Leinonen, Juho; Castro, Francisco Enrique Vicente; Hellas, Arto

Fine-Grained Versus Coarse-Grained Data for Estimating Time-on-Task in Learning Programming

Published in:
Proceedings of The 14th International Conference on Educational Data Mining (EDM 2021)

Published: 02/07/2021

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Leinonen, J., Castro, F. E. V., & Hellas, A. (2021). Fine-Grained Versus Coarse-Grained Data for Estimating Time-on-Task in Learning Programming. In *Proceedings of The 14th International Conference on Educational Data Mining (EDM 2021)* (pp. 648-653). International Educational Data Mining Society (IEDMS). https://educationaldatamining.org/EDM2021/EDM2021Proceedings.pdf

Fine-Grained Versus Coarse-Grained Data for Estimating Time-on-Task in Learning Programming

Juho Leinonen
Aalto University
Espoo, Finland
juho.2.leinonen@aalto.fi

Francisco Enrique Vicente Castro
University of Massachusetts Amherst
Amherst, MA, USA
fcastro@cs.umass.edu

Arto Hellas
Aalto University
Espoo, Finland
arto.hellas@aalto.fi

ABSTRACT

The time that students spend on assignments, i.e. *time-on-task*, has been used frequently in prior research to understand student affect, study habits, and course performance, among others. The choice for how time-on-task is calculated, however, is typically based on available data. This data can be very coarse-grained, such as the timestamps from students' assignment submissions. Using coarse-grained data to calculate time-on-task has limitations, such as not being able to determine whether students take breaks when working on an assignment. In this work, we analyze the differences between two time-on-task metrics, one based on coarse-grained data—in this case, student submissions—and one based on fine-grained data—in this case, students' keystrokes during an assignment. We compare these two metrics and examine how well they correlate to find out whether time-on-task based on coarse-grained data can be an accurate metric for understanding the time spent by students on an assignment. Our results show that the correlation between the two metrics that are supposed to measure the same underlying phenomena—time-on-task—is only weak to moderate. This suggests that fine-grained data might be needed to accurately estimate time-on-task.

Keywords

time-on-task, fine-grained data, coarse-grained data, data granularity, keystroke data, programming process data, learning analytics, educational data mining

1. INTRODUCTION

Time-on-task—the amount of time that a student spends actively engaged in a task—is considered as one of the most important factors that contribute to learning and achievement [14, 30, 32]. Measuring time-on-task focuses on identifying *active time* that is spent on a task, instead of the overall time that includes breaks and time spent on unrelated activities. Time-on-task has been measured through various means: student self-reports [27], stopwatches [8], periodic observations [3], video recordings and eye movement data [4], and learning management system log data [17]. While all of these can be considered as proxies for time-on-task, accurately estimating time-on-task remains a challenging problem that deserves further attention [14, 17].

In this work, we study (a) to what extent two different types of log data—timestamped keystroke data and timestamped submission data from an introductory programming course—can be used to measure students' time-on-task and (b) to what extent time-on-task estimates produced with this data represent the same phenomenon. Our work is motivated by the need to distinguish between different types of data and the time-on-task estimates that can be produced with them. As numerous metrics have been used as proxies for time-on-task, if these metrics are not in line with each other, results from studies using them may not be comparable. That is, differences between observed results, or even contradictory results, could be explained to some extent by the difference in the chosen time-on-task metric.

Some studies similar to ours include work by Kovanović et al. [16] and Nguyen [23]. Kovanović et al. [16] built and compared a range of time-on-task metrics for evaluating students' performance, highlighting methodological issues. Nguyen [23], on the other hand, evaluated methods for identifying off-task behavior, also correlating the resulting estimates with academic performance. While these studies have used click-stream or event data from learning management systems such as Moodle, the data in our study comes from an introductory programming course where work on programming assignments is logged keystroke by keystroke.

This article is structured as follows. In Section 2, we discuss related time-on-task studies, starting with an overview of earlier studies on time-on-task and time-on-task estimates within learning programming, with a brief outline of studies that have analyzed different time-on-task estimates. We describe our context, data, research questions, and metrics in Section 3, and outline the analyses and results in Section 4. We discuss our findings and outline future work in Section 5.

