Annotated Examples and Parameterized Exercises: Analyzing Students' Behavior Patterns

Mehdrad Mirzaei[1](✉), Shaghayegh Sahebi[1], and Peter Brusilovsky[2]

[1] Department of Computer Science, University at Albany - SUNY, Albany, USA
{mmirzaei,ssahebi}@albany.edu

[2] School of Computing and Information, University of Pittsburgh, Pittsburgh, USA
peterb@pitt.edu

Abstract. Recent studies of student problem-solving behavior have shown stable behavior patterns within student groups. In this work, we study patterns of student behavior in a richer self-organized practice context where student worked with a combination of problems to solve and worked examples to study. We model student behavior in the form of vectors of micro-patterns and examine student behavior stability in various ways via these vectors. To discover and examine global behavior patterns associated with groups of students, we cluster students according to their behavior patterns and evaluate these clusters in accordance with student performance.

Keywords: Student sequence analysis · Frequent pattern mining

1 Introduction

With the improvement of online learning systems, students are provided with more opportunities for learning. In modern learning systems, students are usually free to choose to access multiple learning material types. While some systems provide restrictions on order of accessing learning content, in many systems, there are no predefined activity sequences and students are free to choose to work with any learning materials in any order. This choice provides students with more freedom to learn according to their own pace and background knowledge, and to repeat their past activities as they seem fit. For example, students can skip some learning materials and work on the more advanced ones if they believe that they have already mastered the prerequisite concepts. Similarly, they can go back and repeat some learning materials.

Despite these advantages, this freedom could lead to some inefficient and non-productive behavior. For example, past research on students' problem-solving behavior has found that students tend to practice the same set of concepts, well after mastering them, instead of moving to new concepts and more difficult problems [7,10,19]. While past research on behavior patterns has mainly focused on problem-solving behavior, student behavior can get more complex as other types

S. Isotani et al. (Eds.): AIED 2019, LNAI 11625, pp. 308–319, 2019.
https://doi.org/10.1007/978-3-030-23204-7_26

of learning materials are introduced. For example, consider a learning system that includes both reading materials and self-assessment problems. Here, a student can spend a significant amount of time persisting in reading on an advanced concept and failing in related problems, without having the prerequisites.

Ideally, a learning system should be able to detect inefficient behavior and guide the students towards efficient ones. To do this, the main challenge is to understand the relationship between students' behavior and performance. This challenge, translates into two main questions: (1) could we discover stable student behavior patterns which could be recognized in-time to react? are them persistent, or they happen at random; and (2) could we recognize efficient and inefficient student behavioral patterns by associating it with their learning performance?

Past research in the area of student problem-solving behavior indicated that stable patterns of student behavior do exist, however, theses patterns might not be directly related to their performance [7]. Instead, different patterns characterize students' individual ways to learn and approach a problem. To find stable patterns of student behavior, Guerra et al. [7] built student "problem-solving genomes" from micro-patterns ("genes") and grouped the students based on their "genomes" into clusters. While students belonging to the same cluster tend to show the same behavior patterns, these clusters included both high and low performing students. However, there was an indication that within each cluster, the "genomes" could help to discover efficient and inefficient behaviors.

In this paper, we attempt to apply the sequence mining-approach suggested in [7] to a more complex case, where students are working with two types of learning material, and are repeating their attempts within the same topics. The two learning material types we focus on are: parameterized problems and annotated examples. Our research questions within this more complex context are still the same: (1) do individual students exhibit stable behavioral patterns in their work with learning content, or their approach to learn vary by factors, such as time in the semester or learning material difficulty? (2) to what extent student behavioral patterns are associated with their learning performance?

2 Related Work

Online educational systems collect increasing volumes of information from students' interactions with various kinds of learning content. The process of data collection has been followed by a rapid increase in research, which focused on using this data to better understand and improve the learning process. Among the explored topics was the early prediction of student success or failure [2], which could be helpful to identify and support students-at-risk [20]. While early work focused mostly on cumulative factors such as frequency of watching videos or using discussion forums [18], recent work attempted to build more complex models of student behavior and identify various kinds of behavior patterns to help students make progress and improve education outcomes. Analyzing students' sequences and trajectories have been of increased interest recently.

