

Example Explorers and Persistent Finishers: Exploring Student Practice Behaviors in a Python Practice System

Allison Poh¹, Anurata Prabha Hridi², Jordan Barria-Pineda³, Peter Brusilovsky³ and Bitu Akram²

¹University of Massachusetts Amherst

²North Carolina State University

³University of Pittsburgh

Abstract

Understanding student practice behavior and its connection to their learning is essential for effective recommender systems that provide personalized learning support. In this study, we apply a sequential pattern mining approach to analyze student practice behavior in a practice system for introductory Python programming. Our goal is to identify different types of practice behavior and connect them to student performance. We examine two types of practice sequences: (1) *by login session* and (2) *by learning topic*. For each sequence type, we use SPAM (Sequential PAttern Mining) to identify the most frequent micro-patterns and build behavior profiles of individual learners as vectors of micro-pattern frequencies observed in their behavior. We confirm that these vectors are stable for both sequence types ($p < 0.03$ for session sequences and $p < 0.003$ for topic sequences). Using the vectors, we perform k -means clustering where we identify two practice behaviors: *example explorers* and *persistent finishers*. We repeat this experiment using different coding approaches for student sequences and obtain similar clusters. Our results suggest that *example explorers* and *persistent finishers* might represent two typical types of divergent student behaviors in a programming practice system. Finally, to better understand the relationship between students' background knowledge, learning outcomes, and practice behavior, we perform statistical analyses to assess the significance of the associations among pre-test scores, cluster assignments, and final course grades.

Keywords

Sequence mining, learning gain, behavior patterns, programming practice, computer science education

1. Introduction

Understanding how students use interactive computer science (CS) educational resources on online learning platforms and how this use shapes their learning is essential for developing efficient tools to support learning, such as personalized learning systems. Insight into this behavior could be gained by mining student activity logs, an approach widely used in numerous studies [1, 2, 3]. The results of such an analysis could inform decisions or support the development of predictive models.

Over the last 10 years, the educational data mining (EDM) community has developed a wide range of activity log mining approaches. Researchers have applied these approaches to various types of log data, including Massive Open Online Course (MOOC) learning behavior [4], blended learning across multiple platforms [5], problem-solving behavior [3], and course-taking patterns [6]. As new types of learning systems become popular, the log data accumulated by these systems offer new opportunities for research and potential new discoveries.

In this paper, we explore student learning behavior in a new type of learning system known as *practice system* [7, 8, 9]. These systems support student *free practice*, i.e., self-directed study in which students independently engage to gain skills in some domain or to complement their studies in regular classes. Unlike college classes and MOOCs, which combine knowledge delivery (lectures, textbooks, videos) with assessment (labs, assignments, exams), practice systems focus on learning through a combination of worked examples [10] and problem-solving. To support this approach, modern practice systems provide various types of interactive learning content with feedback and self-assessment. To examine

9th Educational Data Mining in Computer Science Education (CSEDM) Workshop, July 20, 2025, Sicily, Italy

✉ apoh@cs.umass.edu (A. Poh); aphridi@ncsu.edu (A. P. Hridi); jab464@pitt.edu (J. Barria-Pineda); peterb@pitt.edu (P. Brusilovsky); bakram@ncsu.edu (B. Akram)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

this relatively new type of learning data and uncover patterns in student practice behavior, we applied a sequential pattern mining approach. Focusing on student transitions between different activities, we uncovered two groups of students with divergent practice behaviors: *example explorers* and *persistent finishers*. These groups emerged consistently across two experiments using different sequence coding methods, suggesting that they may reflect recurring types of student behavior in free-practice systems that offer both worked examples and programming problems.

Finally, we conducted a series of statistical hypothesis tests to reveal patterns between students' background knowledge as demonstrated through their pre-test scores, learning behavior represented through their cluster assignment, and performance revealed through their final course grades. Our experiments showed a significant relationship between student learning behavior and performance.

2. Related Work

Analyzing student learning behavior through activity logs became a popular research topic following the rise of MOOCs [11, 12, 13, 3, 4]. On one hand, MOOCs provided an abundance of data to explore various data mining approaches. On the other hand, the low retention rate observed in early MOOCs challenged the research community. To understand learner behavior in MOOCs, many researchers focused their exploration of MOOC data on revealing student *behavior patterns*. Most prior studies on behavior pattern analysis have focused on resource usage (e.g., viewing course lectures and worked examples, answering quizzes, solving problems, participating in forums) to identify behaviors of different groups of students and relate those behaviors to high and low levels of learning [12, 14, 15]. While students have been observed to alternate between learning at the surface level (more effort in challenge completion) and going deep (more reliance on worked examples) [16], providing novice students with examples followed by similar practice tasks led to better learning [17, 18].

