DiSCS: A New Sequence Segmentation Method for Open-Ended Learning Environments

James P. Bywater1(✉)[0000-0001-6053-6382], Mark Floryan2[0000-0002-0171-5900], and Jennifer L. Chiu2[0000-0001-7663-5748]

1James Madison University, Harrisonburg, VA, USA.
2University of Virginia, Charlottesville, VA, USA.
bywatejx@jmu.edu, mrf8t@virginia.edu, jlc4dz@virginia.edu

Abstract. Open-ended learning environments afford students opportunities to explore, manipulate, and test concepts, and have the potential to provide students with feedback and support by leveraging the log data generated by them. However, within open-ended contexts, student log data is often noisy and identifying periods of meaningful activity is difficult. This paper introduces a new sequence mining method to overcome this challenge. The Differential Segmentation of Categorical Sequences (DiSCS) algorithm finds segments within a sequence of actions that are maximally or near-maximally different from their immediate neighbors. Segments are then clustered to reveal common periods of student activity. We examine the performance of this method under a variety of conditions to find how well DiSCS can identify where different states of simulated activity start and end. We report that when provided with only the observed actions, DiSCS is able to identify the hidden states of simulated student activity with strong and very strong associations. This strong performance is robust across a variety of contexts including those where observed actions are noisy or common to multiple states. We discuss the implications and limitations of this method for open-ended learning environments.

Keywords: Sequence mining, Segmentation, Open-ended learning environments.

1 Introduction

Open-ended learning environments use the affordances of technology to create opportunities for students to explore, manipulate, and test concepts and knowledge [1]. Open-ended learning environments are designed to engage students with ill-structured tasks and typically involve computer-based games, simulations, visualizations, design, or experimentation tools [2, 3, 4]. However, implementing open-ended learning environments in classrooms presents challenges for teachers [2, 5, 6]. For example, to provide effective guidance to students using design-based learning environments, teachers need to understand and notice the different design strategies that each student takes [7]. Given that each student will likely have a unique solution instead of one "right" answer, and their paths to that solution are likely to differ, noticing these design strategies for each student can be complex and challenging [8, 9].

Various ways to document or capture student's learning behaviors within open-ended learning environments include think-alouds [10] or reflections [11]. These methods are typically labor intensive, usually implemented in undergraduate or professional settings [12], and have limited applicability to precollege classroom settings. Given the computer-based nature of open-ended learning environments, students can be provided with automated guidance by leveraging log data of students' actions within the environment. As a result, various research investigates applying data mining techniques to log data to provide insight into students' learning behaviors. Within open-ended learning environments, microlevel student interaction data [13] from a variety of contexts such as coding tasks [14], inquiry tasks [15, 16], and iterative design tasks [17, 18] have been used to identify and support a broad range of constructs including metacognitive states [19, 20], inquiry or design strategies [15, 16, 18] and problem-solving [17].

However, while there are numerous sequential data mining techniques used to analyze student interaction data [21], less attention has been given to applying these methods to open-ended design-based contexts, and especially with real, noisy, open-ended classroom data. Many challenges with classroom data remain, particularly when segmenting sequences into meaningful periods of student activity [13]. To improve this important area of data mining this paper introduces and describes a new sequential data mining method for segmenting noisy student activity from open-ended learning environments. The Differential Segmentation of Categorical Sequences (DiSCS) algorithm uses dynamic programming and genetic techniques to find segments within an individual student's sequence of actions that are maximally or near-maximally different from segments that are their immediate neighbors. Clustering techniques then group segments and reveal common periods of activity within an individual student's sequence, or across the sequences of multiple students. This makes it possible to identify different phases of activity and when they start and stop. To examine the performance of this method, this paper addresses the following research questions:

1. To what degree is DiSCS able to identify optimal segmentation when using classroom data?
2. What is the strength and robustness of the association between states identified by DiSCS and simulated hidden states?