Since student behavior is commonly considered as a sequence of students' actions or interactions with the system, various kind of sequence-oriented Markov models were explored for behavior analysis. For example, Hansen et al. [8] analyzed log data of an online education system, and modeled student behavior as interpretable Markov chains. The model compares action sequences across different lengths, focusing on the flow of actions. A two-layer Hidden Markov Model is used in [5] to automatically detect student behavior patterns from logged data of a MOOC platform. They have shown that the extracted patterns are meaningful and have a correlation with students' learning outcome. In such works, all of the students' activities are observed to generate latent states.

Another popular group of approaches used matrix factorization to transform a student learning traces to a smaller number of "soft clusters". For example, Gelman et al. [6] segmented student use of content and assessment, into weeks. Then, they used non-negative matrix factorization (NMF) to distill five basic behaviors of students ("Deep", "Consistent", "Bursty", "Performance", and "Response") and built vectors specifying how much each student shows each behavior. Similar approaches were used successfully with different kind of data in [13,15] that can be incorporated with ranking techniques from social networks [3].

Some work in the literature focused on student trajectories [11,21]. For example, Boubekki et al. [1] compared the navigation behavior of students in reading textbooks and discovered student clusters that were indicators of student performance. Sawyer et al. [17] proposed a time-series representation of student problem-solving trajectories in a learning game. They used Euclidean distance and trajectory slope to measure students' distance with "expert paths", which was correlated with students' learning gain.

The least explored, but potentially very powerful group of approaches focused on using sequence mining to identify behavior patterns. In [14], authors have extracted frequent action sequences in a collaborative learning environment to distinguish high achieving student from low achieving students. The analysis in this work is on interaction with resources on a tabletop. Students' actions on the tabletop are logged and coded into events to create sequences. Then, frequent sub-sequence sequential patterns are extracted using n-grams. The patterns are clustered to build higher-level patterns and students are compared based on them. In [7] patterns of student work with parameterized exercises are modeled and analyzed. In this work, micro-patterns are extracted using a sequential pattern mining algorithm and used to build student behavior profiles ("genomes"). Then, students with similar genomes are clustered into behavior groups.

3 System and Dataset

In our experiments, we use the student interaction data with learning materials in the "Introduction to object-oriented programming" course using *Progressor+* interface [9]. The system includes parameterized exercises and worked-out code examples as two different types of learning material. The learning material in the practice system were grouped into topics. For each topic, multiple problems and examples are available. Although the order of topics was shown to the students, they could choose any topic, problem, or example to practice at any time in any preferred order. The parameterized exercises are small problems focused on program behavior prediction. Each exercise is a template with a parameter, which is generated randomly every time a student chooses to work on them. Consequently, students can use the same exercise template for practice multiple times with different parameters. Worked-out annotated code examples are small complete programs annotated with short explanations for each line of code. Students can click on the lines of code in any order to read the explanation. The dataset includes three semesters of student activities in Java classes in a large public US university. After data cleaning, the dataset contains 83 students. There are 103 parameterized problems and 42 annotated examples in the dataset. The number of correct attempts to solve problems is 13796, the number of incorrect attempts is 6233, and the number of clicks on examples is 12713. In addition to the student behavior log, the dataset includes pre-test and post-test scores for each student. The pre- and post-tests included the same set of program behavior prediction questions administered at the beginning and at the end of each semester correspondingly. The minimum and maximum score in pre-test are 0 and 14 and in post-test are 5 and 24 respectively. To measure students' improvement over the course of the semester, learning gain is calculated for each student as the normalized difference between post-test and pre-test.

4 Modeling Student Behavior

To extract micro-patterns from student logs, we code them into sequences and analyze them using a frequent pattern mining algorithm. We build macro-pattern vectors or *genome* as a representation of each individual student's behavior.

4.1 Coding Student Behavior

To discover behavioral patterns of students, we first label student attempts. Inspired by [7], we focus on two aspects when labeling student problem solving attempts: whether the student succeeds (or fails) in solving the problem, and whether the student spent shorter or longer time to answer a problem, compared to a median answering time. The median answering time is calculated separately for each problem, considering all attempts on it[1]. If a student solves a problem correctly in less time than the median, the attempt is labeled as 's' (short success). Likewise, if a student's successful attempt takes longer than the median it will be labeled as 'S' (long success). Similarly, if the student solves a problem incorrectly in a short time (vs. long time), her attempt will be labeled as 'f' (vs. 'F'). In total, the dataset included 760 short successes, 6030 long successes, 2242 short failures, and 3991 long failures.

