

# What will you do next?

## A sequence analysis on the student transitions between online platforms in blended courses

Niki Gitinabard  
North Carolina State  
University  
Raleigh, NC, USA  
ngitina@ncsu.edu

Tiffany Barnes  
North Carolina State  
University  
Raleigh, NC, USA  
tmbarnes@ncsu.edu

Sarah Heckman  
North Carolina State  
University  
Raleigh, NC, USA  
sarah\_heckman@ncsu.edu

Collin F. Lynch  
North Carolina State  
University  
Raleigh, NC, USA  
cflynch@ncsu.edu

### ABSTRACT

Students' interactions with online tools can provide us with insights into their study and work habits. Prior research has shown that these habits, even as simple as the number of actions or the time spent on online platforms can distinguish between the higher performing students and low-performers. These habits are also often used to predict students' performance in classes. One key feature of these actions that is often overlooked is how and when the students transition between different online platforms. In this work, we study sequences of student transitions between online tools in blended courses and identify which habits make the most difference between the higher and lower performing groups. While our results showed that most of the time students focus on a single tool, we were able to find patterns in their transitions to differentiate high and low performing groups. These findings can help instructors to provide procedural guidance to the students, as well as to identify harmful habits and make timely interventions.

### 1. INTRODUCTION

Modern blended classrooms are defined by suites of educational tools such as learning management systems, online forums, intelligent textbooks, video lectures, groupware tools, and even ticketing systems for office hours. The ubiquity of such tools provides researchers with a rich amount of data on students' study behaviors, work habits, and their learning trajectories. This data can help researchers to identify good and bad study habits among students as well as to define measures for estimating students' performance early on in the courses. Large datasets of this type first became available in Massive Open Online Courses (MOOCs) that have supported informative research on students'

online study habits. While these tools have now become the norm in many classroom settings and while there has been substantial research on how students use the individual tools, we have far less understanding of how students work across tools and how different patterns of use may affect their learning. Our goal in this research is to address this question through the use of sequence mining. By developing a better understanding of student activities in different online systems and their transitions between these tools, we can provide the instructors with insight on how their students usually behave when they are not in class.

Prior research has shown that there are several features easily extracted from user logs that can distinguish high performing students from the lower performing ones. Researchers have found several informative features such as number of videos watched per week, completing assignments [27], starting early [28, 34], or skipping videos and assignments [12] that were associated with students' performance and dropout in MOOCs. Studies in blended courses showed that features such as course attendance, web page views, number of watched videos, number of pauses in videos, and the number of attempts before getting each question right are correlating with student dropouts [6].

More recent work in MOOCs, Intelligent Tutoring Systems (ITSs), and blended courses has focused on grouping the student activities into study sessions and analyzing these sessions and the sequence of students' actions in them. Some researchers have analyzed features based upon these sessions in MOOCs and blended courses, such as the duration [2, 29]. However, those studies overlook the patterns of student transitions between different states or different tools. Other researchers have studied the sequences of student actions in each session, but most of those studies are focused on MOOCs or ITSs and not many of them have focused on blended courses and the data collected from the several tools that the students use for these classes. Some of these studies have relied on Hidden Markov Models on the sequences of student actions and compared the diagrams between high and low performing students (e.g. [8, 10, 14]), while others have clustered these sequences to find groups of similarly behaving students in classes (e.g. [4, 11, 7, 17, 18, 19, 25, 30]). These studies have often been able to identify relevant clusters

Niki Gitinabard, Sarah Heckman, Tiffany Barnes and Collin Lynch "What will you do next? A Sequence Analysis of the Student Transitions between Online Platforms" In: *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, Collin F. Lynch, Agathe Merceron, Michel Desmarais, & Roger Nkambou (eds.) 2019, pp. 59 - 68

among the students such as “confirmers” and “non-confirmers” [11] or “behind”, “on-track”, “auditing”, and “out” [17]. Also, other sets of studies have performed differential pattern mining on such sequences to find the patterns that are different between high and low performing students [16, 15, 13, 24]. And finally, another part of this research treats the sequences of actions as strings and uses analysis of N-grams to identify the popular trends in student activities and transitions [22, 5, 31, 32]. These methods are helpful in revealing many of the students’ behavioral patterns and the differences between the different performance groups, but are mostly focused on MOOCs or ITSSs.

Despite the extensive research in this area on MOOCs and ITSSs, studies on student transactions in blended courses are limited and most of them focus on correct/incorrect attempts on the same platform (e.g. the assignment submission systems) [11]. In this work, we collected activity logs from four online platforms for two offerings of two on-campus classes at North Carolina State University. In these classes, Piazza was used as a discussion forum, Moodle as a Learning Management System (LMS) was the means of sharing the course material and assignments, Github was used in one class as a version control as well as a code submission tool for the projects, and WebAssign was used for assignment submissions and automated grading in the other class. We aligned the logs into a single coherent transaction record, grouped the individual student actions into study sessions, and extracted the sequences of student actions from them. Finally, we labeled the students as the “Distinction” group who gained an A- or above and the “Non-distinction” group who gained a B+ or below in these courses and used N-gram analysis as well as Apriori studies to find the answers to the following research questions:

- RQ1** What are the most common transitions between different course tools?
- RQ2** Which transitions are significantly different between the distinction and non-distinction groups?

The answers to these questions can help us understand the trends of student activities better, to find key differences between high-performing students and the lower performing ones, and help the instructors to provide guidance to the students as they work or identify harmful patterns early in the semesters.

## 2. LITERATURE REVIEW

### 2.1 Students’ Online Activity Analysis

Since detailed online student logs have been available for the MOOCs, there have been extensive studies of student behaviors using these logs to identify their association with the students’ performance and attrition. Even simple measures such as number of videos watched are shown to be predictive of students’ attrition and performance in MOOCs. Some examples of these features include the number of videos watched per week, whether the student watched all of the lectures, or completed all of the assignments [27]. They also included joining the course early [28, 34], skipping videos or assignments, assignment performance [12], spending more time on each assignment [3], the number of lecture views/downloads, quiz attempts, and forum views/-posts/comments [9]. Some researchers such as Yang et al. have gone further and constructed more complex features to represent student confusion and shown that increased confusion is associated with dropout in MOOCs [33]. Chen et al. has studied

blended courses and has also shown that features such as course attendance, web page views, videos watched, video pauses, and assignment attempts are also correlated with student dropout [6]. All of these features, while informative, overlook an important part of the information that online logs provide us: the sequences of actions and transitions among different platforms.

