Analyzing Fine-Grained Material Usage Behavior

Charles Koutcheme, Juho Leinonen, Juha Sorva, Arto Hellas
Aalto University
Espoo, Finland
charles.koutcheme,juho.2.leinonen,juha.sorva,arto.hellas@aalto.fi

ABSTRACT

Most prior work in log data analysis within introductory programming courses has focused on log data gathered from programming environments. While such data can provide insights into the programming process of students, it misses other parts of the learning process, such as students' use of e-textbooks or other online learning materials. In this work, we present preliminary results from an analysis of learning material use at a fine grain. We discuss the possibilities that such data provides for studying the learning process of students in introductory programming courses.

Author Keywords

educational data mining; material usage; online material; online textbook

CCS Concepts

•**Social and professional topics** → **Computing education;**

INTRODUCTION

Analyzing log data gathered from students' learning process can provide valuable information for improving the quality of introductory programming courses. In past research, when log data has been used in learning analytics for programming education, the logs have usually been gathered from programming environments [4]. One concrete proposal for how log data gathered from programming can be used to improve education is the process model for IDE-based learning analytics by Hundhausen et al. [3]. They suggest that programming instructors first gather process data, then analyze the data, design an intervention based on findings from the data, and then deploy that intervention directly to the IDE from which the data was originally gathered.

However, analyzing students' learning purely based on data gathered from programming environments misses parts of the learning process. In most introductory programming courses, students have other study materials they use besides programming assignments; online textbooks are common, for example [11, 1]. Prior studies have found that how students use online learning materials can be used to both improve the materials [7] and to predict students' success in the course [8], even when controlling for time spent [6].

DATA AND CONTEXT

Context and Participants

The data used in this study comes from an introductory programming course offered at Aalto University in Finland. The course uses an open online textbook[1] with interactive content such as visualizations, quizzes, and programming assignments. The course teaches basic programming skills and concepts and emphasizes the object-oriented paradigm. It is taken by approximately one thousand students annually. Most of the students are from Aalto University, although there are some external participants as well.

Since the individual pages in the online textbook are long, fully understanding how students use the material requires the collection of more fine-grained data than just page request logs. Additionally, while clickstream data can provide insight into how students' use materials, it requires active clicks from students and thus does not gather information about passive browsing of the textbook.

Data Collection

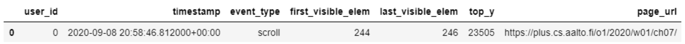
Material usage data was collected using a JavaScript component[2] that, on page load, tags HTML elements with unique identifiers and keeps track of the user's movements, storing information on which elements are visible on the user's screen at which times. The component can be added to any (statically generated) online material. A server to which the data can be sent to is required.

The component records (1) a user identifier, if available, (2) an event type, (3) a timestamp, (4) the current URL, (5) the identifier of the topmost visible element, (6) the identifier of the bottom-most visible element, and (7) the y coordinates of the topmost and bottom-most visible elements. The event types fall broadly in two categories, active and inactive, where active events depict scrolling actions and inactive events depict staying in place. The data is periodically sent to a server for analysis. An example data row is presented in Figure 1.

Seventh SPLICE Workshop at SIGCSE 2021 "CS Education Infrastructure for All III: From Ideas to Practice", SPLICE'21, March 15-16, 2021, Virtual Event

[1] The course materials from 2020 are available at `https://plus.cs.aalto.fi/o1/2020/`.

[2] Available at `https://bit.ly/3v8ZN15`.



	user_id	timestamp	event_type	first_visible_elem	last_visible_elem	top_y	page_url
0	0	2020-09-08 20:58:46.812000+00:00	scroll	244	246	23505	https://plus.cs.aalto.fi/o1/2020/w01/ch07/

Figure 1. Example record collected by the component.

High-level Analysis of Time Management

In our first analysis, we looked at how students divide their time into learning sessions. We define a learning session as a block of consecutive minutes of study. Two parameters influence the results. The first parameter is the minimum time gap, between logs, that should separate two study blocks. When a time gap is larger than the set threshold, we consider that the student is not studying. The second parameter is the minimum length of a study session. After identifying the different blocks of study, we remove the ones that are too short in length. We divide the data by week because this granularity better matches the course progression. We computed the following metrics on each week of data: the number of study sessions, the average length of a study session, and the total number of minutes studied.

DISCUSSION

Preliminary Results

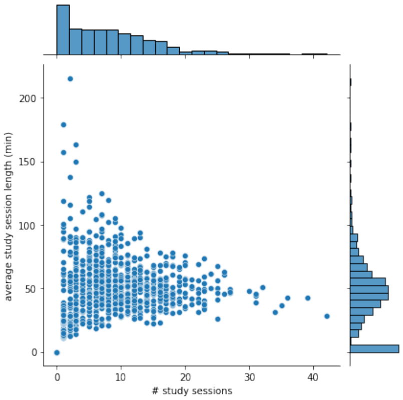


Figure 2. Joint distribution of the average length of a study session and the number of study sessions.

We performed our early analysis on the second week of data of the course. Figure 2 shows the bivariate distribution between the number of study sessions and the average study session length. Let us first look at the univariate distributions. The number of students with a given number of learning sessions decreases as the number of learning session increases. The average study session length seems to follow a normal distribution, if we ignore the proportion of students that did not study at all. What we observe as a result is that there is a large variation in the average study session lengths when the number of study sessions is small, but this variation decreases as the number of study sessions increases. In other words, there is a bigger disparity in the time allocated per study session between students who have fewer sessions (and there are more of such students), than students who have more study sessions.

