Data-Driven Feedback Beyond Next-Step Hints

Michael Eagle
North Carolina State University
890 Oval Dr, Campus Box 8206
Raleigh, NC 27695-8206
Maiku.Eagle@gmail.com

Tiffany Barnes
North Carolina State University
890 Oval Dr, Campus Box 8206
Raleigh, NC 27695-8206
Tiffany.Barnes@gmail.com

ABSTRACT

Intelligent tutors have been shown to be as effective as human tutors in supporting learning in many domains. Although they can be very effective, the construction of intelligent tutors can be costly. One way to address this problem is to use previously collected data to generate domain models to provide intelligent feedback to otherwise non-personalized tutors. These data-driven methods for providing next-step hints have been successful in providing feedback to students in procedural problem solving tutors. We seek to expand on next-step hints with other data-driven methods. We outline three different interventions, all of which can be generated using previously collected student data.

1. INTRODUCTION

Intelligent tutors have been shown to be as effective as human tutors in supporting learning in many domains, in part because of their individualized, immediate feedback, enabled by expert systems that diagnose student's knowledge states [13]. For example, students provided with intelligent feedback in the LISP tutor spent 30% less time and performed 43% better on post-tests when compared to other methods of teaching [1]. Similarly, Eagle, and Barnes showed that students with access to hints in the Deep Thought logic tutor spent 38% less time per problem and completed 19% more problems than the control group [4]. In another study on the same data, Stamper, Eagle, and Barnes showed that students without hints were 3.6 times more likely to drop out and discontinue using the tutor [12].

Procedural problem solving is an important skill in STEM (science, technology, engineering, and math) fields. Open-ended procedural problem solving, where steps are well-defined, but can be combined in many ways, can encourage higher-level learning [2]. However, understanding learning in open-ended problems, particularly when students choose whether or not to perform them, can be challenging. The Deep Thought tutor allows students to use logic rules in different ways and in different orders to solve 13 logic proof problems for homework.

Although they can be very effective, the construction of intelligent tutors can be costly, requiring content experts and pedagogical experts to work with tutor developers to identify the skills students are applying and the associated feedback to deliver [9]. One way to reduce the costs of building tutoring systems is to build data-driven approaches to generate feedback during tutor problem-solving. Barnes and Stamper built the Hint Factory to use student problem-solving data for automatic hint generation in a propositional logic tutor [10]. Fossati at el. implemented Hint Factory in the iList tutor to teach students about linked lists[7]. Evaluation of the automatically generated hints from Hint Factory showed an increase in student performance and retention [12].

Hint Factory creates hints by modeling previously collected student data into a Markov Decision Process and generating a next step policy, when students request a hint they are directed to the best next step. For this work, we are interested in looking into ways to expand the feedback offered to students beyond these next-step hints. We have outlined three different interventions, all of which can be generated using previously collected student data.

2. THE DEEP THOUGHT LOGIC TUTOR

In Deep Thought propositional logic tutor problems, students apply logic rules to prove a given conclusion using a given set of premises. Deep Thought allows students to work both forward and backwards to solve logic problems [3]. Working backwards allows a student to propose ways the conclusion could be reached. For example, given the conclusion B, the student could propose that B was derived using Modus Ponens (MP) on two new, unjustified (i.e. not yet proven) propositions: $A \to B, A$. This is like a conditional proof in that, if the student can justify $A \to B$ and A, then the proof is solved. At any time, the student can work backwards from any unjustified components (marked with a ?), or forwards from any derived statements or the premises. Figure 1 contains an example of working forwards and backwards with in Deep Thought.

3. DATA-DRIVEN FEEDBACK

In this section we will outline three different data-driven methods that we can use to provide hints to students. These methods are all intended to be used in conjunction with the next-step hints that have already been shown as successful.

