Toward Improving Effectiveness of Crowdsourced, On-Demand Assistance From Educators in Online Learning Platforms

Aaron Haim[0000−0002−9287−4201], Ethan Prihar[0000−0002−5216−9815], and Neil T. Heffernan[0000−0002−3280−288X]

Worcester Polytechnic Institute, Worcester, MA 01609, USA

Abstract. Studies have proven that providing **on-demand assistance**, additional instruction on a problem when a student requests it, improves student learning in online learning environments. Additionally, crowdsourced, on-demand assistance generated from educators in the field is also effective. However, when provided on-demand assistance in these studies, students received assistance using **problem-based randomization**, where each condition represents a different assistance, for every problem encountered. As such, claims about a given educator's effectiveness are provided on a per-assistance basis and not easily generalizable across all students and problems. This work aims to provide stronger claims on which educators are the most effective at generating on-demand assistance. Students will receive on-demand assistance using **educator-based randomization**, where each condition represents a different educator who has generated a piece of assistance, allowing students to be kept in the same condition over longer periods of time. Furthermore, this work also attempts to find additional benefits to providing students assistance generated by the same educator compared to a random assistance available for the given problem. All data and analysis being conducted can be found on the Open Science Foundation website[1].

Keywords: Online Education · On-Demand Assistance · Crowdsourcing.

1 Introduction

As online learning platforms expand their content base, the need to generate on-demand assistance grows alongside it [7]. Crowdsourcing provides an effective method to generate new assistance for students [7,6,11]. As on-demand assistance generally improves student learning [3,6,11,13], educators and their assistance must be evaluated to maintain or improve the current level of quality and effectiveness [10].

In 2017, ASSISTments, an online learning platform [4], deployed the Special Content System, formerly known as TeacherASSIST. The Special Content System allows educators to create on-demand assistance for problems they assigned

[1] https://osf.io/zcbjx/

to their students. On-demand assistance was known as ***student-supports***, most commonly provided in the form of hints and explanations. Additionally, educators marked as *star-educators* had their *student-supports* provided to students outside their class for any problem the class's educator did not generated a *student-support* for.

While studies analyzed the effectiveness of educators who generated *student-supports* [10] using problem-based randomization, students learn cumulatively across problems [5], making it difficult to provide substantial claims on overall effectiveness in the platform. The first part of this work will develop and use an educator-based randomization, where all *star-educators* are ordered randomly for each student with a *student-support* provided from the top-most educator in the ordering who has generated a *student-support* for a problem, in place of problem-based randomization [7], where a *student-support* was provided randomly from the available *student-supports* for a problem, within the Special Content System to determine an educator's effectiveness.

Since an educator-based randomization will prevent students from receiving certain educators over the first study, benefits from other educators for a student may be unknown. A student may be put in an educator-based randomization where a certain *student-support's* effectiveness is poor compared to other *student-supports* on the problem. The second part of this work will develop an use a **reverse educator-based randomization**: a student uses the reverse order of educators from the first part of this work with a *student-support* provided from the bottom-most educator in the ordering who has generated a *student-support* for a problem.

Other benefits of educator-based randomization compared to problem-based randomization may also be revealed through additional analysis. After this work has collected the necessary data and determines which educators are the most effective, a comparison between previous measures of effectiveness across *student-supports* and educators will be conducted.

In summary, this work aims to answer the following research questions:

1. Which educators are the most effective at generating *student-supports*?
2. How did the effectiveness of the given educator ordering compare to reversed ordering?
3. Was there any hidden benefits from receiving educator-based randomization compared to problem-based randomization?

2 Background

In this work, ASSISTments will be used to conduct the studies. ASSISTments[2] is a free, online learning platform providing feedback and insights on students to better inform educators for classroom instruction [4]. ASSISTments provides problems and assignments from open source curricula, the majority of which is K-12 mathematics, which teachers can select and assign to their students.

[2] https://assistments.org/

Students complete assigned assignments within the ASSISTments Tutor. For most problem types, students receive immediate feedback when a response is submitted for a problem, which tells the student whether the answer is correct [2]. When a *student-support* has been written by the assigning educator or a *star-teacher* for a problem, a student can request to receive the *student-support* at any time while completing the problem. *Student-supports* may come in the form of hints which explain how to solve parts of the problems [3,11], similar problem examples [6], erroneous examples[11,1], and full solutions to the problems [12].

