 1-3: Relevance, Progress and Interpretability, as described in Section 3.1. While these attributes were selected to be relatively independent, we first inspected the relationships among them. Each attribute pair showed a significant positive Spearman's correlation: Relevance and Progress ($\rho = 0.664$; $S = 46454$; $p < 0.001$); Progress and Interpretability ($\rho = 0.777$; $S = 30842$; $p < 0.001$); Relevance and Interpretability ($\rho = 0.632$; $S = 50919$; $p < 0.001$). Due to the high correlations, we also computed a Quality attribute, the sum of the other three, ranging from 3 to 9, and use this for most of our analysis.

An assumption that underlies our investigation is that the quality of data-driven hints is not uniform. We attempted to verify this empirically on our sample of rated hints. While the median Quality ratings for the first and second hints received were high, 7 and 8 respectively, the standard deviations were also high: 2.1 and 2.2, respectively. Out of all 94 rated hints, 30 (31.9%) were rated below the middle possible rating of 6. This suggests that the quality of our sample did vary enough to potentially impact students.

We then investigated the relationship between hint Quality ratings and whether or not a hint was followed. This was in part to confirm the validity of our ratings, since we hypothesize that valid hint ratings would be predictive of whether or not a hint was followed. However, we also note that this *is* an assumption. The theory of ideal help use put forward by Aleven et al. [2] suggests a student *should* evaluate the helpfulness of a hint before applying it, but to our knowledge this has not been investigated empirically in ITSs.

First hints had a median Quality rating of 7, and those with at least that rating were four times as likely to be followed (14/34 = 41.2%) as those with lower ratings (2/21 = 9.5%). Second hints had a median Quality rating of 8, and those with at least that rating were three times as likely to be followed (11/21 = 52.4%) as those with lower ratings (3/18 = 16.7%). Put another way, first hints that were followed had higher Quality ratings (n = 16; Med = 8) than unfollowed hints (n = 39; Med = 7), and a Mann-Whitney U test showed that the difference was significant ($U = 437.5$; $p = 0.018$). For second hints, followed

Table 2. Mean attribute ratings (with standard deviation) for first and second hints of attempts with each hint usage label. All hint ratings are summarized at the bottom.

Hint	Label	N	Relevance	Progress	Interpretability	Quality
1	Low	23	1.96 (0.88)	2.13 (0.92)	1.83 (0.72)	5.91 (2.13)
	Med.	15	2.27 (0.88)	2.47 (0.74)	1.80 (0.68)	6.53 (2.13)
	High	17	2.35 (0.70)	2.65 (0.70)	2.12 (0.86)	7.12 (1.90)
2	Low	7	1.57 (0.79)	1.86 (0.90)	1.57 (0.79)	5.00 (2.38)
	Med.	15	2.13 (0.83)	2.53 (0.74)	2.00 (0.76)	6.67 (2.06)
	High	17	2.53 (0.72)	2.71 (0.69)	2.29 (0.77)	7.53 (1.97)
1, 2	All	94	2.18 (0.83)	2.43 (0.81)	1.97 (0.77)	6.57 (2.14)

hints were also rated higher (n = 13; Med = 8) than unfollowed hints (n = 26; Med = 6.5), and the difference was significant ($U = 276.0$; $p = 0.001$).

We conclude that hint quality strongly affects the likelihood that a student will follow a hint, and that the Quality metric is reasonable. Students also have quite high standards for hints they will follow, with median 8 and 8.5 Quality ratings out of a possible 9, for followed first and second hints respectively. While the magnitudes of our ratings are somewhat relative, this is still strong evidence that students are capable of noting and disregarding flawed hints.

4.4 Hint Quality and Future Hint Use

It is encouraging that students can identify and disregard lower quality hints, but what effect does encountering such hints have on a student's future hint use? To answer this question, we labeled each attempt that used hints as Low, Medium or High hint usage, where each label corresponds to a 3-quantile of the number of hints requested for a given assignment. For example, hint requests for the Guessing Game 1 had 3-quantiles [3, 11], so the labels were $n \leq 3$ (Low), $3 < n \leq 11$ (Medium) and $11 < n$ (High), where n is the number of hints requested. Table 2 shows average attribute ratings for the first and second hints per attempt for each usage label. There is a consistent trend of ratings increasing with label, with only one exception for the Interpretability of the first hint.