To analyze a group of student actions as a whole, researchers have suggested defining study sessions. Prior work has suggested different methods for defining study sessions such as having a “fixed duration” [5], using “browser navigations”, or having a ‘cutoff’ [2]. But as Kovanovic et al. showed, the choice of the method or the cutoff time is not trivial and there is no best method for everyone [20]. They suggested exploring the data to find the cutoff or method that matches the dataset best. Amnueypornsakul et al. defined study sessions and used the actions and the sessions to calculate measures such as the length of the action sequence, the number of occurrences of each activity, and the number of Wiki page views [2]. Sheshadri et al. also defined study sessions based on the time difference between student actions and extracted measures such as the average number of actions in each session, inconsistency of the student (i.e. how different the number of the sessions started by a student is from the class average and how infrequent they get online), average length of sessions, and sessions including discussion forum activity [29]. While these features can add to the information collected directly from the online tools, they still do not consider transitions from one type of action to the other.

### 2.2 Sequence Analysis

#### 2.2.1 Markov Models

Several methods have been used for analyzing the sequences of student actions. The first and most popular is the use of Markov chains and Hidden Markov Models. Jeong et al. for example, trained models based upon system logs of a learning-by-teaching system called Betty’s Brain in which the students learn material by teaching an artificial agent, Betty [14]. The possible student actions in this platform are reading the material they are trying to teach Betty; editing the material; using links and concepts in forms of adding, removing, or changing (e.g. link add); querying the agent by asking questions about the provided material; asking Betty the agent to explain the answer she just gave; and giving a quiz to assess how well Betty has learned. The authors extracted sequences of student actions on the platform and used a Hidden Markov Model to analyze their behavior. They found that students who generated better concept maps used balanced learning strategies that include moving between different actions, while the students who generated low scoring concept maps typically focused too much on getting the quiz answers correct. Faucon et al. used semi-Markov chains to model student activities in 61 MOOCs offered by EPFL university on Coursera and EdX platforms [8]. They utilized an Expectation Maximization algorithm for fitting the model and showed a graphical representation of their results on the transitions between different states (e.g. submission, forum participation, video watching, etc) for students of different behavior profiles. Similarly, Geigle et al. used clickstream data from a UIUC Text Retrieval MOOC on Coursera to generate a transition diagram between the different tools [10]. While Markov models are suitable for modeling student transactions between different states and are easy to visualize, the differences they show are often hard to quantify and compare between groups [14].

### 2.2.2 Sequence Clustering

Another approach for analyzing sequences of student actions is by clustering them. Desmarais et al. for example, collected the action logs of students in a college math learning environment [7]. In that work, they defined distinct sessions where the students paused for more than 5 minutes between them unless the action after the pause was a submission to an exercise, which might take longer. They then clustered the sequences using the Levenshtein distance and identified three types of sessions. The first was when the students showed exploratory behavior and engaged in a mixture of browsing through exercises and notes. The second type were the short sessions comprising a variety of behaviors such as browsing and attempting the exercises and quizzes. The third were exercise intensive sessions mostly consisting of exercise logs. Kizilcec et al. used a similar approach on the student engagements in a MOOC [17]. For each assessment period, they labeled the students as either “behind”, “on track”, “auditing”, or “out” based on their engagement with the course material. Then, they applied K-means clustering on the sequences of the student states in all assessment periods to identify the prototypical engagement patterns and were able to observe four clusters of students as completing, auditing, disengaging, and sampling.

A similar analysis was performed by Guerra et al. on data collected from QuizJET, which was a voluntary practice platform for students in an introduction to programming blended course [11]. They extracted the sequence of correct and incorrect submissions for each student and each question. Then, by comparing the sequences of different students to the sequences of the same student, they observed that these sequences are personal and can show people’s study approaches like a “study genome”. While these genomes were shown to evolve throughout the semester, the evolved genomes for a single user were still more similar than the genomes across different users. They were able to cluster the students based on their genomes and identify two groups as the confirmers and the non-confirmers. The confirmers kept trying examples of the same topic even after they got one correct, while the non-confirmers moved on to the next topic after they were able to solve one example correctly. Finally, Boroujeni et al. clustered student activities in a MOOC and were able to identify four user profile types: users who watch videos before making submissions (44% of the users), users who make submissions without watching videos (2% of the users), users who watch videos and never submit (7% of the users), and the users who change their habit in the semester (47% of the users) [4]. These categories are similar to the ones suggested by Kizilcec et al. [17]. While clustering seems to offer much insight on similar sequences and differences between different groups of students, it is often challenging to interpret these clusters and get to real world groups of students.

To account for the randomness in the generation of Markov Models, some researchers have generated Markov Models based upon each individual sequence and then clustered them to obtain more meaningful results. Köck et al. for example, designed an analysis pipeline which included a pre-processor which extracted activity sequences from the raw data, a modelling unit which converted the sequences into Deep Markov Models, and a final clustering unit [19]. They applied this pipeline to extract common transitions exhibited by different performance groups in a Physics course at the US Naval Academy. Similarly, Shih et al. applied the same clustering method on the Hidden Markov Models based on student activities in a Geometry Cognitive Tutor

[30]. Klingler et al. developed an evolutionary clustering pipeline to improve cluster stability over multiple training sessions in the presence of noise [18]. This pipeline extracts action sequences from log data, transforms them into per-session Markov Chains, computes pairwise similarities between students for every session, then performs clustering using evolutionary clustering, and uses the Akaike information criterion with correction (AICc) to select the best model. They suggested that this pipeline can be used as a black box on any ITS. While the combination of clustering and Markov Models might overcome some disadvantages of each individual, the results are still challenging to interpret as noted by Shih et al. [30].