2 Sequence Mining

The field of educational data mining is broadly defined as developing and applying computer algorithms to detect patterns in educational data that would be difficult to do otherwise [22]. These algorithms can be used at the national, institutional, or classroom levels, with the latter often referred to as fine-grained or microlevel analysis [13]. Microlevel techniques include a variety of knowledge tracing strategies [23], time-series analyses [24], or the development of 'evidence models' or sensors for the constructs of interest [25]. To different degrees, these techniques combine deductive expert knowledge with inductive knowledge found using a variety of data mining methods.

Within educational settings, microlevel (i.e., student interaction level) sequential log data are commonly analyzed using process or pattern mining. This approach looks for the presence of specific ordered patterns of interactions or keystrokes within a sequence, or for the frequency of the most commonly occurring patterns. However, with noisy educational data, sequential pattern mining algorithms such as the Sequential Pattern Mining (SPAM) [26] algorithm and the Generalized Sequential Pattern (GSP) [27] mining algorithm often find large numbers of frequent patterns that make it difficult to identify which patterns are important or meaningful [19]. Furthermore, students are known to use different microlevel pathways to achieve similar goals or performance. While some studies have identified sequential patterns that characterize high and low achieving students [28], or found patterns that are different between periods of increasing and decreasing performance [19], any set of specific patterns identified this way are likely to be incomplete and problematic to use for instructional feedback [29]. Therefore, when examining sequences for phases of inquiry or design behavior, techniques that place less emphasis on the temporal ordering of microlevel actions and which take a more holistic approach may be more appropriate.

An alternative approach involves segmenting sequences and examining each segment holistically. These approaches can overcome some of the problems highlighted above, however, rules for when to cut a sequence into segments can also create their own problems. In contexts where students complete a series of small finite tasks or levels, segmentation has occurred at the start and end of each of these tasks [30]. However, this approach has limited applicability to open-ended learning contexts given that these settings are inherently less prescriptive. Within open-ended learning settings, segmentation has been performed by examining concurrent performance data and creating segments corresponding to periods of increasing and decreasing performance scores [19]. However, during phases of student activity, performance scores may fluctuate as students investigate the positive and negative impact of different factors, thus segmenting on a performance basis has limited applicability. An alternative approach has been to set a temporal resolution and use the timestamps of the dataset to segment after a fixed time period (e.g., every minute [18] or every 20-seconds [31]). However, to avoid slicing distinct periods of activity into unrepresentative segments this approach requires that the chosen duration is set much shorter than the expected length of periods of activity. Other heuristics for segmenting sequences, such as using periods of inactivity, risk conflating meaningful transitions in learning behavior with less meaningful transitions such as bathroom breaks, the end of class, or an unreliable internet connection.

For data from open-ended learning settings, there is still a need for a method to segment a sequence of student actions that is based on differences between phases of those actions. That is, there is not yet a method that determines how best to segment a sequence by comparing, holistically, the segments themselves. This paper introduces such a method.

3 Differential Segmentation of Categorical Sequences (DiSCS)

The DiSCS algorithm takes a *categorical sequence* and splits it into *segments* so that each of the segments is as *different* from its immediate neighbors as possible. In effect, it finds the most distinct segments within a sequence of categories (see Fig. 1 for a simple example).

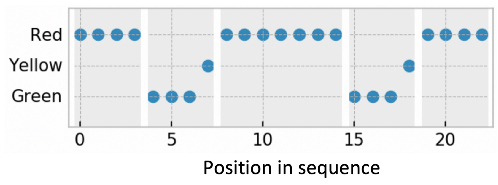

Fig. 1. In this simple example, DiSCS splits a sequence into five distinct segments separated by vertical white lines.