There was a significant, positive Spearman correlation between an attempt's label and the Quality rating of the first hint received ($\rho = 0.267$; $S = 20322$; $p = 0.049$) and the second hint received ($\rho = 0.409$; $S = 5838$; $p = 0.010$). Additionally, hint Quality ratings had a significant positive Spearman correlation with the percentage of future hints that were *followed* for second hints ($\rho = 0.413$; $S = 2384$; $p = 0.026$) but not first hints ($\rho = 0.201$; $S = 7892$; $p = 0.219$). Together, these findings strongly suggest that better initial hints encourage students to make more use of hints in the future. Another way to interpret this result is that students can somewhat easily be deterred from using hints if they encounter one that does not seem useful. Looking only at the first 2 hints in an attempt, students stop asking for hints after receiving a hint that is at least median Quality only 18.2% of the time (10/55), while they stop 41.0% of the time after lower Quality hints (16/39), over twice as often.

5 Discussion

RQ1 *How frequently did students use and follow hints in iSnap?* Consistent with prior results [1], we found that help use in iSnap was quite low overall. Few assignment attempts included help requests (17-30% across assignments), and many attempts which had help requests only had one or two hints (38-64%). A quarter of student homework attempts missed multiple objectives, but few of these had meaningful hint usage, suggesting Help Avoidance. We could help to address this with a more detailed introduction on iSnap and how to use it effectively, or by offering proactive hints. Under half of hints were followed (41.3%), calling into question the hints' overall utility. However, this result was expected to some extent due to iSnap's UI, which provides students with buttons for all available hints at once, only some of which are likely to address the student's original reason for requesting help.

RQ2 *How did hint use relate to student performance on assignments?* We saw that students who followed at least one hint on homeworks performed at least as well as those who did not and rarely performed poorly. This supports the idea that help in iSnap can improve student performance, at least on some assignments. However, also consistent with the literature, we saw evidence of potential Help Abuse, with some students requesting over a dozen hints on one assignment. Since this behavior is negatively associated with learning [2], improved performance may not lead to improved learning for these students.

RQ3 *How did a given hint's quality affect whether it was followed?* We confirmed our assumption that hint quality does vary considerably among our data-driven hints. We also found empirical evidence to support the intuitive notion that students are more likely to follow higher-quality hints. We saw that the bar is quite high, with followed first and second hints having medium Quality ratings of 8 and 8.5 out of 9. This suggests that anything less than a great hint will go unused more often than not. That has serious implications for the design of data-driven hints, emphasizing the need for a vetting mechanism to ensure lower quality hints are not shown, perhaps using data on which hints have gone unfollowed in the past. However, just as there are many factors that influence a student's willingness to *seek* help in an ITS [4], we should also consider contextual factors besides hint quality that might encourage or discourage students from *following* requested hints, such as student affect.

RQ4 *How did the quality of students' first hints affect their future help use?* The impact of hint quality appears to continue on to later hints, with a positive correlation between the Quality ratings of the first two hints a student receives and the student's later level of hint usage. While we cannot speak directly to the mechanism at work here, it seems likely that a student's interactions with hints establish the student's level of trust in the system, which impacts future hint use. The first few interactions with help may be especially important, though we did not investigate later hints to verify this. Interestingly, we see that higher initial hint quality is predictive of High hint use, which could indicate Help Abuse. One might hypothesize that students who abuse help are pre-disposed to do so, due to individual factors like prior knowledge and motivation; however, our results suggest that trust in the ITS could also play a role. We currently take no steps to prevent abuse of help in iSnap, but our results suggest this may be necessary to ensure that the majority of a completed assignment is written by the student.

6 Conclusion

In this paper we have presented evidence that in practice, data-driven hints in an ITS for programming vary in quality, which has both an immediate impact on whether students follow hints, and a long-term impact on how many hints they request. This work is novel in that it directly measures hint quality using expert ratings and investigates the impact of hint quality on students. One hypothesis that would explain this impact is that hint quality affects a student's trust in the ITS, and trust affects future help use. This has important implications for the design of ITSs, whether or not they rely on data, since all intelligent systems operate under some degree of uncertainty. Designers should carefully craft ITS help to avoid negative interactions with students, which might mean choosing *not* to offer help in uncertain situations.