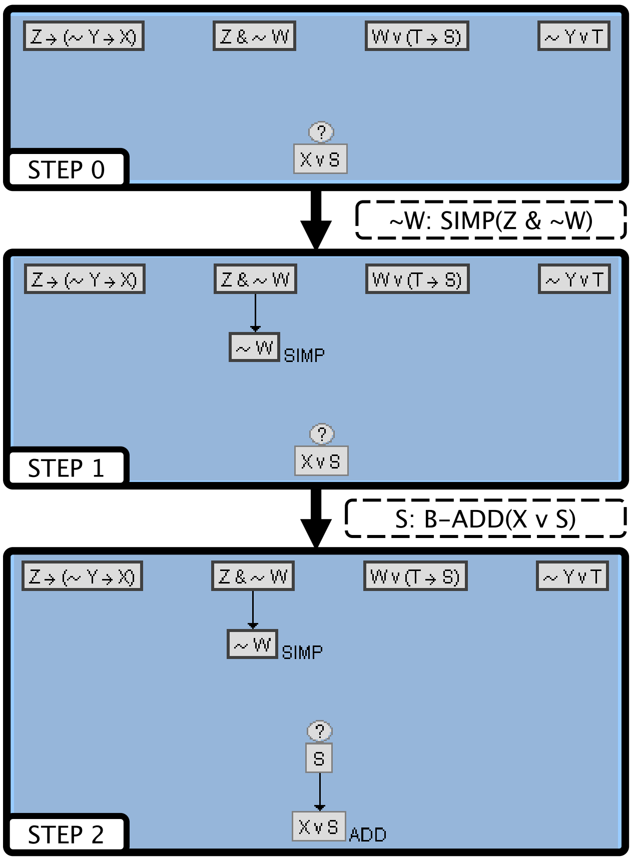

Figure 1: This example shows two steps within the Deep Thought tutor. First, the student has selected $Z \wedge \neg W$ and performed Simplification (SIMP) to derive $\neg W$. Second, the student selects $X \vee S$ and performs backward Addition to derive S.

3.1 High Level Hints

Interaction Networks describe sequences of student-tutor interactions [5]. Interaction networks form the basis of the data-driven domain model for automatic step-based hint generation by the Hint Factory. Eagle et al. applyied Girvan-Newman clustering to interaction networks to determine whether the resulting clusters might be useful for more high-level hint generation [5]. Stamper et al. demonstrated the differences in problem solving between two groups by coloring the edges between Girvan-Newman clusters of interaction networks based on the frequencies between two groups, revealing a qualitative difference in attempt paths [12]. Eagle and Barnes expanded this work into Approach Maps [6], which summarize interaction networks into the higher-level approaches used by students to solve the proofs.

In order to encouage student planning we can use the higher-level approaches discovered with Approach Maps to provide sub-goals to the students. In figure 2 we show a mock up of how the sub-goal could be presented to the student. We hypothesize that these hints will help students learn which parts of the proof to focus on in order to complete problems.

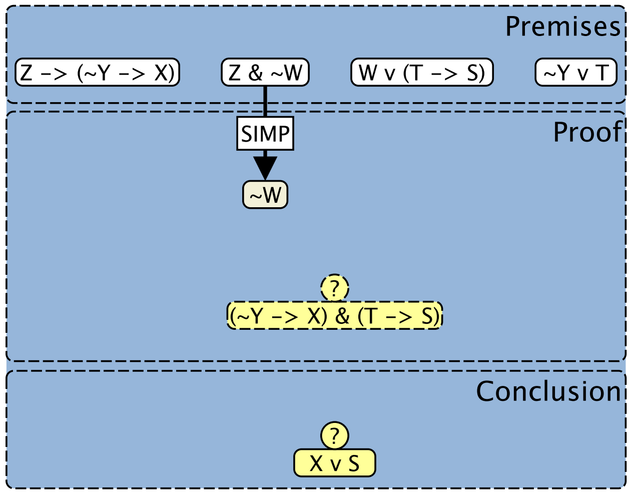

Figure 2: Example of a high level hint. DT offers the student a sub-goal based on commonly derived steps from previously collected data.

3.2 Hazard Hints

Stamper et al. in [12] and Eagle et al. in [5] found evidence that students would sometimes spend a lot of time in approaches that were unlikely to result in a solution. This discovery is important as interventions can be added to warn away from regions that do not lead to goals. For example, we could offer a message that warns them that most students who attempt the same type of proof are not successful. Fossati et al. showed that human tutors helping students with the iList tutor, suggest that students delete unproductive steps [7]. In figure 3, we show an example interface for a *hazard hint*. These types of hints would be offered whenever a student was performing a task that was unlikely to result in a successful proof, with the goal of reducing the amount of "wasted" time.

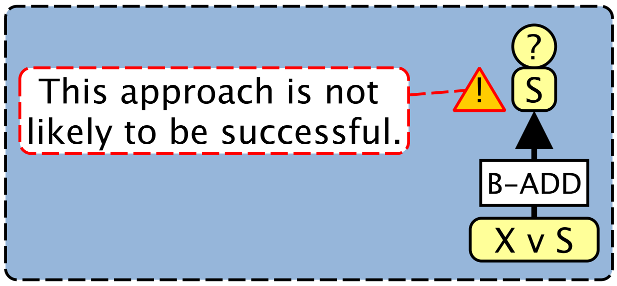

Figure 3: We can warn a student when they approach the problem in a way that is not productive.

3.3 Time Hints

Eagle, and Barnes used survival analysis to model student time-in-tutor and student dropout[4]. Survival analysis is a series of statistical techniques that deal with the modeling of time to event data [8]. It derived its name from its start within medical literature. Survival analysis is also known as reliability analysis or duration analysis.