3.1 Optimization Function

More specifically, DiSCS splits an individual student's sequence of L time-ordered, categorical actions into m non-overlapping, contiguous segments such that the average of all the differences between neighboring segments is maximized. The difference between a given pair of neighboring segments is measured by finding the difference between the proportions of the actions in the neighboring segments. The paired z-scores for each of the actions is then found and averaged. That is, if $A = \{a_1, a_2, \ldots., a_k\}$ is the set of possible categorical actions and $S = \{s_1, s_2, \ldots, s_L\}, s_i \in A$ is the time-ordered sequence of these actions, the algorithm splits S into m non-overlapping, contiguous segments $\{C_1, C_2, \ldots., C_m\}$ where $C_1 = \{s_1, s_2, \ldots, s_{l_1}\}$, $C_2 = \{s_{l_1+1}, s_{l_1+2}, \ldots, s_{l_2}\}$, etc. Given that both the number of segments, m, and the positions at which the segments end, l_1, l_2, …, l_{m-1}, can vary, DiSCS adjusts these parameters to maximize the average of the differences between all the pairs of neighboring segments, i.e., it averages the difference between C_j and C_{j+1} for j from 1 to m-1.

The difference between a given pair of segments C_j and C_{j+1} , Z_j, is measured by first finding the proportion of each action for the segments, P_j and P_{j+1}, where $P_j = \{p_{1j}, p_{2j}, \ldots., p_{kj}\}$ and p_{ij} is the count of action a_i in segment C_j divided by n_j, the total number of actions in segment C_j. Then, the paired-sample z-test statistic, z_{ij}, is calculated for each of the k pairs of corresponding proportions, i.e. p_{ij} and $p_{i(j+1)}$,

$$z_{ij} = \frac{p_{ij} - p_{i(j+1)}}{\sqrt{p_{pooled}(1 - p_{pooled})\left(\frac{1}{n_j} + \frac{1}{n_{j+1}}\right)}}. \tag{1}$$

The difference between a given pair of segments C_j and C_{j+1}, Z_j, is calculated by averaging the absolute values of z_{ij}, i.e.,

$$Z_j = \frac{\sum_{i=1}^{k} |z_{ij}|}{k}, \tag{2}$$

and an overall score for the differences between all the m-1 pairs of neighboring segments in the sequence, Z, is calculated by averaging each of the values of Z_j, i.e.,

$$Z = \frac{\sum_{j=1}^{m-1} Z_j}{m-1}. \tag{3}$$

Given that the optimal number of segments, m, is not known, we evaluate Z for values of m from 2 (i.e., the smallest possible number of segments) to a value much higher than the number of segments expected. While the maximum possible value of m is equal to the length of the sequence, L, (i.e., when segments consist of only one action per segment), given the computational burden of testing all these values, it is possible to use a heuristic for determining this value. For example, as we explain later, in this paper we test *m* up to three standard deviations above the expected number of segments, but other heuristic such as *L*/3 or *L*/4 would likely be sufficient.

Therefore, by varying the parameter *m* between 2 and *M* (either *L* or an alternative heuristic) and the end positions, l_1, l_2, …, l_{m-1}, of the segments to all their possible combinations, DiSCS maximizes *Z*, i.e.,

$$\underset{2 \le m \le M}{\operatorname{argmax}} \left(\underset{l_1, l_2, \ldots, l_{m-1}}{\operatorname{argmax}} \left(\frac{\sum_{j=1}^{m-1} z_j}{m-1} \right) \right) \tag{4}$$

In this form, our optimization function tends to favor small numbers of long segments. This is because the initial segmentation tends to capture the largest differences within a sequence and averaging in subsequent segmentations typically lowers the average. In order to adjust for this tendency, we introduce a smoothing variable, *t*, which has the effect of favoring larger numbers of segments, or smaller grain-size segments of student actions. The final optimization function can therefore be written as:

$$\underset{2 \le m \le M}{\operatorname{argmax}} \left(\underset{l_1, l_2, \ldots, l_{m-1}}{\operatorname{argmax}} \left(\frac{\sum_{j=1}^{m-1} z_j}{m-1+t} \right) \right). \tag{5}$$

3.2 Algorithm

While it is possible to find the optimal solution described above using a brute-force approach, the number of calculations required to test all the possible values of the parameters *m*, l_1, l_2, …, l_{m-1}, with *m* up to a maximum *M*, on a sequence of length *L*, is proportional to L^M. Therefore, DiSCS uses two different optimization techniques (a genetic and a dynamic programing algorithm) to find solutions more quickly, and takes the best result from each approach. The code for each of these algorithms is available at the DiSCS code repository [32]. While this approach has the advantage of differentially segmenting categorical sequences quickly, it introduces the possibility of finding only local-maxima and may report non-optimal parameter values which could be particularly problematic when analyzing noisy classroom data. This potential problem is investigated in research question 1.