This work has several limitations. Our dataset included only one programming course and four assignments, and we cannot claim that the results will generalize to other domains. iSnap differs from traditional ITSs (e.g. Cognitive Tutors) in its lack of a student model and the open-ended nature of the task it supports. It also offers multiple hints at once, giving students some control over the hints they receive, so it is possible student-specific factors played a confounding role in our analysis. Due to the small number of attempts with hint requests, we analyzed hints from different assignments together, but our results in Section 4.2 suggest that the assignment may impact how hints affect the student. Additionally, our hint ratings were made by experts, not students, and may be subject to the "expert blindspot." Our analysis was exploratory, and our primary goal was to generate hypotheses, not conclusions. We therefore chose not to correct for multiple significance tests, but this means our results should be interpreted cautiously until they can be confirmed.

While the data analysis we present here is an important first step, future work should employ more detailed methods, such as collecting think-aloud and interview data, to understand how hint quality impacts students' help-seeking behavior, including the mechanisms that translate hint quality into hint usage. While Aleven and colleagues' model of *ideal* help-seeking behavior in a Cognitive Tutor is useful [2], we should also work towards developing a model of how students seek help *in practice*, to design systems which can leverage these behaviors for better learning.

References

1. Aleven, V., Koedinger, K.: Limitations of student control: do students know when they need help? In: Int. Conf. on Intelligent Tutoring Systems. pp. 292–303 (2000)
2. Aleven, V., Mclaren, B., Roll, I., Koedinger, K.: Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. Int. J. of Artificial Intelligence in Education 16(2), 101–128 (2006)
3. Aleven, V., Roll, I., Mclaren, B.M., Koedinger, K.R.: Help Helps, But Only So Much: Research on Help Seeking with Intelligent Tutoring Systems. Int. J. of Artificial Intelligence in Education 26(1), 1–19 (2016)
4. Aleven, V., Stahl, E., Schworm, S., Fischer, F., Wallace, R.: Help Seeking and Help Design in Interactive Learning Environments Vincent. Review of Educational Research 73(3), 277–320 (2003)
5. Anderson, J.R.: Rules of the Mind. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1993)
6. Barnes, T., Stamper, J.: Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In: Int. Conf. on Intelligent Tutoring Systems. pp. 373–382 (2008)
7. Corbett, A., Anderson, J.: Locus of Feedback Control in Computer-Based Tutoring: Impact on Learning Rate, Achievement and Attitudes. In: SIGCHI Conference on Human Computer Interaction. pp. 245–252 (2001)
8. Garcia, D., Harvey, B., Barnes, T.: The Beauty and Joy of Computing. ACM Inroads 6(4), 71–79 (2015)
9. Kluger, A.N., Denisi, A.: The Effects of Feedback Interventions on Performance: A Historical Review, a Meta-Analysis, and a Preliminary Feedback Intervention Theory. Psychological Bulletin 119(2), 254–284 (1996)
10. Luckin, R., Du Boulay, B.: Ecolab: The Development and Evaluation of a Vygotskian Design Framework. Int. J. of Artificial Intelligence in Education 10, 198–220 (1999)
11. Piech, C., Sahami, M., Huang, J., Guibas, L.: Autonomously Generating Hints by Inferring Problem Solving Policies. In: ACM Conf. on Learning @ Scale. pp. 1–10 (2015)
12. Price, T.W., Dong, Y., Barnes, T.: Generating Data-driven Hints for Open-ended Programming. In: Int. Conf. on Educational Data Mining (2016)
13. Price, T.W., Dong, Y., Lipovac, D.: iSnap: Towards Intelligent Tutoring in Novice Programming Environments. In: ACM Technical Symposium on Computer Science Education (2017)
14. Rivers, K., Koedinger, K.R.: Data-Driven Hint Generation in Vast Solution Spaces: a Self-Improving Python Programming Tutor. International Journal of Artificial Intelligence in Education 16(1) (2015)
15. Roll, I., Aleven, V., McLaren, B.M., Ryu, E., Baker, R.S.J.D., Koedinger, K.R.: The help tutor: Does metacognitive feedback improve students' help-seeking actions, skills and learning? In: Int. Conf. on Intelligent Tutoring Systems. pp. 360–369 (2006)
16. Stamper, J., Eagle, M., Barnes, T., Croy, M.: Experimental Evaluation of Automatic Hint Generation for a Logic Tutor. Int. J. of Artificial Intelligence in Education 22(1), 3–17 (2013)
17. Wood, D.: Scaffolding, contingent tutoring and computer-supported learning. Int. J. of Artificial Intelligence in Education 12, 280–292 (2001)
18. Wood, H., Wood, D.: Help seeking, learning and contingent tutoring. Computers & Education 33(2-3), 153–169 (1999)