### 2.2.3 Sequences as N-grams

Another approach often taken when analyzing students’ sequence data is treating the sequence of actions as a sequence of strings, and then identifying the common N-grams in it. Li et al. and Sinha et al. for example, extracted the sequences of actions for users in MOOCs and used the frequency of N-grams in such sequences as predictive features to predict students’ performance and certification [22, 31]. Maldonado et al. also performed a similar analysis on data extracted from an interactive tabletop (Digital Mysteries) and were able to identify frequent sequences of actions that distinguish between different performance groups [23]. Wen and Rosè applied this method to extract the most common types of sessions among students and were able to identify 4 types of sessions as lecture and peer assessment sessions, browse course sessions, assignment and forum sessions, final quiz and survey sessions, and lecture and quiz sessions [32]. Brooks et al. defined fixed duration sessions during the semester (i.e. 1 day, 3 days, 1 week, and 1 month) and recorded students’ activity in each frame as a binary feature [5]. They used frequent N-grams extracted from these sequences as features to make early and cross-class predictions of student dropout. While N-grams are easier to process since there are many available libraries for analyzing them, extracting information from them can still be challenging and require expert help at times.

### 2.2.4 Differential Pattern Mining

A newer approach which is mostly applied to ITS data is Differential Pattern Mining. The algorithms in this approach are able to identify patterns that are more frequent than a specific threshold and are significantly different between the two specified groups such as pass/fail students [1]. Kinnebrew et al. for example used a differential sequence mining algorithm to extract the sequences that are different between the high performers and low performers using the Betty’s Brain platform [15, 16]. They found that the high performers more frequently engaged in reading activities in a monitoring context, while the lower performers usually perform short reads mostly not relevant to their recent actions [16]. Herold et al. applied the same analysis to the sequences collected with LivescribeTMDigital pens, used to complete all of their homework and exams [13]. These pens are able to log students’ handwriting as time-stamped pen strokes providing the sequence in which it was written. Using this method, they were able to identify 98 patterns in total and use them to make predictions on the students’ performance in the closest exam after the task with an  $R^2$  of 0.3. While this approach is able to make the differences in performance groups bolder, it is still relatively new, the libraries for it are limited, and it is also possible to end up with a large number of rules that will need clustering again.

### 3. DATASET

We collected data from two offerings each of two distinct courses, a Discrete Math course (DM) in the Fall semesters of 2013 and 2015, and a Java programming course (Java) in the Fall of 2015 and 2016. The 2015 offerings of these courses occurred contemporaneously. Both of these courses are core undergraduate courses, required for students majoring and minoring in Computer Science. They both use significant online materials and support and can be considered blended courses. The online materials include online assignments, supplemental material, and student forums.

In all these classes, Moodle is used as an LMS for providing the course material and the assignment descriptions to the students. Piazza is used as the discussion forum and the main resource for the students in these courses to ask questions and get answers from the teaching staff as well as to have discussions with their peers. The students were able to post completely anonymously for a brief time in DM-2013 but it was blocked in all other courses. Posting anonymously to other students was always allowed. Posting on Piazza was not required in any of these classes, but it was encouraged by the teaching staff as the best choice of asking for help. In the DM classes, the instructors used multiple answer questions on WebAssign for a large portion of the assignments. WebAssign was configured to allow the students to attempt each question several times to get it correctly and provides the students with instant automatic feedback on their answers. In the Java classes, the students use Github as a version control for keeping track of their code and editing in teams, as well as the means for submitting their code for grading. The students' Github repositories were connected to Jenkins servers, which ran several test cases on their code after each pushed commit. Some of the tests were predefined and authored by the instructional staff and some others were the tests designed by the students to test their own code. This enabled students to get instant feedback on their code and possibly revise it after each submission. Our datasets in this study consist of the Piazza discussions, Moodle logs, and final grades for all the classes as well as Github commit logs for Java classes and WebAssign logs for DM-2013.

While some of the tools used in these classes are different, they play similar roles in the classes. In the DM classes students use WebAssign to submit their assignments and to receive immediate automated feedback. And in this class they can re-submit as many times as they wish to get the right answer. Similarly, in the Java classes the students use Github for making submissions on their projects. While these submissions often take more time than answering a simple question on WebAssign, the students are still able to get immediate feedback from Jenkins and to try again. Consequently, while some visible trends in these classes might be different, we expect the trends for WebAssign and Github to be similar, because they play a similar role. Similarly, in both these classes, Moodle and Piazza can be considered as support platforms since the students can use the course material, project descriptions, and the questions on the forum to resolve their confusions. The types of support these platforms are offering are quite different, since asking questions on Piazza is a more direct means of asking for help than referring to the class material.

More information on the population of these classes is shown in Table 1. The grade distributions for these classes are shown in Figure 1. Both these courses are C-wall courses, where the students need a C or better in them to proceed to the next computer science courses in the curriculum. As shown in these

Table 1: Statistics of Each Class

Class	DM-2013	DM-2015	Java-2015	Java-2016
Total Students	251	255	181	206
Teaching Assistants	5	5	9	9
Instructors	2	2	4	4
Average Grade	81.2	87.6	79.7	79.9

figures, most of the students performed well in these classes. Thus, we decided that clustering them into pass/fail groups would be uninformative and result in a skewed dataset. Since the median grades for all these datasets were close to 90, the cutoff between an A- and a B+ in the courses, we decided to partition the classes into two groups, the *distinction* group earning an A- or above, and the *non-distinction* earning a B+ or below. This partitioning resulted in an almost even groups of the students. We believe that this segmentation leaves room for adjusting the analysis for other classes with different grade distributions.