However, even the first generation of behavior analysis research suggested that focusing solely on resource usage might not lead to a reliable method to separate weak and strong students [19]. To address this problem, an increasing number of studies attempted to look deeper than how much of each activity type a student does by focusing on the order in which activities occurred. This shift enabled a deeper understanding of learning strategies and behavioral trajectories of students. Additionally, clustering and tracking students' activity timelines uncovered common behavioral patterns of engagement that evolve over a semester [20]. Building on this temporal perspective, several articles used relatively simple transition mining approaches [21, 4, 22] and reported interesting results. Nonetheless, more complex approaches such as sequential pattern mining gradually become more popular [23, 24, 6, 25, 26]. Sequential pattern mining is a group of machine learning techniques focused on finding time-related behavior in sequences. Its basic idea is to discover frequent subsequences (patterns) in a sequence database, in which each sequence is a time-ordered list of events [27]. In CS education, sequential pattern mining has been used to analyze a broad range of time-ordered data, including sequences of courses taken by students [6], sequences of student code-editing actions when solving construction problems [3, 28], and sequences of student attempts on code-tracing problems [24]. In [23], a productive approach to using sequential pattern mining in the educational context, known as differential sequence mining, was introduced in [23]. They used the SPAM method [29] to find common patterns in the sequences and applied statistical tests to check for differences in frequencies of those patterns among distinct groups.

In our work, we apply a combination of exploratory and differential sequence mining to analyze student sequences of work with different types of interactive learning content in a programming practice system. Unlike most previous studies, which focus on finding and differentiating individual patterns [23, 4, 6, 30], our work follows a more advanced approach suggested in [24], where individual *micro-patterns* are combined into frequency vectors to more reliably capture individual student behavior.

Table 1

Tokens detailing attempts.

Token	Activity Focus	Outcome	Attempt ($n > 1$)
sc1	construction	correct	1
scn	construction	correct	n
si1	construction	incorrect	1
sin	construction	incorrect	n
se1	construction	example	1
sen	construction	example	n
pc1	comprehension	correct	1
pcn	comprehension	correct	n
pi1	comprehension	incorrect	1
pin	comprehension	incorrect	n
pe1	comprehension	example	1
pen	comprehension	example	n

Table 2

Tokens detailing topic switching.

Token	Description
–	start of a topic
<	switching to a previous topic
>	switching to a subsequent topic

3. Dataset

Our study used activity log data from *Python Grids* [7], a practice system for introductory Python programming. The data is available via Carnegie Mellon University’s LearnSphere [31]. The system offers worked examples and practice problems across 15 core topics of a typical introductory Python course (e.g., variables and operations, if-else statements). Each logged activity corresponds to either an exploration of a worked example or an attempt at a practice problem within one of these topics. The system offers two types of problems: *construction* and *comprehension*. Construction problems focus on writing code (e.g., interactive code examples, coding from a prompt, filling in blanks, Parsons problems). Comprehension problems focus on interpreting and analyzing code behavior (e.g., animated code execution examples, code-tracing questions such as “What is the final value of x?”). Students can freely choose both the programming topics and problem types they wish to practice.

The dataset consists of anonymized activity logs from two sessions of the same undergraduate introductory Python course held in the summer of 2021 at a large public university. The first session contains 12,383 logged attempts (7,068 construction, 5,315 comprehension), and the second session 11,296 (6,294 construction, 5,002 comprehension). Although 174 undergraduate students were enrolled, using *Python Grids* for practice tasks was not a required component of the course. As a result, our dataset includes data from 41 students with no demographic information.

4. Methodology

4.1. Sequence Construction

To begin exploring student practice behaviors in *Python Grids*, we constructed sequences from its activity log data. Each attempt to access learning content (either an example or a problem) is encoded as a single token, and tokens are concatenated to form sequences. These tokens use three symbols to capture details of each learning action (Table 1): a type of practiced knowledge (‘s’ for conStruction, ‘p’

Table 3

Example of building a student vector from the top 5 most frequent micro-patterns using sequences *by learning topic*. A student vector is the average of all normalized frequency vectors for a student.

Topic	Frequency of Micro-Patterns					Total	Normalized Frequency Vector
	_, pe1	pe1, PEN	si1, sin	sin, sen	si1, SIN		
if-else	0	0	2	2	4	8	(0, 0, 0.25, 0.25, 0.5)
nested loops	0	1	2	0	1	4	(0, 0.25, 0.5, 0, 0.25)
while loops	2	2	1	0	0	5	(0.4, 0.4, 0.2, 0, 0)
$\mathbf{v} = (0.133, 0.217, 0.317, 0.083, 0.25)$							

for comPrehension), the nature of the action ('c' for Correct problem-solving attempt, 'i' for Incorrect problem-solving, 'e' for examining a step of a worked Example), and attempt number ('1' for first, 'n' for not first). Since we aimed to capture student transitions between different activities, we condensed long repetitions of attempts. A long repetition is defined as three or more consecutive actions within the same activity, based on the median repetition length, and was coded using uppercase letters. Additionally, three special tokens were used to represent topic switching (see Table 2). For example, in the sequence `_, si1, SIN, <, pe1, PEN`, the student begins a topic, made several attempts to solve a construction problem with at least two incorrect responses in a row, and then switches to an example from a previous topic, which the student examines two or more steps.

We explored creating sequences for two scenarios: *by login session* and *by learning topic*. In the *by login session* scenario, we created a separate sequence for each student for each time they logged into the system. In the *by learning topic* scenario, we created a separate sequence for each student for each topic they practiced. We applied the same methodology to each scenario independently.