In addition to students' problem-solving, we code their example-reading behavior. Unlike the problem-solving attempts, working on annotated examples is not associated with correctness. Thus, we only measure the time spent by each student

[1] The median split can be calculated within each students also. Since we are interested in capturing content access differences between students, and since time-spent variance among problems is larger than among students, we chose to split the data according to problem-answering medians.

on each annotated example. To do this, we sum up all sequential student clicks on example lines of one annotated example, as the time spent on that example. We first calculate the median time spent on each annotated example by all students. For each example-reading activity of a student, if the time spent on the annotated example is less than its median, it is labeled ‘e’, otherwise as ‘E’. The dataset included 6348 short example attempts and 6365 long attempts.

The students continue to work in the system during the whole semester. To chunk the large sequence of student actions into smaller comparable sequences, we define a “session” as a consecutive set of student activities within one topic. In other words, a session is a sequence of attempts on parameterized problems and annotated examples inside the same topic. An attempt on an example or problem from another topic starts another session. To indicate session borders within each student’s sequence, we insert ‘_’ between two consecutive sessions. For instance, the sequence ‘_ffSsee_’ means that the student has a long success after two short failures, then a short success and finally is quickly examined two annotated examples within the same session.

4.2 Sequential Pattern Mining

To discover the most frequent micro patterns of student behavior, we use CM-SPAM [4] sequential pattern mining algorithm. We set the minimum support to 1% (i.e., we are interested in patterns that could be found in at least 1% of sequences) and require no gap between encoded attempts. Besides that, we only consider the patterns with more than one sequential attempts. In total, 111 frequent patterns are discovered using this approach. The top 30 frequent patterns are illustrated in Table 1.

Interestingly, we can see that the top frequent patterns are either problem-solving micro-patterns or example-reading micro-patterns. In other words, there are no mixed activity patterns (such as ‘eF’) among the top frequent ones. From this, we conclude that switching from one type of activity to another was considerably more rare than continuing with the same kind of activity.

Table 1. Top 30 extracted patterns ordered by support

	Pattern	Support		Pattern	Support		Pattern	Support
1	ss	1516	11	Fs_	680	21	_FF	486
2	Ss	1456	12	Ff	680	22	_Sss	449
3	ss_	1378	13	sss_	668	23	FS_	449
4	Fs	1153	14	Sss	663	24	_Ss_	443
5	_Ss	974	15	_FS	630	25	_ee	431
6	_Fs	901	16	ee	593	26	fs_	393
7	FS	828	17	ee_	552	27	ssss	373
8	fs	788	18	FF	546	28	ff	367
9	sss	735	19	_Fs_	539	29	Fss	361
10	Ss_	692	20	_Ff	515	30	_FS_	351

4.3 Building Pattern Vectors

The top frequency patterns found in Sect. 4.2 represent a variety of patterns used by all students. Each student could use each micro-pattern with different frequency or not at all. To model the behavior of an individual student, we build a behavioral pattern vector for each student. We use the top 60 of frequent patterns to build this vector. This pattern vector includes normalized frequencies of observing each of the top frequent patterns in the behavior log of the modeled student. To build it, we first count the number of times each frequent pattern occurred in the student's sequence. These *absolute* frequencies, however, could vary depending on the total length of student behavior sequence, i.e., how much the modeled student interacted with the system. To capture the relative importance of each micro-pattern regardless of total sequence length, we normalized the count vectors (i.e., the frequency of patterns are summing to one for each vector). These vectors represent the behavior of individual students and are used to discover macro-patterns by clustering student vectors.

5 Behavior Stability Analysis

Before establishing a relationship between students' micro-patterns and their performance, we should make sure that the patterns are representative of students' behavioral traits, and not other environmental factors. To do this, we analyze the *stability* of student patterns in three different setups: randomized, longitudinal, and complexity-based. In each of these setups, we split student sequences into two equal sets. Then, we independently build a pair of two pattern vectors for each student: one for each set. If our model of student behavior is stable, the vectors in each pair should be more similar to each other than to vectors form other pairs. Thus, in each of the setups, we test whether the students' behavior vector built from the first set is significantly more similar to their own behavior vector in the second set than to the behavior vectors of the rest of the students. To measure the similarity, we use Jensen-Shannon divergence [12].