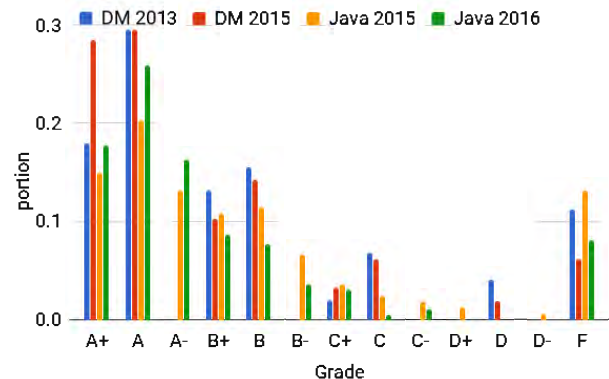


Figure 1: The Distribution of Grades in Different Classes

#### 3.1 Discrete Math

This course covered material such as propositional logic, predicate calculus, methods of proof, elementary set theory, the analysis of algorithms, and the asymptotic growth of functions. The total enrollments in these classes consisted of 251 students in DM-2013 and 255 students in DM-2015. Both of these classes were offered in two sections by two instructors with 5 shared teaching assistants. The average final grade in DM-2013 was 81.2 and 87.6 in the 2015 class. Both sections in each year shared the same Moodle page for assignments and class material, a Piazza forum for discussions, and both used WebAssign as well as hand-graded assignments. The only major difference between these two offerings was that in 2015 the instructor consciously delayed responding to posts on Piazza so that the TAs and other students would be more involved. However, most of the posts were still answered in a similar time frame to the ones in 2013 by the lead TA in that class.

#### 3.2 Java Programming Concepts

The material of the Java class mainly consisted of software design and testing, encapsulation, polymorphism, inheritance, linear data structures, finite-state machines, and recursion. The total enrollment in these classes was 181 students with an average grade of 79.7 in 2015 and 206 students with an average grade of 79.9 in 2016.

Both of these classes were offered in two different in-person sections by two separate instructors as well as a distance education section by two other instructors, having a total of four

instructors with nine shared teaching assistants. We removed the data for the distance education students from our analysis since they were a much smaller group and differed substantially from the local students who could engage in face-to-face interactions. These classes used Piazza for discussions, Moodle for sharing course materials, Github for working on group projects, and Jenkins for automated code evaluation.

While the teaching material and the methods were mostly similar across both the offerings, there was a major difference in the lab structures for these classes. Both course offerings included lab sessions. In each session, the students completed a short assignment in a team of three with assistance from the teaching staff. One key difference between the course offerings was in the structure of the lab sessions. In 2015, the labs were conducted in 8 class sessions, thus engaging all of the students and the TAs simultaneously. In 2016 however, students were enrolled in separate lab sessions (approximately 24 students each) with a dedicated TA and participated in 12 lab sessions. Additionally, in 2015, students continued to work with the same peers for all lab assignments while in 2016, they rotated partners after every four tasks, thus giving them a chance to meet and work with a wider variety of people.

## 4. METHODS

### 4.1 Action Sequence Generation

We began by collecting the logs from Piazza, Moodle, Github, and WebAssign for the courses. Later, we merged them into a single class-level transaction file sorted by time. We then generated study sessions on student activities based on their online transactions as discussed in our prior work [29].

As Kovanovic et al. suggested, we decided to explore our data to find the best method for generating study sessions [20]. Since there was no specific time length for the student sessions in our data, we decided to use a set cutoff time,  $m$ , for defining the sessions. If two consecutive actions are less than  $m$  minutes apart, they belong to the same study session. Otherwise, that session ends and the second activity after  $m$  minutes is a start of a new session. We plotted the average time differences between sessions, the total number of sessions, and the average number of activities per session for different cutoff times. These plots showed us two points with major changes that were chosen as the cutoff times for “study sessions” and “browser sessions”. We chose 15 minutes as the cutoff time for browser sessions, which show the times that the students have been online for the entire session. We also chose 40 minutes as the cutoff time for study sessions, which allows the time for the students to go offline for coding or solving problems on paper and get back online. We used this gap between online actions of the students considering that they often work offline before committing their code to Github or solve a problem on paper before submitting an answer on WebAssign. In this work, we focused on study sessions since they showed more transitions between different platforms. The total number of sessions for each group in each class is shown in Table 2. In the end, we recorded the sequence of student actions in these sessions for further analysis.

Table 2: The Total Number of Sessions for Distinction and Non-distinction Groups in Different Classes

Class Name	Count in Distinction	Count in Non-distinction
DM-2013	7,697	6,533
DM-2015	6,574	3,434
Java-2015	12,219	12,786
Java-2016	19,913	9,829

Similar to Kinnebrew et al. and Maldonado et al., we decided to compact the action sequences [16, 15, 23]. For that purpose, we replaced consecutive occurrences of the same actions by the “+” notion (e.g. MMM was replaced with M+). Our prior work showed that 90% of the student sessions consisted of access logs to the same platforms [29]. Also, the nature of most of these platforms requires consecutive submissions, such as multiple commits to Github for solving issues or multiple submissions on WebAssign until they find the right answer and there is not much of a difference between asking a question on Piazza after 5 submissions or 6. Abstracting these repetitions helps us spot the transitions between these platforms more easily and spot more similar sequences among students.

### 4.2 Sequence Mining

In order to explain our methods, we first need to define the common terminology in sequence mining. Based on Agrawal et al., the “support” of a sequence is defined as the ratio of occurrences of that sequence among all the sequences in the data [1]. For example, if a sequence  $S$  has happened 10 times among a student’s study sessions and the student has a total of 100 occurred sequences, the support for  $S$  will be 0.1 for that student. Looking at the support metric helps us to look into what percentage of this student’s sequences are  $S$ , rather than how many occurrences of  $S$  this student has. It also simplifies the comparisons between high and low performing students, since generally, the number of all actions for high performing users are higher and this might stop us from spotting the major differences between the students from different performance groups.