4.2. Sequential Pattern Mining

We used the SPAM [29] sequential pattern mining algorithm to identify frequent sequences in our sequences. SPAM is an efficient breadth-first search strategy that has been successfully used to uncover behavioral patterns in educational datasets in prior studies [24]. To identify frequent and meaningful patterns, we focus on short sequences by defining a minimum and maximum pattern length of [2, 6] and limited our analysis to the top 50 most frequent sequences, which we call *micro-patterns*.

4.3. Clustering Students

To cluster students based on similar behavior patterns, we first represented each student as a vector derived from the frequencies of micro-patterns in their sequences. For each sequence, we created a 50-dimensional vector that captured the frequencies of the 50 most frequent micro-patterns identified by SPAM. To avoid biasing our data on the total amount of practice, which varies considerably between students, we focused on relative frequencies of frequent patterns in student behavior, i.e., each vector was normalized according to the respective student's overall number of attempts. We then averaged these vectors for each student to obtain a single vector that represented their overall behavior patterns. Table 3 illustrates this process using a small subset ($n = 5$) of frequent micro-patterns.

To ensure the consistency of our vectors, we checked their stability by splitting each student's sequence into two groups based on session number (*even* and *odd*). We then used Jensen-Shannon divergence to calculate two types of distances: the *self-distance* (the distance between even and odd sessions) and the *other-distance* (the distance between a student's even session and the even sessions of all other students). We performed a *t*-test on the difference of these distances to ensure stability. We then applied *k*-means clustering to group the vectors and identify behavioral patterns among students.

4.4. Mann-Whitney U Test

To investigate the relationship between students’ problem-solving behavior as represented through their cluster assignment, background knowledge, and performance, we conducted a series of statistical significance test analyses. We first ran two Mann-Whitney U tests to identify a potential relationship between background knowledge (pre-test scores) and behavior (cluster assignment), as well as performance (final course grade) and behavior.

We further hypothesized that students who are at the more extreme ends of the clusters (i.e., who are farther from the centroid of the other cluster) may represent more persistent and distinctive behavioral patterns, significantly affecting their performance. We also hypothesized that performance differences might exist between *extreme* and *moderate* members of each cluster. To evaluate these hypotheses, we first calculated the distance between each student’s behavioral vector and the centroid vector of the opposite cluster. We then divided the students in each cluster into *extreme* and *moderate* groups based on the median of their distance to the centroid of the opposite cluster. We then performed a set of Mann-Whitney U tests to evaluate the significance of proposed patterns.

The use of the Mann-Whitney U test in all scenarios is due to the lack of a normal distribution of performance data per group.

5. Experiments and Results

5.1. Experiments

We conducted two experiments at different granularities: *every attempt* and *every problem*.

In the *every attempt* experiment, we analyzed each logged activity made by students, creating tokens and sequences as described in the Methodology section. Each token represented either a distinct student attempt or a repetition of the same student attempt. This approach allowed us to capture detailed information about each step students took within the system, offering insights into their interactions at a fine-grained level. These sequences averaged 21 tokens in length.

In the *every problem* experiment, each token represented a single problem. This less granular representation resulted in more condensed sequences (on average, three less tokens), which further magnified student transitions between different activities. By running two experiments, we wanted to explore whether different levels of granularity in encoding sequences reveal different patterns.

5.2. Results and Discussion

5.2.1. Micro-Patterns

To identify frequent micro-patterns in our sequences, we used SPAM and selected the top 50 most frequent micro-patterns based on support values. Table 4 provides a sample of the top 10 micro patterns for each experiment. To assess the diversity of the most frequent 50 micro-patterns, we calculated the Gini coefficient for their frequency distribution (see Table 5). The distribution of micro-pattern frequencies in the *every attempt* experiment shows moderate equality. However, the consolidation of sequences in the *every problem* experiment led to a slightly more even distribution of micro-pattern frequencies, indicating greater diversity in the micro-patterns observed.

5.2.2. Student Vectors

To evaluate the stability of our student vectors, we used the Jensen-Shannon divergence to compute the self-distance and other-distance for each student. We then performed a *t*-test to compare these distances, as described in the Methodology section. The results are summarized in Table 6. In all cases, the self-distance is significantly smaller than the other-distance, showing that students’ behavior is more similar within their own interactions than compared to others. Furthermore, the Cohen’s *d* values indicate a high degree of consistency in how students engage with the system across different topics or

Table 4

Top 10 most frequent micro-patterns for experiments *every attempt* and *every problem*.

	<i>every attempt</i>				<i>every problem</i>			
	<i>by login session</i>		<i>by learning topic</i>		<i>by login session</i>		<i>by learning topic</i>	
	pattern	supp.	pattern	supp.	pattern	supp.	pattern	supp.
1	_ , <	101	_ , pe1	134	_ , <	101	_ , pe1	58
2	si1 , sin	79	pe1 , PEN	90	sin , scn	42	sin , scn	51
3	sin , scn	70	si1 , sin	89	_ , >	41	_ , PEN	47
4	pe1 , pen	66	sin , scn	81	scn , SIN	41	scn , SIN	47
5	pe1 , PEN	65	si1 , SIN	80	SIN , scn	39	pe1 , pen	45
6	< , pe1	61	pe1 , pen	80	pe1 , pe1	37	SIN , scn	42
7	_ , < , pe1	57	pi1 , pin	63	pe1 , pen	37	pe1 , pe1	42
8	si1 , sin , scn	53	pe1 , pe1	62	< , PEN	34	scn , scn	39
9	si1 , SIN	52	pin , pcn	61	scn , scn	33	PEN , pen	36
10	pi1 , pin	46	scn , si1	60	< , pe1	32	pen , pen	36

Table 5

Gini coefficient of top 50 micro-patterns for experiments *every attempt* and *every problem*.