2. RELATED WORK

2.1 Measuring Time-on-Task

Early work with time-on-task often involved on-site observations (e.g. in classrooms) where coders manually recorded

and/or timed behaviors based on a coding rubric of on-task behaviors, also linking teacher's behavior with students' behavior (e.g. [2, 10, 15]). Over time, technology advancement led to the now-prevalent practice of mining *user interaction logs* from educational software used in classrooms or technology-augmented learning activities. This has made the analysis of user logs a vital component of more recent learning and behavior studies, such as in predicting help-seeking behavior [5] or assessing performance [9]. Time-on-task studies have likewise turned to this direction. Some examples (proxies for time-on-task in parentheses) include: analyzing the relationship of gamified elements to time-on-task (number of edits) [18] and comparing the impact of different course interventions to time-on-task (online interactions with peers and accessing course materials) [25]. Of note, both in earlier and more recent time-on-task studies, are the different measures or proxies used for time-on-task, and the methods used for identifying or approximating off-task activity and breaks. These are key factors that we explore in our comparison of coarse- and fine-grained time-on-task metrics.

In research focused on time-on-task in learning to program, a conventional approach has been to log user interactions within integrated development environments (see e.g. [13, 21, 22, 24, 29]). For example, Jadud [12] used the BlueJ IDE to capture code "snapshots" (copies of source code) whenever students compiled their programs, including compiler-reported errors and related metadata. Rodrigo et al. used BlueJ logs in combination with student surveys and observations to explore relationships between novice programmers' achievement, debugging, and syntax errors [26].

Submission data has been used to estimate students' *total elapsed time*, the total time between a student's first submission and last submission. Edwards et al. noted that the difference in total elapsed time between high- and low-scoring students is only small [6]. Similarly, the *time between compilation events* has been studied previously; Jadud observed that students are likely to recompile quickly after encountering a syntax error, but spend more time working on code after a successful compilation [11]. Definitions of *work sessions* also differ between studies. For example, Fenwick et al. [7] considered a "work session" terminated when no events were logged for 60 minutes. While the previous examples demonstrate the use of time from snapshots for estimating time-on-task, other studies in programming have explored using event counts (similar to other fields) for building predictive models of student achievement. For example, Ahadi et al. [1] used assignment-specific log data that included the number of "steps" that students took to solve each assignment for predicting course outcomes.

The time-on-task metrics in these (and other studies, e.g. [20, 28, 31]), however, suffer from similar problems of failing to capture the nuances around actual working time, as even the "work sessions" may fail to account for when and how students are working offline.

2.2 Analyzing Time-on-Task Estimates

Variations on time-on-task measures across studies and research instruments make it difficult to interpret and compare findings and bring into question whether or not the different metrics are indeed measuring or evaluating similar constructs. Some researchers have begun to explore this by looking at the different ways that researchers estimate time-on-task and analyzing how these estimation choices impact conclusions drawn from these measures.

Kovanović et al. [16], for example, looked at different time-on-task estimates from learning management system data and examined the impacts of these across courses from different subject domains. Their findings suggest that strategies for time-on-task estimation can have significant effects on learning analytics models of student performance. Using data collected from an introductory programming course, Leinonen et al. [19] examined a family of time-on-task-related metrics such as self-reported study time, log-based time spent on assignments, and event counts correlated with each other as well as course exam outcomes. They noted that while similar metrics such as edit counts and event counts tended to have higher correlations, exam scores were not strongly correlated with any of the metrics, except for the number of completed assignments.

While Leinonen et al. [19] did not analyze the impact of different break durations when estimating time spent on assignments, different break durations have been studied by both Kovanović et al. [16] and Nguyen [23]. Kovanović et al. and Nguyen both used time-on-task estimates based on timestamp differences between two subsequent events in learning management systems and highlight the importance of a good time-on-task estimation strategy.

Our work builds on this prior work by looking into data from an introductory programming course, where each keystroke associated with a course assignment was recorded and timestamped. Using this fine-grained log data, we study the impact of different thresholds for measuring off-task behavior, contrasting the keystroke data with submission-based data more commonly used in studies focusing on academic achievement in learning programming.