Table 2. Comparing average of students' pattern vector distances with themselves vs. other students according to various splits

	Self distance		Distance to others			
	Mean	SE	Mean	SE	t Stat	P-value
Random split	0.2082	0.0207	0.4639	0.0105	−16.0279	<0.0001
First half/second half	0.2995	0.0211	0.5207	0.0113	−12.3501	<0.0001
Random split (Easy)	0.3644	0.0258	0.5769	0.0110	−9.9099	<0.0001
Random split (Medium)	0.3266	0.0246	0.5465	0.0092	−11.1404	<0.0001
Random split (Hard)	0.4219	0.0266	0.5703	0.0106	−6.4266	<0.0001

5.1 Randomized Analysis

In the randomized analysis, our goal is to examine whether a student's pattern vector is stable across all sessions, if we split them randomly. It is to test if we can distinguish a student from other students according to their pattern vectors. To do this, we randomly split student sequences into two halves and build a pattern vector for each half. If a student's pattern vector from the first half is significantly more similar to her own pattern vector in the second half – compared to being similar to other students – then, we conclude that the student's patterns are stable and do not change randomly. To test the significance, we run paired sample t-test. The results are shown in Table 2. As we can see, the distance between the two pattern halves for the same student (0.2082 on average) is significantly smaller than the distance to other students (0.4639).

5.2 Longitudinal Analysis

Here, we are interested to see if the student patterns change as the semester advances. To study this, we split each student's activity sequence according to a mid-semester point: we build pattern vectors for the first half and the second half of the semester. Similar to randomized analysis, we compare the distance between halves within each student's vector and between student vectors. As shown in Table 2, we see that the distance between first half and second half of one student is significantly smaller than the distance to other students. Individual student behavior pattern changes slightly over the semester, yet this change is by far not sufficient to cross the difference from other students.

5.3 Complexity Analysis

Another factor that can affect students' behavior is activity complexity. Each learning material is labeled with "easy", "medium" or "hard" in our dataset. Accordingly, we build separate pattern vectors for each group of learning activities for each student. E.g., in each topic and session, we separate the "easy" problems and examples as one "easy" session. We assess pattern vector stabilities by comparing the difference within a student (comparing according to complexity) and between students. Table 2 represents the distances and statistical tests that show student pattern vectors are stable across learning material complexities.

6 Behavior Cluster Analysis

Having stable student pattern vectors, we aim to distinguish efficient patterns. To do this, we study if student behavioral patterns are associated with student performance. Namely, we would like to understand if students with similar behavior have a similar performance. First, we cluster the students based on their behavior patterns to have students with similar patterns together. Afterward, we analyze the patterns to recognize useful patterns in each cluster.

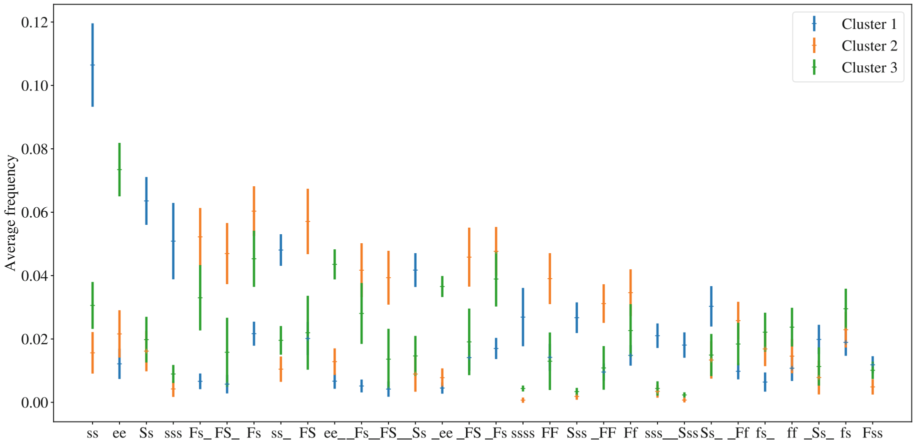

Fig. 1. Top 30 patterns and their frequencies in 3 clusters. Patterns are ordered by the maximum difference of frequencies between three clusters.