Future Research

In our preliminary analysis, we looked at how students divide their time. Next, we will identify where they allocate that time. We want to track across weeks the parts of the materials that students spend most of their effort on. Such information can provide an estimate of the student's progression with respect to the course progression, and an estimate of the student's general consistency in their study habits.

We plan on modelling how students "move" within the course materials. Our data enables us to determine the student's transitions within a single page, within a chapter, and across chapters of the same or different weeks. Most online course materials are designed in a linear fashion. Instructors expect students' transitions within the materials to follow this linear progression. However, it is unknown whether this is the case. By looking at students' movements within the course we can identify their study patterns. Such patterns could be indicative of difficulties in understanding concepts that students encounter repeatedly across the materials.

This type of modelling has been shown to be effective when applied to programming snapshots within a programming assignment. Piech et al [9] represented how students tackle an assignment by developing a method that automatically builds a Hidden Markov Model of students' transitions from different milestones/steps (nodes). By analyzing (clustering) students trajectories in this network, they found patterns in how students progressed in their assignments. Subsequently, clustering these patterns into larger groups was found to be correlated with midterm performances. We aim to apply the same modelling technique with the expectation of identifying different patterns of browsing.

Improving Computer Science Education

Using machine learning techniques, we plan to leverage the time-usage information and the browsing behavior model to predict student performance. Many studies in the past showed that log data can predict student performance [2], but very few studies have employed browsing data similar to ours. In particular, we suspect that time-usage metrics that can be collected early in a course—such as students' consistency in study habits—could be discriminative enough to identify which students are likely to drop out during the first few weeks. Consequently, we aim to combine time-usage metrics with browsing patterns to predict students in need of assistance.

The patterns and information found by our analysis might also be used to improve the course materials themselves. If students do not follow the intended course progression, this might be due to some part of the materials not being clear enough. By identifying the parts of the materials where students come regularly back to, and how students come back to these materials, instructors could adapt their course to make students' progression easier [7].

Finally, we believe combining fine-grained material usage data with other data sources (such as programming logs [10] or keystroke data [5]) will expand our general understanding of students' learning process, and provide many possibilities for future research.

REFERENCES

[1] Barbara J Ericson and Bradley N Miller. 2020. Runestone: A Platform for Free, On-line, and Interactive

Ebooks. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1012–1018.

[2] Arto Hellas, Petri Ihantola, Andrew Petersen, Vangel V Ajanovski, Mirela Gutica, Timo Hynninen, Antti Knutas, Juho Leinonen, Chris Messom, and Soohyun Nam Liao. 2018. Predicting academic performance: a systematic literature review. In *Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education*. 175–199.

[3] Christopher David Hundhausen, Daniel M Olivares, and Adam S Carter. 2017. IDE-based learning analytics for computing education: a process model, critical review, and research agenda. *ACM Transactions on Computing Education (TOCE)* 17, 3 (2017), 1–26.

[4] Petri Ihantola, Arto Vihavainen, Alireza Ahadi, Matthew Butler, Jürgen Börstler, Stephen H Edwards, Essi Isohanni, Ari Korhonen, Andrew Petersen, Kelly Rivers, Miguel Angel Rubio, Judy Sheard, Bronius Skupas, Jaime Spacco, Claudia Szabo, and Daniel Toll. 2015. Educational data mining and learning analytics in programming: Literature review and case studies. *Proceedings of the 2015 ITiCSE on Working Group Reports* (2015), 41–63.

[5] Juho Leinonen. 2019. *Keystroke Data in Programming Courses*. Ph.D. Dissertation.

[6] Leo Leppänen, Juho Leinonen, Petri Ihantola, and Arto Hellas. 2017. Predicting academic success based on learning material usage. In *Proceedings of the 18th Annual Conference on Information Technology Education*. 13–18.

[7] Leo Leppanen, Juho Leinonen, Petri Ihantola, and Arto Hellas. 2017. Using and collecting fine-grained usage data to improve online learning materials. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*. IEEE, 4–12.

[8] Liang-Yi Li and Chin-Chung Tsai. 2017. Accessing online learning material: Quantitative behavior patterns and their effects on motivation and learning performance. *Computers & Education* 114 (2017), 286–297.

[9] Chris Piech, Mehran Sahami, Daphne Koller, Steve Cooper, and Paulo Blikstein. 2012. Modeling how students learn to program. *SIGCSE'12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (02 2012). DOI: **http://dx.doi.org/10.1145/2157136.2157182**

[10] Thomas W Price, David Hovemeyer, Kelly Rivers, Ge Gao, Austin Cory Bart, Ayaan M Kazerouni, Brett A Becker, Andrew Petersen, Luke Gusukuma, Stephen H Edwards, and David Babcock. 2020. Progsnap2: A flexible format for programming process data. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. 356–362.

[11] Clifford A Shaffer, Ville Karavirta, Ari Korhonen, and Thomas L Naps. 2011. Opendsa: beginning a community active-ebook project. In *Proceedings of the 11th Koli Calling International Conference on computing education research*. 112–117.