3.3 Clustering

After segmenting a sequence, DiSCS clusters the segments. This step is intended to help identify similarities in the design behaviors across all students and examine the proportion of actions that are typical of that cluster. To do this, we used *k*-means clustering with the distance measure set to the Euclidean distance between the proportions of actions within each segment. We repeated the *k*-means clustering 100 times for every possible value of the total number of clusters, each time calculating the average silhouette width [33] to measure of the quality of the clustering. The clustering with the largest average silhouette width was selected. Segments that were in the same cluster were then given the same arbitrary label. For example, the final DiSCS output for the simple example given in Fig. 1 would create two clusters, one for segments where the observations are 'red' and another for segments with 'green' and 'yellow'.

4 Optimality of DiSCS

The first research question that this paper addresses is the degree to which DiSCS is able to identify optimal segmentation when using noisy classroom data. To do this we compared DiSCS segmentations with brute-force optimal segmentations under conditions that allow for a timely brute-force result (i.e., when the maximum number of segments tested, M, is 5).

4.1 Method

Context. The classroom data used for this investigation was obtained from 75 environmental science high school students who worked in 38 small groups (typically pairs) in their regular classroom setting to complete a design activity over multiple days. The design activity consisted of a series of scaffolded design challenges related to building a house that consumed no net energy over a year while still meeting cost, size, and aesthetic constraints. Students used an open-ended CAD environment called Energy3D [34] that enabled them to build and test different building designs that incorporate solar panels. Embedded tools could be used to examine energy gains and losses under various conditions and help students understand concepts such as energy transfer. The high school was located in the Eastern United States, two of the classes were 'honors' classes and three 'regular' level classes, and the school demographics were 34% Black, 9% Hispanic, and 45% White students with 45% of students receiving free or reduced lunch.

Data Collection. Student action data was collected while students were engaged with the design activity. Examples of the types of actions recorded were "edit roof", "add a solar panel", "change the tilt of the solar panel", and "do annual energy analysis". For each of the 38 small groups, one sequence of action data was collected containing all the actions performed by that group throughout the duration of the design activity. The sequences were long (mean = 379; standard deviation = 245) and included up to 42 different actions.

Analysis. To assess the degree to which the genetic and the dynamic programing algorithms used by DiSCS were able to find the optimal segmentation, we compared the maximum optimization function parameter values (see equation 5) for these algorithms with the guaranteed maximum value found by the brute-force algorithm. This analysis was conducted with the sequence of actions recorded by each group of students.

4.2 Results

When considered separately, the genetic algorithm reported maxima that were on average 0.2% smaller than brute-force, and the dynamic programming algorithm reported maxima that were on average 0.4% smaller. This indicates that both algorithms provide near optimal solutions. However, this result is improved when following the approach used by DiSCS and selecting the best performing result from these two algorithms. In this case, the maxima were on average only 0.03% smaller than the actual maxima found using the brute-force algorithm. This indicates that while sub-optimal, using an approach that uses both a genetic algorithm and a dynamic programming algorithm, and takes the best of the two solutions generated provides very near optimal solutions.

5 Strength and Robustness of DiSCS

The second research question that this paper addresses relates to the strength and robustness of the association between states identified by DiSCS and simulated hidden states. To do this we examined the strength of the association and considered how well this strength is maintained under different input conditions.

5.1 Method

To explore the strength and robustness of DiSCS, we conducted a simulation study where stochastically created sequences were generated, segmented with DiSCS, and the association between the segment clusters and the hidden states of the original sequence calculated.

Sequence generation. We used a hidden Markov model to stochastically generate the sequences used in the simulation study. The hidden states represent simulated phases of inquiry or design activity that students may be engaged in and the probability that the student transitions to a different phase is represented by *p*. Different hidden states were modeled to have multiple equally likely observable actions including an action that was common to all hidden states (see Fig. 2).