Gini Coefficient		
<i>every attempt</i>	<i>by login session</i>	0.229
	<i>by learning topic</i>	0.257
<i>every problem</i>	<i>by login session</i>	0.197
	<i>by learning topic</i>	0.180

Table 6

Statistical tests comparing students with themselves and others.

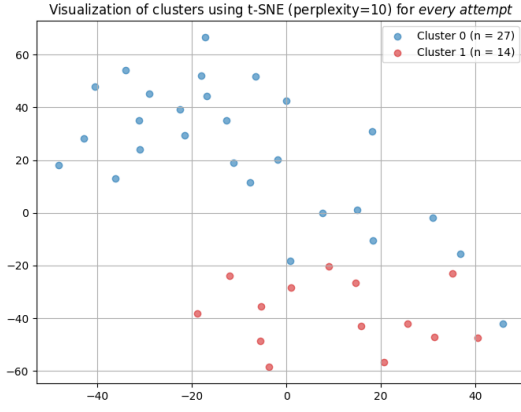
		self-distance		other-distance		<i>t</i>	<i>p</i>	<i>Cohen's d</i>
		M	STD	M	STD			
<i>every attempt</i>	<i>by login session</i>	0.550	0.144	0.637	0.048	-2.670	<0.02	-0.810
	<i>by learning topic</i>	0.550	0.108	0.650	0.042	-3.824	<0.002	-1.208
<i>every problem</i>	<i>by login session</i>	0.558	0.155	0.672	0.043	-2.630	<0.03	-1.002
	<i>by learning topic</i>	0.534	0.121	0.666	0.033	-3.795	<0.003	-1.451

sessions. These results suggest that our student behavior profiles, constructed as frequency vectors of micro-patterns, are stable and valid representations of student behavior.

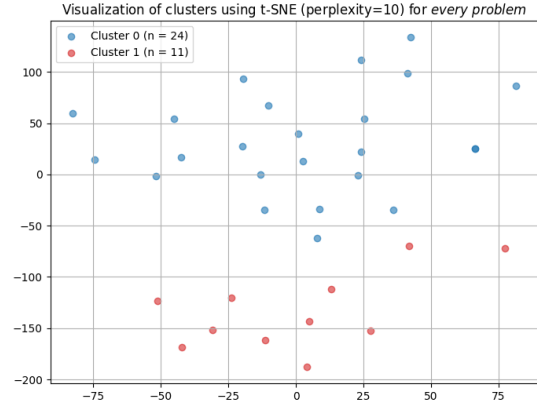
5.2.3. Clustering

We applied *k*-means clustering (using the Elbow Method to determine optimal *k*) to the student vectors to identify groups with similar behavior patterns. Figures 1a and 1b show the results of clustering using t-SNE (t-distributed Stochastic Neighbor Embedding), a dimensionality reduction technique to visualize high-dimensional data. As a hyperparameter, perplexity makes a guess about the number of nearest neighbors each point considers when mapping the high-dimensional space to 2D. We considered perplexity to be 10, meaning more emphasis on small groups of students with very similar topic behavior. According to Figure 1a, cluster 0 has 27 students, while cluster 1 has 14. In Figure 1b, cluster 0 has 24 students, while cluster 1 has 11. Both figures confirm that clustering student topic-based vectors produces meaningful separation, even when done at a detailed attempt level, as these clusters have distinct boundaries with minimal crossing points.

Next, to analyze the differences between these clusters, we compared *cluster profiles* constructed



(a) *every attempt*



(b) *every problem*

Figure 1: t-SNE visualization (perplexity=10) of clustered student topic vectors. Each point represents a student.

