

Research Type: Design and Development Research

Research Title: Data-driven Support for Novice Programmers

Authors: Thomas W. Price, Tiffany Barnes

Contact Information: twprice@ncsu.edu; tmbarnes@ncsu.edu

Purpose

The goal of this project is to create and evaluate algorithms for generating data-driven hints and feedback for novice programmers working on open-ended assignments. Our research will result in an augmented novice programming environment that provides automated, customized feedback to students on their current program when they are stuck and unable to proceed on their own. This should allow students to continue working, even in the absence of an instructor or TA. We will evaluate our success by measuring the impact of our hints on novice students' performance and learning in an introductory CS setting.

Justification

CS instruction is arguably more effective now than ever, with new curricula that engage students' creativity and interests and programming environments that make CS more accessible through block-based editors [6]. However, even when we engage students and lessen the burden of syntax, programming remains a difficult subject to learn. When students struggle or get stuck, the burden falls almost entirely on instructors, who are not always available.

Intelligent Tutoring Systems (ITS) are adaptive learning tools that attempt to fill this gap by playing the role of a tutor, guiding a student's learning. ITSs offer support during problem solving, often with adaptive hints and feedback. Like many tutor interventions, these hints can be seen as scaffolding to keep the student in the Zone of Proximal Development [3]. Empirically, this approach has been shown to improve student performance both inside the tutor and on subsequent assessments [5].

Our work attempts to make these ITS-style hints available in existing, effective novice programming environments and curricula. Our data-driven algorithms automate the process of hint generation, lowering the barrier to use. We aim to improve on the state of the art in hint generation by designing an algorithm specifically for the open-ended programming assignments used in many introductory courses, which pose difficulties for existing techniques [2].

Research Plan

Our methods for achieving our research goals are as follows:

- 1) Instrument a programming environment (Snap!) and collect data from novices in a class setting as they work on introductory assignments. [Complete]
- 2) Design and iteratively refine an algorithm which adapts existing techniques for data-driven hint generation to the context of open-ended programming assignments. [Complete]
- 3) Perform a technical evaluation of the algorithm to determine the feasibility of our approach using historical data to generate and test hints. [Complete]

- 4) Run a small pilot study in a CS classroom and collect qualitative data on how students interact with available hints on a single assignment. [Complete]
- 5) Revise the algorithm to address the problems observed in Step 4.
- 6) Run a larger pilot study over a semester in a CS class, and compare student outcomes to historical data collected in Step 1.

The population we are studying is an introductory CS course for non-majors, consisting of 60-80 students per semester. In Step 6, our measures will be students' performance on assignments with hints available, their performance on a subsequent assignment without hints available and their performance on in-class assessments.

Findings

Our research so far has yielded three primary findings:

Existing hint-generation techniques are not well suited for open-ended programming assignments [2]. The data collected during the fall and spring semesters suggests that there is very little exact overlap among student solution, making it very difficult to apply traditional data-driven techniques. Additionally, open-ended assignments often have objectives that involve user interaction or visual output, making them difficult to assess automatically, which many existing techniques rely upon.

Despite these challenges, we can reliably generate data-driven hints [1]. A technical evaluation of the CTD algorithm shows that even on an open-ended assignment with almost no direct student overlap, the algorithm is still able to reliably generate hints which lead a student to a complete or nearly-complete solution.

These hints have potential to provide great benefit, but there are many challenges to successful implementation. Qualitative analysis of the pilot study reveals that the hints perform ideally for some students, but many use them rarely or not at all. Some of the generated hints were technically correct but still confusing to students. Other hints were quite reasonable but still ignored, indicating the need for further refinement of the algorithm and further study of what makes a good hint.

Publications and Links to Relevant Webpages

Demo page: <http://go.ncsu.edu/isnap>

Publications:

- [1] T. W. Price, Y. Dong, and T. Barnes, "Generating Data-driven Hints for Open-ended Programming," in *Proceedings of the International Conference on Educational Data Mining*, 2016.
- [2] T. W. Price and T. Barnes, "An Exploration of Data-Driven Hint Generation in an Open-Ended Programming Problem," in *Proceedings of the Workshop on Graph-Based Data Mining held at EDM'15*, 2015.

Other References:

- [3] T. Murray and I. Arroyo, "Toward Measuring and Maintaining the Zone of Proximal Development in Adaptive Instructional Systems," in *International Conference on Intelligent Tutoring Systems*, 2002.
- [4] D. Garcia, B. Harvey, and T. Barnes, "The Beauty and Joy of Computing," *ACM Inroads*, vol. 6, no. 4, pp. 71–79, 2015.
- [5] A. Corbett and J. Anderson, "Locus of Feedback Control in Computer-Based Tutoring: Impact on Learning Rate, Achievement and Attitudes," in *Proceedings of the SIGCHI Conference on Human Computer Interaction*, 2001, pp. 245–252.
- [6] T. W. Price and T. Barnes, "An Exploration of Data-Driven Hint Generation in an Open-Ended Programming Problem," in *Proceedings of the Workshop on Graph-Based Data Mining held at EDM'15*, 2015.