6.1 Pattern Analysis

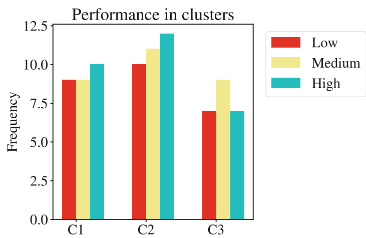

Fig. 2. High, medium, and low-performance student frequencies in three clusters generated based on student patterns

We apply Spectral clustering [16] on student pattern vectors. Our cluster and interpretation analysis showed that 3 clusters will provide the best results with 23, 28, and 33 students in each cluster. To understand the student differences in each cluster, we compare their average pattern frequencies in the top 30 micro-patterns. Figure 1 shows these top 30 frequent patterns and their average frequencies in each cluster. The error bars show 95% confidence interval for each micro-pattern. The micro-patterns are ordered based on the maximum frequency difference in three clusters. As we can see in the figure, in cluster 1, patterns like 'ss', 'Ss' and 'sss' are significantly more frequent than the other two clusters. We can say, students in this cluster tend to repeat practicing an exercise within a topic even if they succeed in it. Significantly frequent patterns in cluster 2 are '_FS_', 'FS' and 'FS_' that demonstrate longer failures and longer successes afterwards. We can conclude that the students in this cluster tend to spend more time on solving a problem, and then succeed afterwards. They do not attempt the problems randomly and do not answer them by chance. Students in cluster 3 read more examples since patterns such as 'ee' and 'ee_' are frequent in this cluster. In [7] students were grouped into "confirmers" and "non-confirmers" according to their patterns. "confirmers" were the students who preferred to confirm their success by repeating it. "non-confirmers" were the ones who ended their session right after having a short success. Here, we see a "confirmers" type of pattern in cluster 1. However, in cluster 2 students are mostly “thinkers” rather than “non-confirmers”. They fail and then succeed, but with thinking and spending time on the activity. Cluster 3 students are mostly “readers”. They tend to spend more time on reading the annotated examples. Therefore, we can recognize 3 types of behaviors with the extracted patterns. We should mention that these labels are provided to distinguish the discovered clusters, rather than exactly describing student behaviors in them.

6.2 Performance Analysis

Here we examine the clusters to detect whether we can associate the macro-behavioral patterns represented by each cluster with students' learning performance, measured by normalized learning gain. Figure 2 shows the number of students with low, medium and high performance (learning gain) in each cluster. As we can see (also by our statistical tests) the clusters do not show a significant difference in the number of high, medium, or low performance students. The similar conclusion holds for pre-test and post-test performance of students. We can conclude that the macro-patterns represented by the clusters are neither related to students' past performance nor to their course-level performance. In other words, the patterns do not separate weak students from strong ones. Instead, they represent students' different approaches to work with learning content. Within the group of students using the same approach, however, we can find both strong and week students. The results are similar to the observation in the original paper [7].

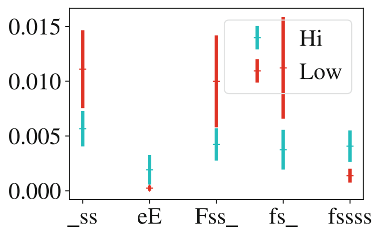

Fig. 3. Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 1

Next, we study the differences in behavioral micro-patterns of high and low-performance students within each cluster. By this, we hope to uncover the efficient and inefficient micro-patterns that happens within students with the same studying traits. To achieve this, we examine the average frequencies of micro-patterns for low and high performance students in each cluster and select the ones with a significant difference. The results are shown in Figs. 3, 4 and 5. As presented in Fig. 3, in cluster 1 (“confirmers”), patterns such as ‘fssss’ and ‘eE’ are found to be significantly more in high performance students. On the other hand, patterns ‘Fss_’, ‘Fs_’, and ‘_ss’ appear more in low performance students. According to this, we can conclude that, the “confirmer” group students do repeat their success. But their approach to this repetition determines their performance in the course: (1) repeat after an initial success (‘_ss’) is associated with weaker students; (2) more repetition after an initial failure (‘fssss’) is associated with stronger students, as short repetitions and quitting after failure (‘Fss_’ and ‘Fs_’) is associated with weaker students; and (3) repeat reading examples is associated with stronger students. We can see that in cluster

2 ("thinkers", Fig. 4), high performance students have patterns such as '_FF', 'FF', and 'Sss', while low performance ones have higher rate of patterns with short failure ('f') in them. This shows that high-performance thinkers think each time they try a problem, until it is sufficiently understood. In contrast, weaker students frequently try to guess and fail in solving problems. Interestingly, low-performance thinkers also have a high frequency of 'Fff' pattern. It can be concluded that they start with serious intentions, but then start to guess the answers.