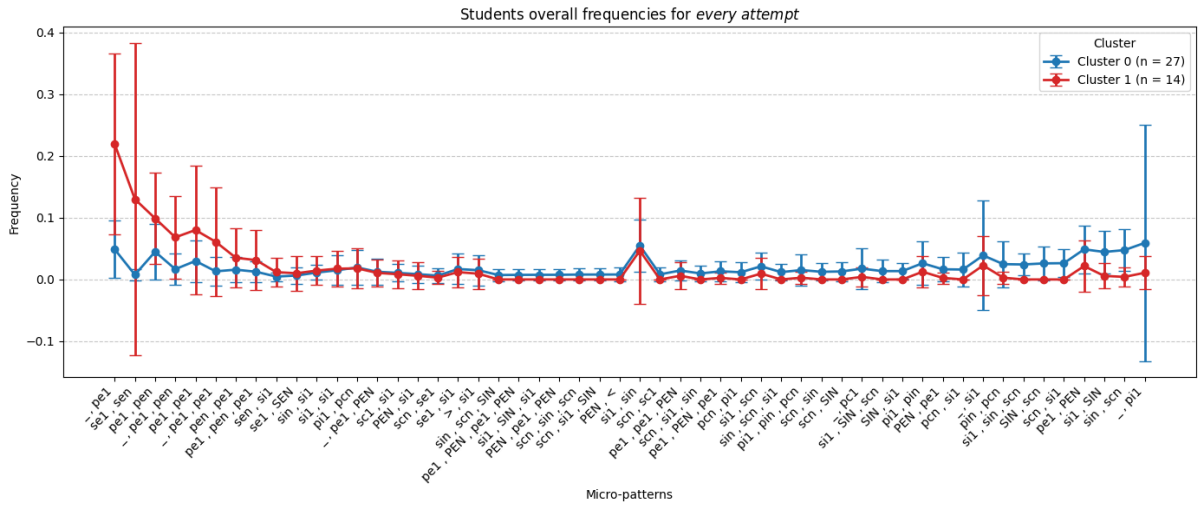


Figure 2: Top 50 micro-patterns and their frequencies in each cluster for experiment *every attempt*. Patterns are ordered by the difference in frequencies between Cluster 1 (*example explorers*) and Cluster 0 (*persistent finishers*).

by averaging frequencies of the top 50 micro-patterns for each cluster. To highlight the discovered differences, we displayed micro-pattern frequencies for both clusters in the same graph, ordering the patterns by the difference in frequency between the clusters (see Figure 2 for *every attempt* and Figure 3 for *every problem*). This revealed that students in these clusters differed in their use of two distinct groups of micro-patterns at opposite ends of the spectrum.

On the left end, we observe micro-patterns related to the focused exploration of comprehension-focused worked examples. For *every attempt*, seven of the 10 leftmost micro-patterns include comprehension example tokens (containing ‘p’ and ‘e’), with six including at least two. Similarly, eight of the 10 leftmost micro-patterns for *every problem* include comprehension example tokens, with five including at least two. These micro-patterns are more frequent in Cluster 1, especially in *every problem* experiment, which attempted to magnify the transition between different activities. The analysis shows that students in Cluster 1 were considerably more engaged in example-based learning than those in Cluster 0. To stress this behavior, we called students belonging to Cluster 1 *example explorers*.

On the right end, we observe micro-patterns involving repeated attempts, mostly at construction problems. For *every attempt*, seven of the 10 rightmost micro-patterns include construction tokens (containing ‘s’), with five including at least two. Similarly, eight of the 10 rightmost micro-patterns for *every problem* include construction tokens, with six including at least two. Furthermore, in both experiments, about half of the rightmost frequent sequences (i.e., sequences used much more frequently

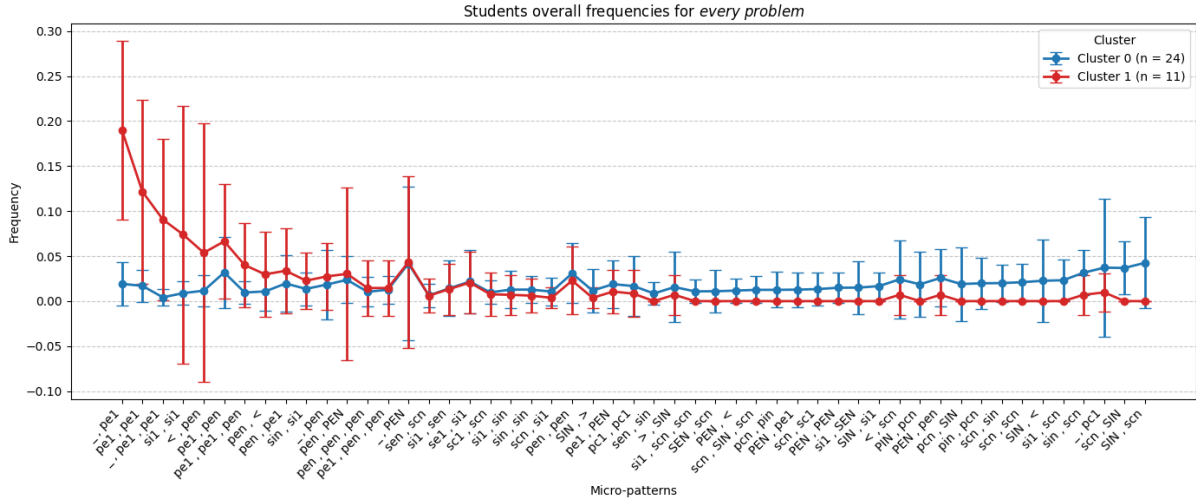


Figure 3: Top 50 micro-patterns and their frequencies in each cluster for experiment *every problem*. Patterns are ordered by the difference in frequencies between Cluster 1 (*example explorers*) and Cluster 0 (*persistent finishers*).

Table 7

Characteristic examples of an *example explorer* and a *persistent finisher* during a practice session.