Another term often used in sequence mining is ‘confidence’. Based on Agrawal et al., the confidence of the action  $B$  following the action  $A$  ( $A \rightarrow B$ ) shows how likely it is for action  $B$  to occur after  $A$  and is defined as:

$$Confidence(A \rightarrow B) = \frac{Support(A \cap B)}{Support(A)}$$

To identify the most common patterns among the students, we applied the idea of N-gram analysis as in prior work [22, 5, 31, 32, 23]. In text mining, an N-gram of length  $N$  (e.g. bigram) refers to a specific sequence of  $N$  words. Many times, the frequency or the count of N-grams are calculated and used as features. In this work, we treated the sequences of student actions as lists of words and using Scikit-learn library in Python [26], for each student, we calculated the support for all sequences of lengths of 2 - 3 to represent the transitions between every two tools and also keep room to count for repetitions. Then, we collected these numbers for the distinction and non-distinction groups into two separate lists for each sequence. We extracted the average support percentage for each sequence in each group to find the most common patterns among them. Additionally, we performed Kruskal-Wallis (KW) ANOVA test between the two lists for all sequences to find the patterns that occur with a different distribution among these

two groups [21]. The Kruskal-Wallis test is a good choice in this context because it does not assume normally-distributed data.

Also, to determine how likely the students are to transfer to a system after using another, we used the Apriori algorithm provided in Apyori library in Python. This algorithm is used to mine frequent item-sets and association rules [1]. It takes a minimum required support and performs in an incremental order, starting with single items (i.e. 1-sequences) that meet the support requirement ( $L_1$ ) and add other items to the set as long as the support meets the criteria ( $L_k$ ). In this work, we set the minimum support to a low number (0.02) to be able to find and compare even the rare transitions and the 1-sequences were defined as single actions on each platform. Based on Agarwal et al. the pseudo-code for this algorithm is as below:

```

 $L_1$  = frequent 1-sequences

for ( $k=2; L_{k-1} \neq \emptyset; k++$ ) do
     $C_k$  = New candidates generated from  $L_{k-1}$ 

    for All possible sequences c do
        Increment the count of all candidates in  $C_k$  that are
        contained in c
    end for
     $L_k$  = Candidates in  $C_k$  with minimum support
end for
Answer = Maximal Sequences in  $\bigcup_k L_k$ 

```

To find the transitions often associated together, we applied the Apriori algorithm on the sequences from distinction students and non-distinction students and calculated confidence for the frequent ones.

While participation on Piazza was not mandatory, it was strongly encouraged by the instructors as the primary venue for help seeking in all of the courses. As a result, we would expect to observe a large number of transitions between the submission tools (i.e. WebAssign and Github) and Piazza. We also expect these transitions to be more frequent after students make consecutive submission attempts since students who struggle with assignments often make several tries before contacting the instructors. We also expect higher-performing students to make more of such transitions because seeking help when they are struggling, rather than postponing it for later or going without, will help them to perform better in the course.

## 5. RESULTS

Since the tools used in these systems are different, we will present our results in each part for each class separately.

### 5.1 RQ1. What are the most common transitions between different course tools?

#### 5.1.1 DM-2013

Our prior study on this class had shown that 90% or more of the student sessions are focused on a single tool and the sessions consisting of all WebAssign actions was the most common across them [29]. As Table 3, shows, consistent with our prior work, the most common sequence for both performance groups is repeated WebAssign Submissions, covering on average 70% of action sequences. This is not surprising due to the fact that the

students had unlimited submissions on this platform and often sought to “brute force” the answers.

The next most frequent pattern in both groups is multiple Moodle actions, which is again a unsurprising as students are required to log in on each session and must often navigate to their desired resources through a series of actions. Interestingly, transitions between WebAssign and Moodle are also comparatively frequent (the most frequent kind of transition between tools), consisting of approximately 4% of the total sequences. The more common transitions would be some submissions on WebAssign and moving to Moodle, while this sequence sometimes gets repeated several times as students move between these two tools and we can observe sequences like “w+m+w” on average in 0.4% of the students’ transitions or even more complicated ones such as “m+w+mw+”. Such transitions show students moving between class material like slides and the assignments and may show them referring to slides to revise their answers on WebAssign. We need to note that the sequences longer than 3 actions were not counted towards the calculation of support and confidence and thus, are not shown in the tables.

One would expect struggling students to move between WebAssign and Piazza to ask questions about the submissions, but as our results show, this transition does not happen frequently. Even among better performing students, it is more common to go to Moodle than Piazza after a couple of submissions, but it is even less likely for the lower-performing students. It seems like the students prefer to find the answers to their confusion among class material rather than asking questions or they prefer to leave help-seeking for another session.

Table 3: The Support for the Most Frequent Sequences in DM-2013 (W = WebAssign, M = Moodle, P = Piazza)

	Avg in Distinction	Avg in Non-distinction
W+	0.7064	0.7227
M+	0.1408	0.1615
W+M, M+W, MW, WM	0.0429	0.0380
P+	0.0303	0.0133
P+W, W+P, PW, WP	0.0039	0.0006
P+M, M+P, PM, MP	0.0004	0.0001

To better understand the student transitions between WebAssign and Moodle or WebAssign and Piazza, we calculated the confidence score for sequences in which Moodle and Piazza actions occur in the same session after one or more WebAssign actions. The results of the Apriori algorithm for this class are shown in Table 4. As we can see, there is almost a 10% chance of the students going to Moodle after one or more WebAssign submissions, while there is less than a 1% chance of them going to Piazza.

Table 4: Confidence for Different Transitions from WebAssign in DM-2013 (W = WebAssign, M = Moodle, P = Piazza)

	Distinction	Non-Distinction
$W \rightarrow M$	0.11	0.09
$W \rightarrow P$	0.007	0.003
$W+ \rightarrow M$	0.11	0.09
$W+ \rightarrow P$	0.007	0.003

#### 5.1.2 DM-2015

The most frequent sequences for this class are shown in Table 5. Unfortunately, in this class, we do not have access to the WebAssign data. Thus, there were far fewer patterns found in this data than the 2013 class. But as with the prior offering,

the Piazza actions seem to be not nearly as common as Moodle actions. Additionally, the transitions between Moodle and Piazza were rare.