3. METHODOLOGY

3.1 Context and Data

The data for our study comes from a 7-week introductory programming course offered at a research first university in Europe. The workload of the course is 5 ECTS, which corresponds to roughly 100 to 125 study hours. In the course, students learn the basics of procedural and object-oriented programming in Java. The course uses a *many small assignments* approach, where many of the course assignments are small, but combine to form larger programs. After working on small assignments, students are given larger assignments as well, where they practice the content and constructs that they have learned earlier.

In total, the course had 147 programming assignments. The programming assignments are worked on in an integrated development environment (IDE), that logs keystroke data for plagiarism detection and research purposes. On each keystroke, the IDE collects the current timestamp and the modification to the source code of the assignment that the student is currently working on. Keystroke data is gathered only from course assignments. Additionally, information on when students submit their assignments is collected.

Students were informed about the data gathering on the course; our analyses included data from 137 students who consented to the use of their data for research purposes and who completed at least 10 assignments in the course.

3.2 Research Questions and Metrics

Our research questions are as follows:

RQ1. How do fine- and coarse-grained time-on-task metrics differ in terms of measuring time-on-task?

RQ2. Are there differences (a) between students and (b) between assignments on how well coarse-grained time-on-task correlates with fine-grained time-on-task?

In this study, we compare two different metrics for time-on-task that we call *coarse-grained time-on-task* and *fine-grained time-on-task*. The metrics are calculated for each student for each exercise they attempted and submitted.

Coarse-grained time-on-task is calculated as the difference between the timestamp of the first submission and the first keystroke event for that assignment. We used the first submission instead of the last submission since some students resubmitted assignments that they had previously completed "just in case" right before the deadline. However, the choice of first versus the last submission does not affect the results considerably: in 95% of the cases, students only had a single submission for each assignment.

Fine-grained time-on-task was calculated by computing the differences between keystroke timestamps in the data until the first submission of the assignment while ignoring any differences that were greater than a *break threshold* that is used to approximate off-task behavior or "outliers". Different values for the break threshold are explored and reported.

The key difference between the two metrics is that the fine-grained time-on-task takes into account the breaks that students take while working on assignments, whereas the coarse-grained time-on-task does not. If the break threshold is arbitrarily large, no breaks are removed when computing the fine-grained time-on-task, and the two metrics are identical.

4. ANALYSES AND RESULTS

4.1 Differences Between Time-on-Task Metrics

To answer *RQ1*, we first analyzed different break threshold values to examine how different thresholds affect the number of distinct study sessions in the data. We define a distinct study session as any sequence of snapshots for an assignment between breaks in the data, where what is considered as a break depends on the break threshold. We then examined how the choice of break threshold affects the correlation between the coarse- and fine-grained time-on-task metrics across the whole data set. The strength of the correlation between the metrics can signal whether the metrics are measuring the same phenomenon, i.e. time-on-task.

Figure 1, in Appendix, shows how having a different break threshold for the fine-grained time-on-task affects the number of distinct study sessions for thresholds between 30 seconds and 1200 seconds (i.e. 20 minutes). We see that having a very low threshold (e.g. anything under 100 seconds) results in a very high number of study sessions compared to having a higher break threshold (e.g. anything over 600 seconds, i.e. 10 minutes). The figure only shows the number of sessions up to a break threshold of 20 minutes since at that point, the decrease in the number of sessions is very small. What this essentially illustrates is that if a student takes a short break of under 200 seconds or so, they are quite likely to return to the task, but if the break is longer (e.g. over 10 minutes), they are not likely to return to the task soon. Based on this, in our data, a break threshold of around 600 seconds would seem reasonable as at that point, the rate of decrease plateaus.

Figure 2 (Appendix) shows the Pearson's correlation coefficient between coarse- and the fine-grained time-on-task metrics for different break thresholds between 30 seconds and 1200 seconds (20 mins.). We first note that for all the thresholds visualized in Figure 2, the correlation is weak since it varies between 0.33 and 0.37. The figure shows that the correlation increases slightly as the break threshold gets bigger, but similar to the number of study sessions, the rate of increase seems to plateau at around the 600 second (10 min.) mark. The correlation does continue increasing beyond what is visualized in the figure and eventually, at around 13 days, it reaches 1, where the fine- and coarse-grained time-on-task metrics are equal. This means that some students had a break of around 13 days within a single assignment.