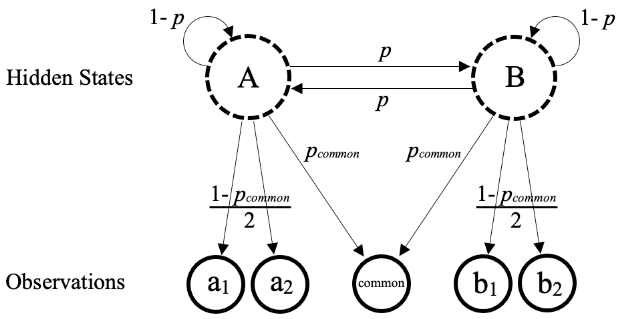

Fig. 2. A hidden Markov model used to generate simulated sequences of student actions. This model shows the case when two hidden states (A and B) each have two observable actions (a_1 and a_2, b_1 and b_2) and a common observable action.

Analysis. For each sequence of observable actions generated, we also recorded the corresponding sequence of hidden states. For example, for the sequence of observations $\{a_1, a_1, a_2, b_2, b_2, b_1, a_1, a_2\}$ we would also record the hidden states {A, A, A, B, B, B, A, A}. The sequence of observations was segmented and clustered with DiSCS and the association between the clusters and the hidden states calculated. See Fig 3. for a summary of the workflow.

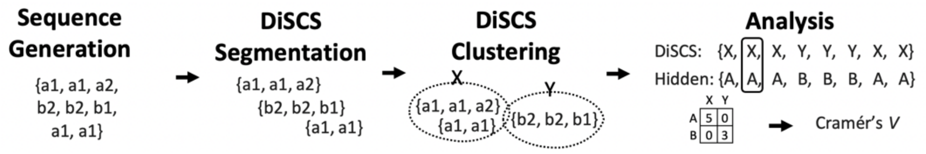

Fig. 3. Workflow for the simulation study.

Given that both the labels of the hidden states and the labels of the DiSCS clusters were nominal categorical values, we used the Cramér's V statistic [35] to measure the association between DiSCS output and the hidden states. As with other measures of association or correlation, $0 \leq V \leq 1$, with one corresponding to a perfect association, and zero for no association between the labels. Values for Cramér's V between 0.6 and 0.8 are considered to be strong associations and values between 0.8 and 1.0 are considered to be very strong [36].

Trials. To understand the robustness of the association, we calculated Cramér's V under different input conditions. First, we examined the impact of sequence length, by testing each of the values, $L = \{25, 50, 75, 100, 150, 200, 300, 400\}$. We did this keeping the number of hidden states = 3, the number of action observations per state = 2, transition probability, $p = 0.1$, the probability of a common action observed, $p_{common} = 0$, and the DiSCS smoothing variable, $t = 3$. For this and other trails in this study, the maximum number of segments that DiSCS optimized over was set to three standard deviations above the expected number of segments i.e., $Lp + 3\sqrt{Lp(1-p)}$. For each value we repeated the test 20 times and recorded the value of Cramér's V each time. Second, we repeated the first trial but changed the number of hidden states from 3 to 2, then to 4. Third, we repeated the first trial again, but changed the number of action observations per state from 2 to 4, then 6. The results from these trials are shown in Fig. 4.

The next set of trials examined the impact of adjusting the sequence generation probabilities p and p_{common}. We examined the impact of different transition probabilities by testing each of the values, $p = \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$ while keeping the sequence length, $L = 200$, the number of hidden states = 3, the number of action observations per state = 2, the probability of a common action observed, $p_{common} = 0$, and the DiSCS smoothing variable, $t = 3$. Again, for each value we repeated the test 20 times and recorded the value of Cramér's V each time. Then we repeated this trial, but instead testing each of the values $p_{common} = \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$ while keeping the transition probability, $p = 0.1$. The results from these trials are shown in Fig. 5.