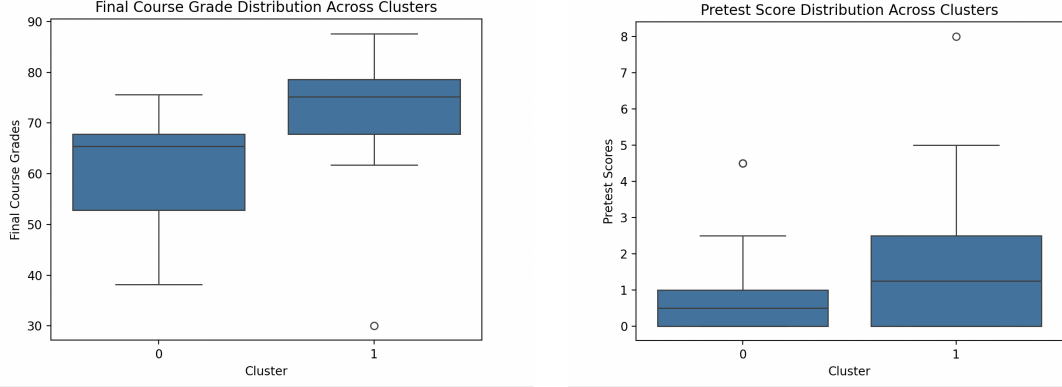
	example sequence	explanation
<i>example explorer</i>	_, pe1, pen, pe1, pe1, PEN, _, <, pe1, pe1, PEN, >, se1, sen, >, sen	This student attempts several examples from different topics.
<i>persistent finisher</i>	_, pi1, PIN, pcn, _, >, si1, scn, pc1, si1, SIN, scn, >, si1, SIN, scn	This student attempts several problems from different topics, reaching a correct solution before moving to the next.

by students in Cluster 0) ended with a correct attempt to solve a problem (tokens containing ‘c’). The dominance of these micro-patterns suggests that another important difference between clusters is a much larger focus of students in Cluster 0 to persistently work on continuous problem solving, aiming to achieve correctness. To stress this behavior, we called students belonging to Cluster 0 *persistent finishers*.

The results of two distinct groups of students, *example explorers* and *persistent finishers*, highlight key differences in how students engage with programming practice. Moreover, a similar split into *example explorers* and *persistent finishers* observed in two experiments with different sequence coding approaches suggests that this split might represent important differences in student practice behavior. Table 7 shows characteristic examples of practice behaviors from each group, illustrating how an *example explorer* and a *persistent finisher* approach practice differently.

5.2.4. Mann-Whitney U Test

A Mann-Whitney U test showed a significant relationship between behavior (cluster assignment) and performance (final course grades) ($U=61$, $P<0.01$), as final grades can be considered a reliable proxy for meaningful learning [32]. According to these results, *example explorers* had significantly higher final course grades compared to persistent finishers. Figure 4a illustrates the final course grade distribution across clusters. A second Mann-Whitney U test between background knowledge (pre-test scores) and performance revealed a non-significant relationship between behavior and background knowledge ($U=103$, $p>0.1$). However, although pre-test scores are low across both clusters, we observe a trend toward higher scores among *example explorers*. We hypothesize that a floor effect may be present, where the pre-test may not have been sensitive enough to capture meaningful differences in background knowledge. Figure 4b illustrates the pre-test score distribution.



(a) Distribution of student final course grades

(b) Distribution of student pre-test scores

Figure 4: Box plots compare student performance across clusters with respect to final grades and pre-test scores.

We further divided students in each cluster into two groups, *extreme* and *moderate*, based on their proximity to the opposite cluster’s centroid. Mann-Whitney U tests revealed a significant difference between students’ performance in each cluster ($U=22$, $p<0.01$), with *example explorers* having significantly higher final course grades compared to *persistent finishers*. On the other hand, no significant difference was found between moderate and extreme groups for *example explorers* ($U=22$, $p>0.1$) and *persistent finishers* ($U=27.5$, $p>0.1$). This suggests certain problem-solving strategies can be more indicative of learning compared to others. This is especially important since no significant relationship was found between performance and background knowledge.

6. Implications of Outcomes

While our quantitative results show a clear distinction between *example explorers* and *persistent finishers* in terms of their final course performance, as computing education researchers, we find it imperative to ground these findings within real classroom learning dynamics. For instance, students who are considered *persistent finishers* often demonstrate a consistent pattern of attempting problems repeatedly until they succeed. However, their persistence may not always translate into deeper understanding via further internalization of concepts [33, 16]. Even while keeping their focus on solving problems, they might essentially engage in surface trial-and-error learning without gaining a deeper understanding of the underlying concepts. This pattern suggests that prioritizing repeated problem-solving attempts over learning from worked examples might not lead to a better conceptual understanding, resulting in lower course grades. In contrast, students exhibiting behavior characteristic of *example explorers* might learn the proper way of solving the main type of problems presented by worked examples [34, 35, 36], and reinforce their understanding and performance [37], as is evident from their higher course grades.

Recognizing these behavior profiles will allow instructors to scaffold learning more effectively [38], leading to direct implications for personalized learning systems. These insights can be leveraged to inform adaptive pedagogy that responds to student behavior pattern types: prompting *persistent finishers* to reflect on examples and rewarding *example explorers* upon challenge completion. Such adaptive actions based on student behavior can support their learning, leading to improved course outcomes [39]. Based on how students engage with learning materials, instructors can also recommend specific strategies to each set of learners to achieve greater conceptual understanding and higher learning gains.

7. Conclusions, Limitations, and Future Work

In this paper, we explored student practice behavior in a Python practice system. We mined frequent micro-patterns from student practice sequences and built micro-pattern vectors consistently reflecting

their learning behavior profile. Through clustering, we revealed two distinct behavior patterns: *example explorers* and *persistent finishers*. A Mann-Whitney U test demonstrated a significant relationship between behavior patterns and final grade scores, with *example explorers* having significantly higher performance.