4.2 Student and Assignment-Specific Correlations Between Time-on-Task Metrics

To answer *RQ2a*, we first calculated both time-on-task metrics for each student for each assignment they submitted. We then calculated the correlation between the metrics for each student separately, which leaves us with a single correlation per student. We examine the distribution of these correlations to understand if there are differences between students on how much the fine- and coarse-grained time-on-task metrics correlate. To answer *RQ2b*, we calculated the correlation between the coarse- and fine-grained metrics for each assignment separately, leaving us with a single correlation per assignment. Similar to *RQ2a*, we study the distribution of these correlations to see if there are assignment-specific differences in how well the two metrics correlate.

For analyzing student and assignment-specific differences in how well the coarse- and fine-grained time-on-task metrics correlate, we used a break threshold of 600 seconds (i.e. 10 minutes) for the fine-grained time-on-task metric. We chose 600 seconds as the results for *RQ1* showed that in our data, 600 seconds seems like a reasonable value to consider a student being on a break (Section 4.1).

Figure 3, in Appendix, shows the distribution of the correlations between the coarse- and fine-grained time-on-task metrics for individual students. The mean correlation is 0.47 with a standard deviation of 0.24 and the 95% confidence interval is 0.43 to 0.51. We notice from the figure that there are differences between students in how well the coarse- and fine-grained time-on-task metrics match each other. On average, the correlation seems moderate, with most students having a correlation between 0.2 and 0.6.

Figure 4, in Appendix, shows the distribution of the correlations between the coarse- and fine-grained time-on-task met-

rics for individual assignments. The mean correlation is 0.33 with a standard deviation of 0.21 and the 95% confidence interval is 0.30 to 0.36. We notice from the figure that similar to students, there are also differences between assignments. Compared to the between-students analysis (*RQ2a*), the assignment distribution is slightly more centered around the mean. Similar to the between-student analysis, the correlations for the assignments are also, on average, moderate.

5. DISCUSSION

5.1 Coarse- vs Fine-Grained Time-on-Task

We observed that our coarse-grained time-on-task metric poorly approximated our fine-grained time-on-task metric. The coarse-grained metric imitates metrics from earlier work where time-on-task has been calculated based on, for example, students' first and last submissions for an assignment [6], while the fine-grained time-on-task metric is somewhat similar to earlier works that utilized LMS trace data [16], although considerably more fine-grained.

We propose that the fine-grained metric explored in this work is a better metric for measuring time-on-task than a metric that relies on coarse-grained data, but removes outliers to keep time-on-task values meaningful. Prior work has suggested, for example, that large values are just ignored [16]. However, if we rely on removing outliers, we are bound to include data that is not accurate that was simply not caught by the outlier detection. For example, if two students both have a time-on-task estimate of two hours with a coarse-grained time-on-task metric, it is possible that one of them worked for ten minutes, while the other worked for a full 120 minutes. In this case, the actual time-on-task is drastically different, but the coarse-grained time-on-task estimate would be the same for both.

One downside of the fine-grained time-on-task metric is that it requires a break threshold to calculate time-on-task. Deciding on a good break threshold is not straightforward, and is most likely context-dependent. This work is not the first to note this issue: for example, both Nguyen [23] and Kovanović et al. [16] examined different cut-offs for outlier detection, which is similar to our work in examining different break thresholds.

5.2 Student- and Assignment-Specific Correlations

We identified student- and assignment-specific differences in how well coarse- and fine-grained time-on-task metrics correlate. This makes sense since the main difference between the metrics is that the fine-grained metric takes the breaks students take into account; thus, if a student does not take many breaks while working on assignments, the difference between the two time-on-task metrics will not be significant compared to a student who takes long breaks within single assignments. Here, factors such as possible previous programming experience and study fatigue may come into play and should be analyzed in future work.