Table 5: The Support for the Most Frequent Sequences in DM-2015 (M = Moodle, P = Piazza)

	Avg in Distinction	Avg in Non-distinction
M+	0.913	0.966
P+	0.081	0.034
PM, MP, M+P, P+M	0.003	0.000

### 5.1.3 Java-2015, Java-2016

The most frequent sequences for these classes are shown in Table 6. These classes are similar to DM-2015 in that consequent Moodle actions is the most common sequence with an average about 55-65% of students' sequences in 2015 and about 70-80% of the student sequences in 2016. While in these classes Github commits are similar to WebAssign activities in DM-2013, the findings show that the students tend to commit their changes far less frequently than they submit questions on WebAssign. Multiple commits on Github are the next most frequent and they occur in about 20-30% of the student sequences in 2015 and 10-15% of sequences in 2016. Similar to DM-2013, where the students often moved between the submission system and the course material on Moodle, in these classes 4-6% of the student sequences are moving between Github and Moodle, where only 0.1-0.5% of the sequences refer to moving between Github and Piazza. In these classes also, moving back and forth a few times between the platforms is observed and we can see sequences such as "g+m+g+m" or "g+mg+m+".

Table 6: The Support for the Most Frequent Sequences in Java Classes (G = Github, M = Moodle, P = Piazza)

	Avg in Distinction	Avg in Non-distinction
<b>Java 2015</b>		
M+	0.566	0.655
G+	0.294	0.204
G+M, M+G, GM, MG	0.041	0.043
P+	0.011	0.010
P+M, M+P, MP, PM	0.004	0.003
P+G, G+P, PG, GP	0.003	0.003
<b>Java 2016</b>		
M+	0.698	0.782
G+	0.134	0.089
G+M, M+G, GM, MG	0.062	0.052
P+	0.012	0.005
P+G, G+P, PG, GP	0.005	0.001
P+M, M+P, MP, PM	0.003	0.002

As with the DM-2013 class, we calculated the confidence score of action sequences that include Moodle and Piazza in the same session after one or more of Github actions. The results of the Apriori algorithm for these two classes are shown in Table 7. As we can see, there is a 28-37% chance of the students going to Moodle Github submissions, while there is only less than a 3% chance of them going to Piazza.

As our results show, the students seem more likely to go to the project descriptions or the course material after some submissions on Github rather than the discussion forum.

Table 7: Confidence for Different Transitions from Github in Java classes (G = Github, M = Moodle, P = Piazza)

	Distinction	Non-Distinction
<b>Java 2015</b>		
$G \rightarrow M$	0.31	0.36
$G \rightarrow P$	0.03	0.03
$G+ \rightarrow M$	0.31	0.37
$G+ \rightarrow P$	0.03	0.03
<b>Java 2016</b>		
$G \rightarrow M$	0.28	0.31
$G \rightarrow P$	0.02	0.007
$G+ \rightarrow M$	0.32	0.34
$G+ \rightarrow P$	0.02	0.01

## 5.2 RQ2. Which transitions are significantly different between the distinction and non-distinction groups?

### 5.2.1 DM-2013

The KW p-value results for the support percentages of different sequences in DM-2013 class is shown in Table 8. The significant values with  $p < 0.05$  are marked as bold, while edge cases with  $p < 0.1$  are marked in italics. We only included the significant and edge-case patterns and the transitions between platforms in the table. As these results show, the distinction students are significantly more likely to have a sequence of Piazza actions than the non-distinction group, with an average of 3% of their activities in the distinction group compared to 1% in the non-distinction group. The distinction students are also more likely to go to Piazza after a repetition of other activities than the non-distinction group. While the transition between WebAssign and Moodle (W+M, WM, M+W, MW) is high in both groups and not significantly different, the distinction group is more likely to move between Piazza and WebAssign (PW, WP, W+P, P+W) on average 0.4% compared to 0.01%.

Table 8: KW p-values between distinction and non-distinction students for different sequence supports in DM-2013 (W = WebAssign, M = Moodle, P = Piazza)

N-gram	Avg in Distinction	Avg in Non-distinction	KW pvalue
+P	0.0018	0.0001	<b>3.31E-03</b>
P-W transitions	0.0039	0.0006	<b>3.08E-03</b>
P+	0.0303	0.0133	<b>1.30E-05</b>
M-W transitions	0.0429	0.0380	0.644608

### 5.2.2 DM-2015

The KW p-values for the different sequences between the distinction and non-distinction group are shown in Table 9. Similar to the previous offering, the distinction group in this class are also more likely to have consequent Piazza activities, as well as go to Piazza after consequent actions on another platform. They are also more likely to move between Moodle and Piazza, while the non-distinction group is more likely to perform consequent actions on Moodle.

Table 9: KW p-values between distinction and non-distinction students for different sequence supports in DM-2015 (M = Moodle, P = Piazza)

N-gram	Avg in Distinction	Avg in Non-distinction	KW pvalue
P-M transitions	0.003	0	<b>1.93E-03</b>
+P	0.001	0	<i>5.82E-02</i>
M+	0.913	0.966	<b>5.80E-05</b>
P+	0.081	0.034	<b>1.18E-04</b>

### 5.2.3 Java-2015

The KW p-values for the different sequences between the distinction and non-distinction groups are shown in Table 10. Similar to the prior classes, the distinction group in this class was also more likely to go to Piazza after consequent actions on other platforms. Also, similar to DM-2015, the transitions between Moodle and Piazza are significantly more likely among the distinction group. Also, the distinction group has significantly more consequent actions on Github than the non-distinction group. However, while on average more sequences have a repetition of Piazza activities among distinction students, this difference is not significant in this class. Similarly, moving between Github and Moodle is more likely on average among the non-distinction group, but this difference is also not significant.