Although our results offer insights for personalized learning systems, the relatively small sample size and specificity of the data limit generalizability and may overestimate the broader applicability. Additionally, our study does not account for external factors such as teaching context, student engagement, and educational support, all of which could influence the results. The dataset included only students who voluntarily sought additional practice, introducing potential self-selection bias as participants may be more self-motivated or in greater need of support than the average student. Lastly, our comparison of pre-test scores to final course grades may be affected by test-taking ability, which can vary independently of course understanding, and the type of assignments administered during the course. We chose to use final course grades over post-test scores due to the very limited number of students who completed the voluntary post-test. Our dataset also lacks information about assignments and exams (e.g., whether students were tested more on example problems versus construction problems), which could bias the comparison between pre-test and final course grades, as well as the types of problems students chose in the practice system.

In the future, we plan to conduct a more in-depth investigation of learning behaviors and outcomes, including classroom experiments testing different problem orderings and temporal analysis of behavior change and learning gains. We also plan to conduct qualitative analysis on sequences to more deeply understand the strategies behind the different student behaviors.

Acknowledgments

We thank Kamil Akhuseyinoglu for his help with the dataset used in this study. This material is based upon work supported by the National Science Foundation under Grant No.2213789.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] L. Malmi, J. Sheard, P. Kinnunen, Simon, J. Sinclair, Computing education theories: What are they and how are they used?, in: Proceedings of the 2019 ACM Conference on International Computing Education Research, 2019, pp. 187–197.
- [2] K. Akhuseyinoglu, P. Brusilovsky, Data-driven modeling of learners’ individual differences for predicting engagement and success in online learning, in: Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, 2021, pp. 201–212.
- [3] R. Hosseini, P. Brusilovsky, M. Yudelson, A. Hellas, Stereotype modeling for problem-solving performance predictions in MOOCs and traditional courses, in: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP ’17, Association for Computing Machinery, New York, NY, USA, 2017, p. 76–84.
- [4] M. S. Boroujeni, P. Dillenbourg, Discovery and Temporal Analysis of MOOC Study Patterns, Journal of Learning Analytics 6 (2019) 16–33.
- [5] N. Gitinabard, S. Heckman, T. Barnes, C. F. Lynch, What will you do next? a sequence analysis on the student transitions between online platforms in blended courses, in: the 12th International Conference on Educational Data Mining (EDM 2019), 2019, pp. 59–68.
- [6] D. Leeds, C. Chen, Y. Zhao, F. Metla, J. Guest, G. Weiss, Generalized sequential pattern mining of undergraduate courses, in: A. Mitrovic, N. Bosch (Eds.), the 15th International Conference on Educational Data Mining (EDM 2022), 2022, pp. 430–437.

- [7] P. Brusilovsky, L. Malmi, R. Hosseini, J. Guerra, T. Sirkiä, K. Pollari-Malmi, An integrated practice system for learning programming in python: design and evaluation, *Research and Practice in Technology Enhanced Learning* 13 (2018) 18.1–18.40.
- [8] A. M. Gaweda, C. F. Lynch, Student practice sessions modeled as icap activity silos, in: *14th International Conference on Educational Data Mining*, 2021, pp. 595–601.
- [9] P. Brusilovsky, Intelligent technologies for personalized practice systems, *Information and Technology in Education and Learning* 4 (2024). URL: <https://doi.org/10.12937/itel.4.1.Inv.p001>.
- [10] K. Muldner, J. Jennings, V. Chiarelli, A review of worked examples in programming activities, *ACM Transactions on Computing Education* 23 (2022) 1–35.
- [11] X. Wang, D. Yang, M. Wen, K. Koedinger, C. P. Rosé, Investigating how student’s cognitive behavior in MOOC discussion forums affect learning gains, in: *Educational Data Mining Conf.*, 2015, pp. 226–233.
- [12] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, D. T. Seaton, Studying learning in the worldwide classroom: Research into edx’s first MOOC, *Research & Practice in Assessment* 8 (2013) 13–25.
- [13] A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec, Engaging with massive online courses, in: *World Wide Web Conf.*, ACM, 2014, pp. 687–698.
- [14] S. Lorenzen, N. Hjuler, S. Alstrup, Tracking behavioral patterns among students in an online educational system, in: *the 11th International Conference on Educational Data Mining*, 2018, pp. 280–285.
- [15] P. F. Carvalho, M. Gao, B. A. Motz, K. R. Koedinger, Analyzing the relative learning benefits of completing required activities and optional readings in online courses., in: *the 11th International Conference on Educational Data Mining (EDM 2018)*, 2018, pp. 418–423.
- [16] P. Ramsden, *Learning to teach in higher education*, routledge, 2003.
- [17] T. Van Gog, L. Kester, F. Paas, Effects of worked examples, example-problem, and problem-example pairs on novices’ learning, *Contemporary Educational Psychology* 36 (2011) 212–218.
- [18] K. Akhuseyinoglu, A. Klačnja-Milicevic, P. Brusilovsky, The impact of connecting worked examples and completion problems for introductory programming practice, in: R. Ferreira Mello, N. Rummel, I. Jivet, G. Pishtari, J. A. Ruipérez Valiente (Eds.), *European Conference on Technology Enhanced Learning (EC-TEL 2024)*, *Technology Enhanced Learning for Inclusive and Equitable Quality Education, Part 1*, volume 15159 of *Lecture Notes in Computer Science*, Springer International Publishing, 2024, pp. 3–18. URL: https://doi.org/10.1007/978-3-031-72315-5_1. doi:10.1007/978-3-031-72315-5_1.
- [19] J. Champaign, K. F. Colvin, A. Liu, C. Fredericks, D. Seaton, D. E. Pritchard, Correlating skill and improvement in 2 MOOCs with a student’s time on tasks, in: *ACM Learning at Scale Conference*, ACM, 2014, pp. 11–20.
- [20] J. McBroom, B. Jeffries, I. Koprinska, K. Yacef, Mining behaviours of students in autograding submission system logs., *International Educational Data Mining Society* (2016).
- [21] A. S. Carter, C. D. Hundhausen, Using programming process data to detect differences in students’ patterns of programming, in: *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 2017, pp. 105–110.
- [22] P. Tschisgale, M. Kubsch, P. Wulff, S. Petersen, K. Neumann, Exploring the sequential structure of students’ physics problem-solving approaches using process mining and sequence analysis, *Physical Review Physics Education Research* 21 (2025) 010111.
- [23] J. S. Kinnebrew, K. M. Loretz, G. Biswas, A contextualized, differential sequence mining method to derive students’ learning behavior patterns., *Journal of Educational Data Mining* 5 (2013) 190–219.
- [24] J. Guerra, S. Sahebi, Y.-R. Lin, P. Brusilovsky, The problem solving genome: Analyzing sequential patterns of student work with parameterized exercises, in: J. Stamper, Z. Pardos, M. Mavrikis, B. M. McLaren (Eds.), *the 7th International Conference on Educational Data Mining (EDM 2014)*, 2014, pp. 153–160.
- [25] Y. Mao, S. Marwan, What time is it? student modeling needs to know, in: *In proceedings of the 13th International Conference on Educational Data Mining*, 2020.