Similarly, we found that there are differences between assignments in how much the two metrics correlate. Since the course has many small assignments, but also some bigger, more complex assignments, it makes sense that, for example, students might take more breaks during the bigger assignments compared to the smaller ones, which would have an effect on the correlation between the two metrics.

5.3 Conclusion and Future Work

In this work, we studied how two different time-on-task metrics built from programming log data correlate with each other. One of the metrics utilizes fine-grained keystroke data and takes the breaks students take during assignments into account by not including the breaks in its time-on-task estimate. The other time-on-task metric is more coarse-grained and includes any breaks students take during assignments in its time-on-task estimate.

Our results show that the correlation between the two metrics is at best moderate, which suggests that the choice of time-on-task metric can significantly impact the results of studies based on time-on-task analysis. This brings into question whether previous results that have used different metrics for measuring time-on-task are comparable with one another. Additionally, our results show that, at least in our context, there are also student- and assignment-specific differences in how much the two metrics correlate.

We acknowledge that we do not have a ground truth for time on task, i.e., both our metrics are only proxies. As part of our future work, we are looking into augmenting keystroke data from the programming environment with log data from other learning environments and self-reported time-on-task estimates. Similarly, in this work, we examined different break thresholds over all the data when identifying a break threshold; in future work, we will be looking at to what extent optimal break thresholds vary between students. We also acknowledge that we did not analyze how time-on-task relates to course outcomes, which has often been included in time-on-task studies (e.g. [16, 19, 23]). In the future, we will also be looking into how the studied metrics and different break thresholds relate to course performance.

6. REFERENCES

[1] A. Ahadi, R. Lister, H. Haapala, and A. Vihavainen. Exploring machine learning methods to automatically identify students in need of assistance. In *Proceedings of the eleventh annual international conference on international computing education research*, pages 121–130, 2015.

[2] L. W. Anderson and C. C. Scott. The Relationship Among Teaching Methods, Student Characteristics, and Student Involvement in Learning. *Journal of Teacher Education*, 29(3):52–57, May 1978. Publisher: SAGE Publications Inc.

[3] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: when students" game the system". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390, 2004.

[4] C. Calderwood, P. L. Ackerman, and E. M. Conklin. What else do college students "do" while studying? an investigation of multitasking. *Computers & Education*, 75:19–29, 2014.

[5] F. E. V. Castro, S. Adjei, T. Colombo, and N. Heffernan. *Building Models to Predict*

Hint-or-Attempt Actions of Students. International Educational Data Mining Society, June 2015.

[6] S. H. Edwards, J. Snyder, M. A. Pérez-Quiñones, A. Allevato, D. Kim, and B. Tretola. Comparing effective and ineffective behaviors of student programmers. In *Proceedings of the fifth international workshop on Computing education research workshop*, pages 3–14, 2009.

[7] J. B. Fenwick Jr, C. Norris, F. E. Barry, J. Rountree, C. J. Spicer, and S. D. Cheek. Another look at the behaviors of novice programmers. *ACM SIGCSE Bulletin*, 41(1):296–300, 2009.

[8] S. Getting and K. Swainey. First graders with ipads?. *Learning & leading with technology*, 40(1):24–27, 2012.

[9] J. D. Gobert, M. S. Pedro, J. Raziuddin, and R. S. Baker. From Log Files to Assessment Metrics: Measuring Students' Science Inquiry Skills Using Educational Data Mining. *Journal of the Learning Sciences*, 22(4):521–563, Oct. 2013.

[10] T. L. Good and T. M. Beckerman. Time on Task: A Naturalistic Study in Sixth-Grade Classrooms. *The Elementary School Journal*, 78(3):193–201, Jan. 1978. Publisher: The University of Chicago Press.

[11] M. C. Jadud. A First Look at Novice Compilation Behaviour Using BlueJ. *Computer Science Education*, 15(1):25–40, Mar. 2005.

[12] M. C. Jadud. Methods and tools for exploring novice compilation behaviour. In *Proceedings of the second international workshop on Computing education research*, pages 73–84, 2006.

[13] P. M. Johnson, H. Kou, J. Agustin, C. Chan, C. Moore, J. Miglani, S. Zhen, and W. E. Doane. Beyond the personal software process: Metrics collection and analysis for the differently disciplined. In *25th International Conf. on Software Engineering, 2003. Proceedings.*, pages 641–646. IEEE, 2003.