By using the Special Content System, it found that delivering *student-supports* to students compared to immediately giving students the answer caused more student learning [7]. In addition, an analysis was conducted which reported evidence about which educators were generally more effective at improving student learning compared to other educators [10]. Those studies used problem-based randomization. The Special Content System will be modified to provide *student-supports* using educator-based randomization to investigate their effectiveness.

3 Methodology

This work will collect data over the course of the three months. During this time period, two studies each lasting a month will run a different selection mechanism. In between the two studies and after the final study has ended, there will be a two week interval where the selection mechanism will use problem-based randomization. These weeks will be treated as the dependent measure to determine a student's performance within the educator-based randomization.

The Special Content System will use the selection mechanism outlined in Table 1 during the associated time period. After the work has completed collecting data, the Special Content System will be restored to its original state before this work.

Name	Time Period	Selection Mechanism
Initial Data	Before Time Period	Problem-Based Randomization
Study 1	1 Month	Educator-Based Randomization
Mid-Test	2 Weeks	Problem-Based Randomization
Study 2	1 Month	Reversed Educator-Based Randomization
Post-Test	2 Weeks	Problem-Based Randomization

Table 1. Breakdown of Work Conducted

3.1 Study 1: Educator Ordered Selection

Study 1 will run over the period of a month. During this study, every student will be given a randomly ordered list of all available *star-educators* within the ASSISTments platform. If an educator has a *student-support* written for a problem

(Table 2 gives an example on the left where Educator A has a *student-support* for Problem Y while Educator B does not), the the student will be provided that educator's *student-support*. Otherwise, the next educator will be chosen to provide a *student-support* and so on until either an educator has written a *student-support* for the given problem or no educators have written a *student-support* (in which case none is provided). Using the example in Table 2, if Student 1 requested a *student-support* for Problem Y, the selection mechanism would determine that student would receive a *student-support* from Educator A. In contrast, Student 2 would receive a *student-support* from Educator B for Problem Y, as Educator C did not write a *student-support* and the next educator in the list, Educator B, has.

	Problem X	Problem Y
Educator A	Yes	Yes
Educator B	Yes	Yes
Educator C	Yes	No
Educator D	Yes	No

	Student 1	Student 2	Student 3
Top	Educator A	Educator C	Educator D
	Educator B	Educator B	Educator A
	Educator C	Educator D	Educator B
Bottom	Educator D	Educator A	Educator C

Table 2. An example of Educator Ordering data. **Left**: shows what educators wrote a *student-support* for certain problems where "Yes" means a educator wrote a *student-support* for a problem and vice versa for "No". **Right**: shows an ordering of all available educators (in this example) for each student from top to bottom.

Benefits of an Educator Ordering Since the ASSISTments platform is used to produce this work, providing each student an ordering of all available *star-educators* is favored over a single educator to better create educator-based randomization. *Student-supports* have been shown to improve student learning [7,10]; if an single educator has not written a *student-support* for a problem which other educators have, the application should still provide an available *student-support*. This is a common occurrence as nineteen *star-educators* have collectively generated 38,737 *student-supports*; however, the top five generated up over 50% with the top two generated approximately 37.6% of the available *student-supports*.

To validate the effectiveness of an educator ordering over a single educator, the ASSISTments Dataset [8,9] was used to simulate Study 1. There are 4,094,728 logged interactions where a *student-support* was selected for a given student on a problem. After pre-processing the data such that only interactions where a student has completed another problem after the current one and more than one *student-support* was available for selection, there are 2,226,779 logged interactions across 94,040 unique students.

As shown on the left of Figure 1, about 90% of the students almost never received their top-most educator in the ordering, instead on average around 12.8% of the time. Those students would never receive a *student-support* if only

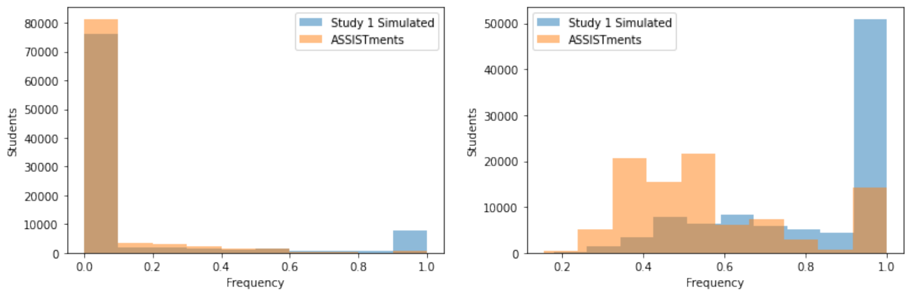

Fig. 1. A comparison of a simulated Study 1 compared to problem-based randomization method used by the ASSISTments platform. Shows the frequency students received their top ordered educator (left) and the frequency students received the educator which they were provided the most *student-supports* from (right).