Table 10: KW p-values between distinction and non-distinction students for different sequence supports in Java-2015 (G = Github, M = Moodle, P = Piazza)

N-gram	Avg in Distinction	Avg in Non_distinction	KW pvalue
M+	0.566	0.655	<b>0.046</b>
P-M transitions	0.004	0.003	<b>0.022</b>
+P	0.003	0.002	<i>0.052</i>
P+	0.011	0.010	<i>0.095</i>
G+	0.294	0.204	<b>0.005</b>
G-M transitions	0.041	0.043	0.788
P-G transitions	0.003	0.003	0.336

### 5.2.4 Java-2016

The KW p-values for the different sequences between the distinction and non-distinction groups are shown in Table 11. Similar to the previous classes, in this class also we observe more repetitions of Piazza activities in the distinction group as well as more Piazza activities after a repetition of activities on other platforms. Also, similar to the 2015 Java offering and DM-2015, the non-distinction group is more likely to have consequent actions on Moodle. Despite the other classes, transitions between Github and Moodle as well as Github and Piazza are significantly different in this class and more likely for the distinction group. Comparing these results to the ones in Table 7, the findings seem conflicting since the non-distinction group is more likely to have Moodle activity in the same session after Github activities. However, we need to note that the Apriori algorithm, unlike N-grams, calculates the possibility of Moodle actions occurring after, but not necessarily consequently after, the Github activities. So, it seems like that the non-distinction group are more likely to move to Moodle at some point of the session after Github activities, but less likely to do so consequently after the Github actions.

Table 11: KW p-values between distinction and non-distinction students for different sequence supports in Java-2016 (G = Github, M = Moodle, P = Piazza)

N-gram	Avg in Distinction	Avg in Non_distinction	KW pvalue
M+	0.6976	0.7822	<b>1.30E-05</b>
+P	0.0021	0.0010	<b>1.97E-02</b>
P+	0.0123	0.0048	<b>1.12E-03</b>
G-M transitions	0.0616	0.0521	<b>3.98E-02</b>
P-G transitions	0.0046	0.0010	<b>3.20E-04</b>
P-M transitions	0.0026	0.0023	0.53

## 6. DISCUSSION

While the classes we analyzed and the offerings within them differ in topic, materials, structure, and instructor approach, our analysis shows that there are common patterns across all of them.

The first visible pattern is that the students are much more likely

to complete consecutive actions on the platform they are already using rather than switching to another platform. In all of the classes, the most common trend is two or more actions on WebAssign followed by Moodle in DM-2013, and Moodle followed by Github in the Java classes, while repetitions of Piazza actions seem to be more rare, even compared to platform switches. This might be due to the fact that most of the activities on Moodle, Github, and WebAssign consist of a sequence of smaller actions. For example, the students are much more likely to solve several problems on WebAssign or attempt a single problem several times, rather than only making a single attempt and leaving the platform. Similarly, on Moodle, the students often need more than one click to reach the material they need to access and on Github, the students are likely to push their code, face a failing test on Jenkins, and make a new commit to solve that issue. However, the actions on Piazza are not as closely monitored. On this platform, only making posts and replies are logged and viewing the posts or replies are not. Thus, the students are much more likely to make a single post or reply without any other visible actions on this platform and that might be a reason why consecutive Piazza actions are not as common as the other tools.

Another common pattern is that in contrast to our expectations, the students in all of the classes were much more likely to go back to the class material and the assignment descriptions on Moodle rather than rely on the discussion forum after one or more tries on their assignments. This was illustrated by the high confidence for transitions from WebAssign and Github (i.e. the submission systems) to Moodle (i.e. the indirect support platform), compared to transitions from these platforms to Piazza (i.e. the direct support platform), even in the higher performing students. As we expected, the visible trends for WebAssign and Github are similar in these classes due to the similarity in their educational role. As mentioned before, since the views are not monitored on Piazza, it is therefore possible that in some cases the students do refer to Piazza posts, only to find their answers in another student's question, without making any posts or replies. Thus, the lower amount of transitions to Piazza might be due to this difference in recording the activities. However, the teaching staff often found that the students did not look for their questions in their peers' posts and kept asking similar questions.

While both the performance groups have a large amount of consecutive Moodle actions, the non-distinction groups have on-average more of such sequences and this difference is often significant in these classes. Also, having repetitive Piazza actions and going back to Piazza after two or more actions on another platform is, on average, more common between the distinction students and this difference is significant in most of the classes, while in other classes an edge case that could be significant if we considered  $p < 0.1$ . This shows that while the non-distinction group seems to insist on finding the answer among the class material (or possibly reading the existing posts on Piazza), the distinction group seems to ask or answer questions on the discussion forum more often.

## 7. CONCLUSIONS

While multiple researchers have applied sequence analysis to educational data, most of this research has been focused on ITS data or MOOC data and there is not much research on the transitions of students between several resources in blended courses. In this study, we gathered logs from several online platforms that students interacted with in two offerings of two



undergraduate courses. We extracted sessions of studies among these activity logs and analyzed the sequences of the student actions in these sessions to find the general patterns in student transitions as well as the patterns that distinguish between the higher performing students and low-performers.

Our results show that consequent actions on the same platform are more likely for the students. Additionally, students are more likely to refer to the class material and the assignment descriptions rather than the discussion forum after a couple of submissions on assignments. However, the higher performers generally had more transitions between platforms and were often more likely to go to the discussion forum than the non-distinction group. We also found that even though some platforms used in classes are different, the results can be generalized across classes as long as the tools play similar educational roles, as WebAssign and Github did in our case. This can help findings to be expanded across a variety of courses using different platforms.

The results of this study can also help instructors identify helpful and harmful patterns among students and offer suggestions for forming more productive habits. The frequencies of these sequences added to the previously defined behavioral features can also help researchers improve the performance of their prediction models on student performances.

One limitation of this study is the differences between the length of the activities and how they are recorded on the different tools. Some types of activities are shorter and thus, more likely to repeat, such as WebAssign submissions where the questions are often multiple answers and quick to submit, while some other activities take a longer time, such as writing a Piazza post or solving an issue with the code and making a new commit. Additionally, while Moodle platform logs every action the users make online, Piazza only records the posts and replies and not the views. These differences in the tools might affect our findings. Further analysis, such as considering the time between actions differently for different tools might help us understand the trends in student activities better. Also, the WebAssign action logs are not available for the DM-2015 class, which limits the findings for this class and makes the comparisons between the two DM offerings less significant. Adding later similar offerings of these courses to the study in the future might help in finding more consistent trends.