[14] N. Karweit. Time-on-task reconsidered: Synthesis of research on time and learning. *Educational leadership*, 41(8):32–35, 1984.

[15] J. S. Kounin and P. V. Gump. Signal systems of lesson settings and the task-related behavior of preschool children. *Journal of Educational Psychology*, 66(4):554–562, 1974. Place: US Publisher: American Psychological Association.

[16] V. Kovanović, D. Gašević, S. Dawson, S. Joksimović, R. S. Baker, and M. Hatala. Does time-on-task estimation matter? implications for the validity of learning analytics findings. *Journal of Learning Analytics*, 2(3):81–116, 2015.

[17] V. Kovanović, D. Gašević, S. Dawson, S. Joksimović, R. S. Baker, and M. Hatala. Penetrating the black box of time-on-task estimation. In *Proceedings of the fifth international conference on learning analytics and knowledge*, pages 184–193, 2015.

[18] R. N. Landers and A. K. Landers. An Empirical Test of the Theory of Gamified Learning: The Effect of Leaderboards on Time-on-Task and Academic Performance. *Simulation & Gaming*, 45(6):769–785, Dec. 2014. Publisher: SAGE Publications Inc.

[19] J. Leinonen, L. Leppänen, P. Ihantola, and A. Hellas. Comparison of time metrics in programming. In *Proceedings of the 2017 ACM conf. on International Computing Education Research*, pages 200–208, 2017.

[20] L. Leppänen, J. Leinonen, and A. Hellas. Pauses and spacing in learning to program. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, pages 41–50, 2016.

[21] J. McKeogh and C. Exton. Eclipse plug-in to monitor the programmer behaviour. In *Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange*, pages 93–97, 2004.

[22] C. Murphy, G. Kaiser, K. Loveland, and S. Hasan. Retina: helping students and instructors based on observed programming activities. In *Proceedings of the 40th ACM technical symposium on Computer Science Education*, pages 178–182, 2009.

[23] Q. Nguyen. Rethinking time-on-task estimation with outlier detection accounting for individual, time, and task differences. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 376–381, 2020.

[24] C. Norris, F. Barry, J. B. Fenwick Jr, K. Reid, and J. Rountree. Clockit: collecting quantitative data on how beginning software developers really work. In *Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pages 37–41, 2008.

[25] S. Park. Analysis of Time-on-Task, Behavior Experiences, and Performance in Two Online Courses with Different Authentic Learning Tasks. *International Review of Research in Open and Distributed Learning*, 18(2):213–233, 2017. Publisher: Athabasca University Press (AU Press).

[26] M. M. T. Rodrigo, T. C. S. Andallaza, F. E. V. G. Castro, M. L. V. Armenta, T. T. Dy, and M. C. Jadud. An Analysis of Java Programming Behaviors, Affect, Perceptions, and Syntax Errors among Low-Achieving, Average, and High-Achieving Novice Programmers. *Journal of Educational Computing Research*, 49(3):293–325, Oct. 2013.

[27] M. Romero and E. Barbera. Quality of learners' time and learning performance beyond quantitative time-on-task. *International Review of Research in Open and Distributed Learning*, 12(5):125–137, 2011.

[28] J. Spacco, P. Denny, B. Richards, D. Babcock, D. Hovemeyer, J. Moscola, and R. Duvall. Analyzing student work patterns using programming exercise data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 18–23, 2015.

[29] J. Spacco, D. Hovemeyer, W. Pugh, F. Emad, J. K. Hollingsworth, and N. Padua-Perez. Experiences with marmoset: designing and using an advanced submission and testing system for programming courses. *ACM Sigcse Bulletin*, 38(3):13–17, 2006.

[30] J. Stallings. Allocated academic learning time revisited, or beyond time on task. *Educational researcher*, 9(11):11–16, 1980.

[31] E. S. Tabanao, M. M. T. Rodrigo, and M. C. Jadud. Identifying at-risk novice java programmers through the analysis of online protocols. In *Philippine Computing Science Congress*, pages 1–8, 2008.