Lastly, to examine the impact of the DiSCS smoothing variable, we tested each of the values, $t = \{3, 6, 9, 12, 15, 18, 21, 24, 27, 30\}$ while keeping the sequence length, $L = 200$, the number of hidden states = 3, the number of action observations per state = 2, the transition probability, $p = 0.1$, and the probability of a common action observed, $p_{common} = 0$. Again, for each value we repeated the test 20 times and recorded the value of Cramér's V each time. We repeated this trial with the transition probability, $p = 0.05$. The results from these trials are shown in Fig. 6.

5.2 Results

When varying the sequence length, L, strong or very strong associations occurred for most values tested. However, this strength decreased as the sequences became longer. The lines for different numbers of hidden states are close together indicating that changing the number of hidden states has little impact on V for all sequence lengths. This is not the case for the number of actions per state which has a dramatic effect on V, especially for small L (see Fig. 4).

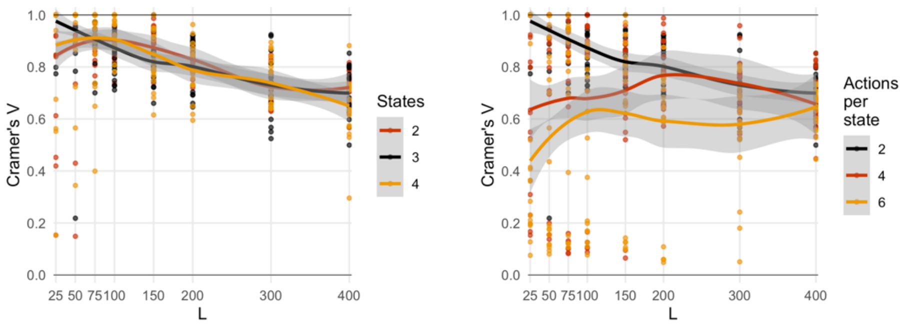

Fig. 4. The impact on association, V, of sequence length, L, using sequences with 2, 3 and 4 hidden states (left) and 2, 4 and 6 actions per state (right). Loess lines with standard errors are shown.

When varying the sequence generation probabilities, p and p_{common}, strong or very strong associations also occurred for most values tested. This strength decreased steadily as these probabilities increased, with larger decreases occurring when the transition probability, p, increases than when p_{common} increases (see Fig. 5).

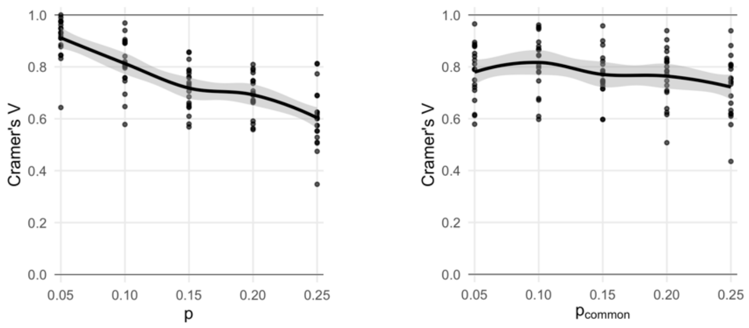

Fig. 5. The impact on association, V, of the sequence generation probabilities, p and p_{common}. Loess lines with standard errors are shown.

When varying the DiSCS smoothing variable, t, very strong associations occurred for all values tested. This strength was most stable for when $p = 0.05$, and improved slightly at larger t values when $p = 0.1$. This indicates that when phases of design activity are expected to change about every 10 actions, adjusting t to larger values improved DiSCS performance, but that when phases of design activity are longer, or change less frequently, adjusting t will have minimal impact on DiSCS performance (see Fig. 6).

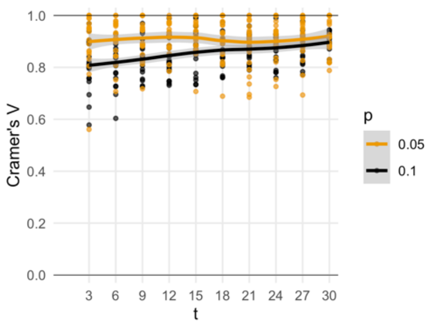

Fig. 6. The impact on association, *V*, of different values of the DiSCS smoothing variable, *t*, at two values of the transition probability, *p*. Loess lines with standard errors are shown.