For the "reader" students (cluster 3, Fig. 5), we see longer attempts (e.g., 'EE', '_FS_', and 'FS') for high-performance students, compared to shorter attempts (e.g., 'ffs' and 'Fs') for low-performance ones. We can see that (1) high-performance students work with examples more carefully; (2) they do not rush after failure, but think and most always get it right; and (3) in contrast, low-performance students do not spend enough time on their attempts, whether it is a success or failure. In general, having patterns that include long attempts among high performance students and short attempts in low-performance ones demonstrate the impact of spending time on the performance.

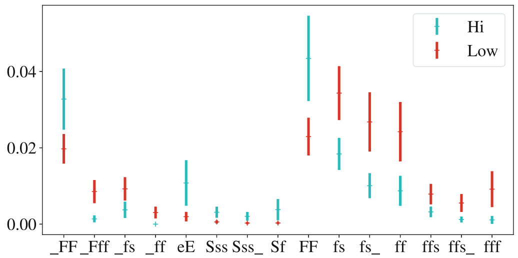

Fig. 4. Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 2

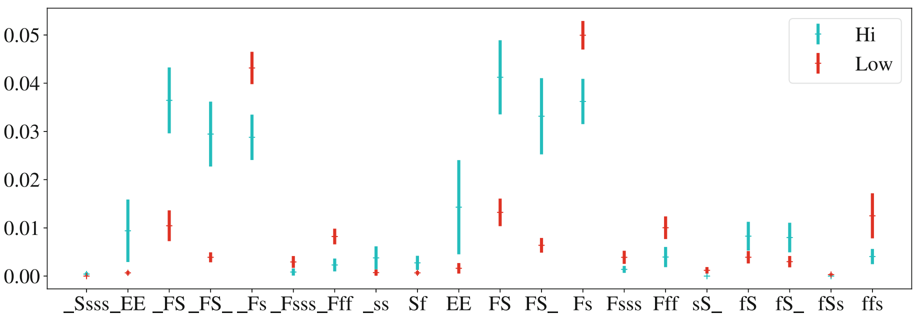

Fig. 5. Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 3

7 Conclusions

In this paper we analyzed students' behavior patterns in working with parameterized exercises and annotated examples. Using frequent pattern mining, we discovered frequent *micro-patterns* of student behavior and used them to construct *macro-pattern* behavior vectors for students. Using data driven approaches, we analyzed the stability of these macro-patterns and showed that these are results of students' behavioral traits. Clustering students according to these macro-patterns, we discovered three groups of students, which we nicknamed as "confirmers", "thinkers", and "readers". Among these groups, we identified students' efficient and inefficient micro-patterns by comparing frequent patterns of high and low-performing students. Our results suggested that for "confirmer" students, it is beneficial to encourage repetitions after they fail in solving a problem. But, repetitions after success is redundant and inefficient. For "thinkers", it is useful to encourage them to continue to think deeper each problem, even after failure. For "readers", working more carefully with examples and spending more time to think is beneficial. Being able to discover a few behavioral clusters that represent different ways of learning is a promising step towards personalization: if learning behavior diversity among students is not that large, we can nudge different student groups towards the optimal behavior in different ways. In future, these results can be extended to be used as encouragements or recommendations to help students of each group to take on more efficient behaviors.

Acknowledgement. This work is partially supported by the National Science Foundation, under grant IIS-1755910.