- [26] Y. Zhang, L. Paquette, Sequential pattern mining in educational data: The application context, potential, strengths, and limitations, in: *Educational data science: Essentials, approaches, and tendencies: proactive education based on empirical big data evidence*, Springer, 2023, pp. 219–254.
- [27] S.-C. Huang, C.-C. Chiou, J.-T. Chiang, C.-F. Wu, Online sequential pattern mining and association discovery by advanced artificial intelligence and machine learning techniques, *Soft Computing* 24 (2020) 8021–8039.
- [28] M. Kong, L. Pollock, Semi-automatically mining students' common scratch programming behaviors, in: *Proceedings of the 20th Koli Calling International Conference on Computing Education Research*, 2020, pp. 1–7.
- [29] J. Ayres, J. Flannick, J. Gehrke, T. Yiu, Sequential pattern mining using a bitmap representation, in: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 429–435.
- [30] D. Perera, J. Kay, I. Koprinska, K. Yacef, O. R. Zaïane, Clustering and sequential pattern mining of online collaborative learning data, *IEEE Transactions on knowledge and Data Engineering* 21 (2008) 759–772.
- [31] J. Stamper, S. Moore, C. Rose, P. Pavlik, K. Koedinger, Learnsphere: A learning data and analytics cyberinfrastructure, *Journal of Educational Data Mining* 16 (2024) 141–163. URL: <https://jedm.educationaldatamining.org/index.php/JEDM/article/download/772/201>.
- [32] S. Li, J. Du, J. Sun, Unfolding the learning behaviour patterns of mooc learners with different levels of achievement, *International Journal of Educational Technology in Higher Education* 19 (2022) 22.
- [33] A. Desierto, C. De Maio, J. O'Rourke, S. Sharp, Deep or surface? the learning approaches of enabling students in an australian public university, in: *STARS Conference*, 2018.
- [34] C.-Y. Chen, Effects of worked examples with explanation types and learner motivation on cognitive load and programming problem-solving performance, *ACM Transactions on Computing Education* (2025).
- [35] K. J. Crippen, B. L. Earl, The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy, *Computers & Education* 49 (2007) 809–821.
- [36] S. Verstege, Y. Zhang, P. Wierenga, L. Paquette, J. Diederren, Using sequential pattern mining to understand how students use guidance while doing scientific calculations, *Technology, Knowledge and Learning* 29 (2024) 897–920.
- [37] L. E. Margulieux, R. Catrambone, M. Guzdial, Employing subgoals in computer programming education, *Computer Science Education* 26 (2016) 44–67.
- [38] R. M. Bernard, E. Borokhovski, R. F. Schmid, D. I. Waddington, D. I. Pickup, Twenty-first century adaptive teaching and individualized learning operationalized as specific blends of student-centered instructional events: A systematic review and meta-analysis, *Campbell Systematic Reviews* 15 (2019) e1017.
- [39] R. Abedi, M. R. N. Ahmadabadi, F. Taghiyareh, K. Aliabadi, S. Pourroustaei, The effects of personalized learning on achieving meaningful learning outcomes, *Interdisciplinary Journal of Virtual Learning in Medical Sciences* 12 (2021) 177–187.