single educator solution was used, which would stymie our ability to improve student learning. On the right of Figure 1, when using an educator ordering, more than 50% of the students nearly always received their most provided educator with the average around 82.4%. As such, an educator ordering is more effective at keeping students in an educator-based randomization while still maintaining improve learning standards within the ASSISTments platform.

3.2 Study 2: Reversed Educator Ordered Selection

Study 2 will run for a month following a two week interval after Study 1. Students will be provided a *student-support* from the lowest-most educator in the ordering determined from Study 1 who has written a *student-support* for the given problem. In the Table 2 example on the left, Student 1 will receive Educator D's *student-supports* first when available, then C's, then B's, then finally A's. As such, Student 1 will receive the *student-support* generated by Educator C for Problem Y while Student 2 will the *student-support* generated by Educator A.

3.3 Analysis Plan

As the data is currently under collection, no analysis has been formalized yet. Instead, a random 10% of the collected data will be used to attempt different modeling approaches and reduce noise. Afterwards the exact analysis method will be formalized and use the remaining 90% of the data.

Acknowledgements We would like to thank the NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), and Schmidt Futures. None of the opinions expressed here are that of the funders. We are funded under an NHI grant (R44GM146483) with Teachly as a SBIR.

References

1. Adams, D.M., McLaren, B.M., Durkin, K., Mayer, R.E., Rittle-Johnson, B., Isotani, S., Van Velsen, M.: Using erroneous examples to improve mathematics learning with a web-based tutoring system. Computers in Human Behavior **36**, 401–411 (2014)
2. Feng, M., Heffernan, N.T.: Informing teachers live about student learning: Reporting in the assistment system. Technology Instruction Cognition and Learning **3**(1/2), 63 (2006)
3. Heffernan, N.T., Heffernan, C.L.: The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education **24**(4), 470–497 (2014)
4. Heffernan, N.T., Heffernan, C.L.: The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education **24**(4), 470–497 (2014)
5. Lee, J.: Cumulative learning and schematization in problem solving. Universität Freiburg (2012)
6. McLaren, B.M., van Gog, T., Ganoe, C., Karabinos, M., Yaron, D.: The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. Computers in Human Behavior **55**, 87–99 (2016)
7. Patikorn, T., Heffernan, N.T.: Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In: Proceedings of the Seventh ACM Conference on Learning @ Scale. p. 115–124. L@S '20, Association for Computing Machinery, New York, NY, USA (2020). `https://doi.org/10.1145/3386527.3405912`, `https://doi.org/10.1145/3386527.3405912`
8. Prihar, E., Botelho, A.F., Jakhmola, R., Heffernan, Neil T, I.: Assistments 2019-2020 school year dataset (Dec 2021). `https://doi.org/10.17605/OSF.IO/Q7ZC5`, `osf.io/q7zc5`
9. Prihar, E., Gonsalves, M.: Assistments 2020-2021 school year dataset (Nov 2021), `osf.io/7cgav`
10. Prihar, E., Patikorn, T., Botelho, A., Sales, A., Heffernan, N.: Toward personalizing students' education with crowdsourced tutoring. In: Proceedings of the Eighth ACM Conference on Learning @ Scale. p. 37–45. L@S '21, Association for Computing Machinery, New York, NY, USA (2021). `https://doi.org/10.1145/3430895.3460130`, `https://doi.org/10.1145/3430895.3460130`
11. Razzaq, L.M., Heffernan, N.T.: To tutor or not to tutor: That is the question. In: AIED. pp. 457–464 (2009)
12. Whitehill, J., Seltzer, M.: A crowdsourcing approach to collecting tutorial videos–toward personalized learning-at-scale. In: Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale. pp. 157–160 (2017)
13. Wood, D., Bruner, J.S., Ross, G.: The role of tutoring in problem solving. Child Psychology & Psychiatry & Allied Disciplines (1976)