6 Limitations

Results demonstrate that DiSCS is a strong and robust technique for segmenting sequences of categorical actions such as those commonly found in open-ended learning environments. For example, with sequences with up to 4 hidden states, strong or very strong associations were found. Strong or very strong associates were also found with up to 4 actions per state, up to very high transition probabilities, (at $p = 0.25$ transitions are expected to occur every 4 actions), and when the probability of common actions is also high. However, our simulation testing indicated important limitations. For example, as might be expected, it is clear that performance decreases with larger sequences, and when larger numbers of states and actions make the task more difficult. When sequences are larger than 400, we recommend that DiSCS results are used with caution unless transitions are expected to occur less often that once every 20 actions (equivalent to $p < 0.05$).

7 Implications

DiSCS offers a novel method for educational data mining that can provide insight into student activity within open-ended learning environments. The algorithm provides an efficient and robust method of segmenting categorical sequences of student activity from recorded log data. DiSCS might be used in combination with other measures (e.g., performance, learning gains from pretest to posttest) to determine patterns of actions that correspond to better student performances or learning [28], or that indicate students need help or support. By providing a more efficient method of finding meaningful segments of learning activity, DiSCS works towards being able to provide targeted feedback to learners in open-ended learning environments [17]. Python functions that perform DiSCS segmentation and clustering with either one or multiple lists of categorical actions are available at the DiSCS code repository [32].

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant No. (NSF DRL-1503170). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Land, S.: Cognitive requirements for learning with open-ended learning environments. Educational Technology Research and Development 48(3), 61–78 (2000).
2. Hannafin, M., Hill, J., Land, S., Lee, E.: Student-centered, open learning environments: Research, theory, and practice. In: Handbook of research on educational communications and technology, pp. 641–651. Springer, New York, NY (2014).
3. de Jong, T., Linn, M., Zacharia, Z.: Physical and virtual laboratories in science and engineering education. Science 340(6130), 305–308 (2013).
4. Young, J.: Technology-enhanced mathematics instruction: A second-order meta-analysis of 30 years of research. Educational Research Review 22, 19–33 (2017).
5. Kolodner, J., Camp, P., Crismond, D., Fasse, B., Gray, J., Holbrook, J., ... Ryan, M.: Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design into practice. Journal of the Learning Sciences 12(4), 495–547 (2003).
6. Moore, T., Stohlmann, M., Wang, H., Tank, K., Glancy, A., Roehrig, G.: Implementation and integration of engineering in K-12 STEM education. In: Engineering in Pre-College Settings: Synthesizing Research, Policy, and Practices, pp. 35–60. Purdue University Press (2014).
7. Crismond, D., Adams, R.: The informed design teaching and learning matrix. Journal of Engineering Education 101(4), 738–797 (2012).
8. Purzer, S., Moore, T., Baker, D., Berland, L.: Supporting the implementation of the Next Generation Science Standards (NGSS) through research: Engineering (2014).
9. Wang, H., Moore, T., Roehrig, G., Park, M.: STEM integration: Teacher perceptions and practice. Journal of Pre-College Engineering Education Research 1(2), 2 (2011).
10. Gero J., Tang H.-H.: The differences between retrospective and concurrent protocols in revealing the process-oriented aspects of the design process. Design Studies 22(3), 283–295 (2001).
11. Chen, H., Cannon, D., Gabrio, J., Leifer, L., Toye, G., Bailey, T.: Using wikis and weblogs to support reflective learning in an introductory engineering design course. Human behaviour in design 5, 95–105 (2005).
12. Cross, N.: Design cognition: results from protocol and other empirical studies of design activity. In: Eastman, C., Newstatter, W., McCracken, M. (eds.) Design Knowing and Learning: Cognition in Design Education, pp. 79–103. Elsevier, Oxford (2001).
13. Fischer, C., Pardos, Z., Baker, R., Williams, J., Smyth, P., Yu, R., Slater, S., Baker, R., Warschauer, M.: Mining Big Data in Education: Affordances and Challenges. Review of Research in Education 44(1), 130–160 (2020).
14. Blikstein, P.: Using learning analytics to assess students' behavior in open-ended programming tasks. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 110–116. (2011).
15. Gobert, J. D., Sao Pedro, M., Baker, R., Toto, E., Montalvo, O.: Leveraging educational data mining for real-time performance assessment of scientific inquiry skills within microworlds. Journal of Educational Data Mining 4(1), 104–143 (2012).
16. Käser, T., Schwartz, D.: Modeling and Analyzing Inquiry Strategies in Open-Ended Learning Environments. International Journal of Artificial Intelligence in Education 30(3), 504–535 (2020).
17. Xing, W., Li, C., Chen, G., Huang, X., Chao, J., Massicotte, J., Xie, C.: Automatic Assessment of Students’ Engineering Design Performance Using a Bayesian Network Model. Journal of Educational Computing Research, 1–27 (2020).
18. Vieira, C., Goldstein, M., Purzer, Ş., Magana, A.: Using learning analytics to characterize student experimentation strategies in the context of engineering design. Journal of