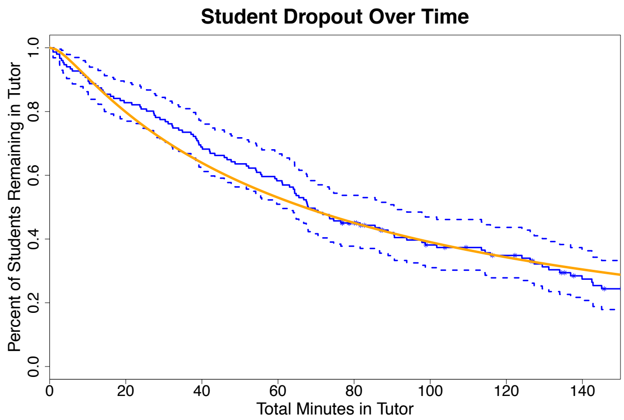

Figure 4: The Kaplan-Meier survival estimation and corresponding 95% confidence intervals show the percent of students remaining in tutor over time. The lighter (orange) line is the AFT model produced from the same data.

We start by first plotting the Kaplan-Meier survival estimator, see figure 4, which is represented as a series of declining steps which is intended to approach the true survival function. We perform our experiments on the Spring and Fall 2009 Deep Thought logic tutor dataset as analyzed by Stamper, Eagle, and Barnes in 2011[11]. We look specifically at 151 students who stopped using the tutor before completing all of the questions required for the homework assignment. Application of the AFT model provides us with coefficients of the model had the intercept (mean) as 4.20 and the SD (scale) as 1.44. The median of the survival function, the location where 50% of people have dropped out of the tutor, is found by $e^{\mu} = e^{4.20} = 66.89$, meaning that half of the students had dropped out after about an hour of tutor interactions. We have plotted the resulting survival curve in figure 4.

We hypothesize that we can prevent dropout by providing feedback when students reach certain thresholds of time within the tutor. To test this we will build survival models based on past student data, using these models we will provide feedback in the form of a pop-up window that will encourage the student, as well as provide them with resources if they are struggling. We can augment these models with information about the students current tutor performance, to get an idea of how likely the student is to complete the tutor. Overall, the use of survival modeling will provide us with more accurate representations of student time-in-tutor, and we can use this information to create interventions that will reduce the number of students who quit the tutor without finishing. In figure 5 we show an example of the type of prompt we can offer a student if our time model shows that the student is in danger of quitting the tutor.

3.4 Evaluation

Data-driven methods for offering next-step hints have been successful. We have outlined three new ways to offer feedback based on previously collected data that can be added in addition to next-step hints. In order to test the effectiveness of these forms of feedback we will seek to repeat studies like

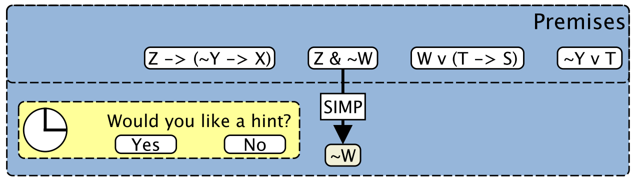

Figure 5: We can remind the student about hints if student is taking longer than predicted.

Stamper, Eagle, and Barnes' 2011 Hint Factory study [11].

4. REFERENCES

[1] J. R. Anderson and B. J. Reiser. The lisp tutor. *Byte*, 10(4):159–175, 1985.

[2] B. S. Bloom. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Taxonomy of educational objectives: the classification of educational goals. Longman Group, New York, 1956.

[3] M. J. Croy. Problem solving, working backwards, and graphic proof representation. *Teaching Philosophy*, 23:169–188, 2000.

[4] M. Eagle and T. Barnes. Survival analysis on duration data in intelligent tutors. In *Intelligent Tutoring Systems*, Honolulu, Hawaii, 2014.

[5] M. Eagle, M. Johnson, and T. Barnes. Interaction Networks: Generating High Level Hints Based on Network Community Clustering. In *Educational Data Mining (EDM2012)*.

[6] M. Eagle and B. Tiffany. Exploring differences in problem solving with data-driven approach maps. *Educational Data Mining (EDM 2014)*, 2014.

[7] D. Fossati, B. Di Eugenio, S. Ohlsson, C. Brown, L. Chen, and D. Cosejo. I learn from you, you learn from me: How to make ilist learn from students, 2009.

[8] D. W. Hosmer, S. Lemeshow, and S. May. *Applied Survival Analysis: Regression Modeling of Time to Event Data*. Wiley-Interscience, New York, NY, USA, 2nd edition, 2008.

[9] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

[10] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. *Educational Data Mining (EDM 2008)*, pages 197–201, 2008.

[11] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. In *Artificial Intelligence in Education*, AIED'11, pages 345–352, Berlin, Heidelberg, 2011. Springer-Verlag.

[12] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education (IJAIED)*, 22(1):3–18, 2012.

[13] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.