[32] H. J. Walberg. Synthesis of research on time and learning. *Educational leadership*, 45(6):76–85, 1988.

APPENDIX
A. FIGURES

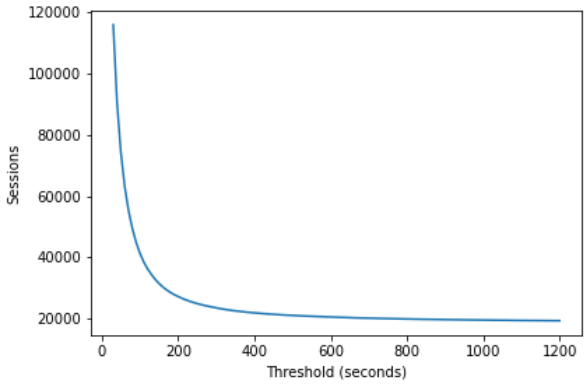

Figure 1: Number of distinct study sessions with different thresholds for breaks for the fine-grained time-on-task metric. The x-axis is the threshold for considering the student to be on a break in seconds. The y-axis is the number of study sessions in the data. Data is shown for thresholds between 30 seconds and 1200 seconds (20 minutes).

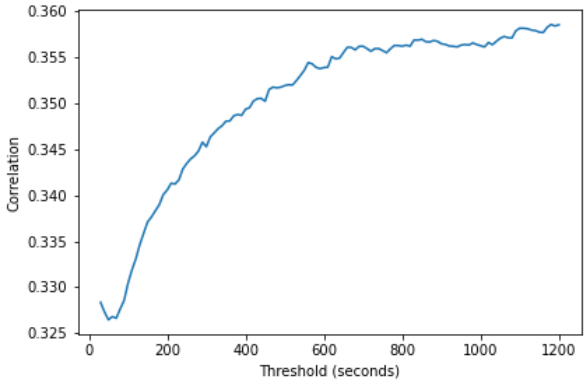

Figure 2: The correlation between the coarse and the fine-grained time-on-task metric with different thresholds for breaks for the fine-grained time-on-task metric. The x-axis is the threshold for considering the student to be on a break in seconds. The y-axis is the Pearson correlation coefficient between the coarse and the fine-grained time-on-task metric. Data is shown for thresholds between 30 seconds and 1200 seconds (20 minutes).

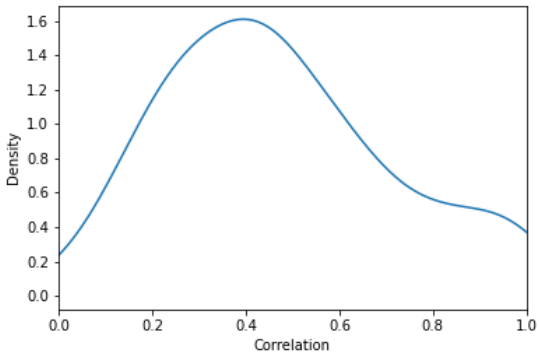

Figure 3: The distribution of student-specific correlations between the coarse- and fine-grained time-on-task metrics, where fine-grained time-on-task was calculated with a 600 second break threshold. The x-axis is the Pearson correlation coefficient and the y-axis is the density.

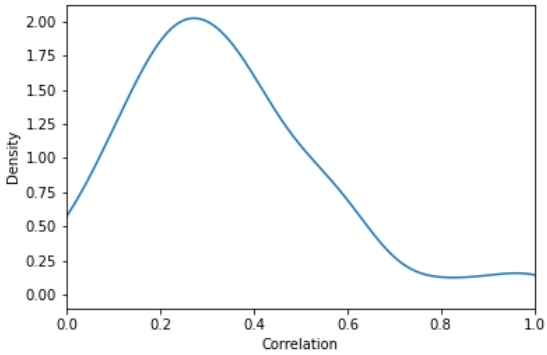

Figure 4: The distribution of assignment-specific correlations between the coarse and the fine-grained time-on-task metrics, where fine-grained time-on-task was calculated with a 600 second break threshold. The x-axis is the Pearson correlation coefficient and the y-axis is the density.