Learning Analytics 3(3), 291–317 (2016).
19. Kinnebrew, J., Loretz, K., Biswas, G.: A contextualized, differential sequence mining method to derive students' learning behavior patterns. Journal of Educational Data Mining 5(1), 190–219 (2013).
20. Taub, M., Azevedo, R.: Using Sequence Mining to Analyze Metacognitive Monitoring and Scientific Inquiry Based on Levels of Efficiency and Emotions during Game-Based Learning. Journal of Educational Data Mining 10(3), 1–26 (2018).
21. Bogarín, A., Cerezo, R., Romero, C.: Survey on educational process mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8(1), e1230 (2018).
22. Romero, C., Ventura, S.: Data mining in education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 3(1), 12–27 (2013).
23. Pelánek, R.: Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques. User Modeling and User-Adapted Interaction 27(3), 313–350 (2017).
24. Xie, C., Zhang, Z., Nourian, S., Pallant, A., Hazzard, E.: A time series analysis method for assessing engineering design processes using a CAD tool. International Journal of Engineering Education 30(1), 218–230 (2014).
25. Shute, V.: Stealth assessment in computer-based games to support learning. Computer Games and Instruction 55(2), 503–524 (2011).
26. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 429–435. (2002).
27. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: International Conference on Extending Database Technology, pp. 1–17. Springer, Berlin, Heidelberg (1996).
28. Martinez, R., Yacef, K., Kay, J., Al-Qaraghuli, A., Kharrufa, A.: Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In: 4th International Conference on Educational Data Mining, pp. 111–120. (2011).
29. Bannert, M., Reimann, P., Sonnenberg, C.: Process mining techniques for analysing patterns and strategies in students' self-regulated learning. Metacognition and learning 9(2), 161–185 (2014).
30. Bouchet, F., Harley, J., Trevors, G., Azevedo, R.: Clustering and profiling students according to their interactions with an intelligent tutoring system fostering self-regulated learning. Journal of Educational Data Mining 5(1), 104–146 (2013).
31. DeFalco, J., Rowe, J., Paquette, L., Georgoulas-Sherry, V., Brawner, K., ... Lester, J.: Detecting and addressing frustration in a serious game for military training. International Journal of Artificial Intelligence in Education 28(2), 152–193 (2018).
32. DiSCS code repository, https://github.com/jpbywater/DiSCS.
33. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65 (1987).
34. Xie, C., Schimpf, C., Chao, J., Nourian, S., Massicotte, J.: Learning and Teaching Engineering Design through Modeling and Simulation on a CAD Platform. Computer Applications in Engineering Education 26(4), 824–840 (2018).
35. Cramer, H.: Mathematical methods of statistics. Princeton Univ. Press, Princeton, NJ (1946).
36. Rea, L., Parker, R.: Designing and Conducting Survey Research: A Comprehensive Guide, 4th Edition. Jossey-Bass (2014).