In the future, we plan to expand the study to use different sequence analysis tools, such as the differential sequence mining tools. Those tools might be able to highlight other differences among the performance groups that are more difficult to spot using the current tools. Also, replicating our analysis on other courses and more offerings of the same courses can give us a better insight on how general some of these findings are. In the end, we plan on extracting predictive features from the student transitional patterns and add them to the other behavioral features to improve the accuracy of the performance prediction models on students, make the models fit better across classes, or make them fit better for earlier predictions in the semester.

## 8. ACKNOWLEDGEMENTS

This research was supported by NSF 1821475 “Concert: Coordinating Educational Interactions for Student Engagement” Collin F. Lynch, Tiffany Barnes, and Sarah Heckman (Co-PIs).

## 9. REFERENCES

- [1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.
- [2] B. Amnueypornsakul, S. Bhat, and P. Chinprutthiwong. Predicting attrition along the way: The uiuc model. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 55–59, 2014.
- [3] J. M. L. Andres, R. S. Baker, G. Siemens, C. A. Spann, D. Gašević, and S. Crossley. Studying mooc completion at scale using the mooc replication framework. 2016.
- [4] M. S. Boroujeni and P. Dillenbourg. Discovery and temporal analysis of latent study patterns in mooc interaction sequences. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 206–215. ACM, 2018.
- [5] C. Brooks, C. Thompson, and S. Teasley. A time series interaction analysis method for building predictive models of learners using log data. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, LAK '15, pages 126–135, New York, NY, USA, 2015. ACM.
- [6] Y. Chen and M. Zhang. Mooc student dropout: Pattern and prevention. In *Proceedings of the ACM Turing 50th Celebration Conference - China*, ACM TUR-C '17, pages 4:1–4:6, New York, NY, USA, 2017. ACM.
- [7] M. Desmarais and F. Lemieux. Clustering and visualizing study state sequences. In *Educational Data Mining 2013*, 2013.
- [8] L. Faucon, L. Kidzinski, and P. Dillenbourg. Semi-markov model for simulating mooc students. In *EDM*, pages 358–363, 2016.
- [9] M. Fei and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 256–263. IEEE, 2015.
- [10] C. Geigle and C. Zhai. Modeling mooc student behavior with two-layer hidden markov models. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 205–208. ACM, 2017.
- [11] J. Guerra, S. Sahebi, Y.-R. Lin, and P. Brusilovsky. The problem solving genome: Analyzing sequential patterns of student work with parameterized exercises. In *Educational Data Mining conference*, July 2014.
- [12] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Experiences and best practices in and around MOOCs*, 7:3–12, 2014.
- [13] J. Herold, A. Zundel, and T. F. Stahovich. Mining meaningful patterns from students’ handwritten coursework. *Proceedings of EDM*, 2013.
- [14] H. Jeong and G. Biswas. Mining student behavior models in learning-by-teaching environments. In *Educational Data Mining 2008*, 2008.
- [15] J. S. Kinnebrew and G. Biswas. Identifying learning behaviors by contextualizing differential sequence mining with action features and performance evolution. *International Educational Data Mining Society*, 2012.
- [16] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students’ learning behavior patterns. *JEDM | Journal of Educational Data Mining*, 5(1):190–219, 2013.
- [17] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing

- disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 170–179. ACM, 2013.
- [18] S. Klingler, T. Käser, B. Solenthaler, and M. H. Gross. Temporally coherent clustering of student data. In *EDM*, pages 102–109, 2016.
- [19] M. Köck and A. Paramythis. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*, 21(1-2):51–97, 2011.
- [20] V. Kovanovic, D. Gasevic, S. Dawson, S. Joksimovic, 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, LAK '15, Poughkeepsie, NY, USA, March 16-20, 2015*, pages 184–193, 2015.
- [21] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.
- [22] X. Li, T. Wang, and H. Wang. Exploring n-gram features in clickstream data for mooc learning achievement prediction. In *International Conference on Database Systems for Advanced Applications*, pages 328–339. Springer, 2017.
- [23] R. M. Maldonado, K. Yacef, J. Kay, A. Kharrufa, and A. Al-Qaraghuli. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In *Educational Data Mining 2011*, 2011.
- [24] P. Mukala, J. Buijs, and W. Van Der Aalst. Exploring students’ learning behaviour in moocs using process mining techniques. *Department of Mathematics and Computer Science, University of Technology, Eindhoven, The Netherlands*, pages 179–196, 2015.
- [25] N. Patel, C. Sellman, and D. Lomas. Mining frequent learning pathways from a large educational dataset. *arXiv preprint arXiv:1705.11125*, 2017.
- [26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [27] B. Pursel, L. Zhang, K. Jablolkow, G. Choi, and D. Velegol. Understanding mooc students: Motivations and behaviours indicative of mooc completion. *J. Comp. Assist. Learn.*, 32(3):202–217, June 2016.
- [28] C. P. Rosé, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sherer. Social factors that contribute to attrition in moocs. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 197–198. ACM, 2014.
- [29] A. Sheshadri, N. Gitinabard, C. F. Lynch, T. Barnes, and S. Heckman. Predicting student performance based on on-line study habits: A study of blended courses. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM) 2018, Buffalo, US*, pages 411–417, 2018.
- [30] B. Shih, K. R. Koedinger, and R. Scheines. Unsupervised discovery of student strategies. In *Educational Data Mining 2010*, 2010.
- [31] T. Sinha, N. Li, P. Jermann, and P. Dillenbourg. Capturing” attrition intensifying” structural traits from didactic interaction sequences of mooc learners. *arXiv preprint arXiv:1409.5887*, 2014.
- [32] M. Wen and C. P. Rosé. Identifying latent study habits by mining learner behavior patterns in massive open online courses. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1983–1986. ACM, 2014.
- [33] D. Yang, R. Kraut, and C. P. Rosé. Exploring the effect of student confusion in massive open online courses. *Journal of Educational Data Mining*, 8(1), 2016.
- [34] D. Yang, T. Sinha, D. Adamson, and C. P. Rosé. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, volume 11, page 14, 2013.