References

1. Boubekki, A., Jain, S., Brefeld, U.: Mining user trajectories in electronic text books. In: The 11th International Conference on Educational Data Mining (EDM 2018) (2018)
2. Boyer, S., Veeramachaneni, K.: Transfer learning for predictive models in massive open online courses. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS (LNAI), vol. 9112, pp. 54–63. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19773-9_6
3. Doan, T.-N., Chua, F.C.T., Lim, E.-P.: Mining business competitiveness from user visitation data. In: Agarwal, N., Xu, K., Osgood, N. (eds.) SBP 2015. LNCS, vol. 9021, pp. 283–289. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16268-3_31
4. Fournier-Viger, P., Gomariz, A., Campos, M., Thomas, R.: Fast vertical mining of sequential patterns using co-occurrence information. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014. LNCS (LNAI), vol. 8443, pp. 40–52. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06608-0_4
5. Geigle, C., Zhai, C.: Modeling student behavior with two-layer hidden markov models. J. Educ. Data Min. **9**(1), 1–24 (2017)

6. Gelman, B., Revelle, M., Domeniconi, C., Veeramachaneni, K., Johri, A.: Acting the same differently: a cross-course comparison of user behavior in MOOCs. In: Barnes, T., Chi, M., Feng, M. (eds.) The 9th International Conference on Educational Data Mining (EDM 2016), pp. 376–381 (2016)
7. Guerra, J., Sahebi, S., Brusilovsky, P., Lin, Y.: The problem solving genome: analyzing sequential patterns of student work with parameterized exercises. In: 7th International Conference on Educational Data Mining, pp. 153–160 (2014)
8. Hansen, C., Hansen, C., Hjuler, N., Alstrup, S., Lioma, C.: Sequence modelling for analysing student interaction with educational systems. In: The 10th International Conference on Educational Data Mining, pp. 232–237 (2017)
9. Hsiao, I.-H., Brusilovsky, P.: Motivational social visualizations for personalized E-learning. In: Ravenscroft, A., Lindstaedt, S., Kloos, C.D., Hernández-Leo, D. (eds.) EC-TEL 2012. LNCS, vol. 7563, pp. 153–165. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33263-0_13
10. Hsiao, I.-H., Sosnovsky, S., Brusilovsky, P.: Adaptive navigation support for parameterized questions in object-oriented programming. In: Cress, U., Dimitrova, V., Specht, M. (eds.) EC-TEL 2009. LNCS, vol. 5794, pp. 88–98. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04636-0_10
11. Kinnebrew, J.S., Segedy, J.R., Biswas, G.: Analyzing the temporal evolution of students' behaviors in open-ended learning environments. Metacognition Learn. **9**(2), 187–215 (2014)
12. Lin, J.: Divergence measures based on the shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145–151 (1991)
13. Lorenzen, S., Hjuler, N., Alstrup, S.: Tracking behavioral patterns among students in an online educational system. In: The 11th International Conference on Educational Data Mining (EDM 2018), pp. 280–285 (2018)
14. Martinez, R., Yacef, K., Kay, J.: Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In: Educational Data Mining 2011, pp. 111–120, June 2010
15. Mouri, K., Shimada, A., Yin, C., Kaneko, K.: Discovering hidden browsing patterns using non-negative matrix factorization. In: The 11th International Conference on Educational Data Mining (EDM 2018), pp. 568–571 (2018)
16. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, pp. 849–856 (2002)
17. Sawyer, R., Rowe, J., Azevedo, R., Lester, J.: Filtered time series analyses of student problem-solving behaviors in game-based learning. In: The 11th International Conference on Educational Data Mining (EDM 2018) (2018)
18. Sharma, K., Jermann, P., Dillenbourg, P.: Identifying styles and paths toward success in MOOCs. In: Santos, O., et al. (eds.) The 8th International Conference on Educational Data Mining (EDM 2015) (2015)
19. Sosnovsky, S., Brusilovsky, P., Lee, D.H., Zadorozhny, V., Zhou, X.: Re-assessing the value of adaptive navigation support in e-learning context. In: Nejdl, W., Kay, J., Pu, P., Herder, E. (eds.) AH 2008. LNCS, vol. 5149, pp. 193–203. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70987-9_22
20. Whitehill, J., Williams, J., Lopez, G., Coleman, C., Reich, J.: Beyond prediction: first steps toward automatic intervention in MOOC student stopout. In: The 8th International Conference on Educational Data Mining (EDM 2015) (2015)
21. Yin, C., Okubo, F., Shimada, A., Oi, M., Hirokawa, S., Ogata, H.: Identifying and analyzing the learning behaviors of students using e-books. In: 23rd International Conference on Computers in Education ICCE 2015. Asia-Pacific Society for Computers in Education (2015)