Supplementary Material of MOOCad

I. Algorithm

We introduce the anomaly detection module in this section, which consists of stage segmentation, anomalous group identification and frequent pattern extraction (Fig. 1).

A. Stage Analysis of Learning Sequence

Summarizing progression stages in event sequences is an important step towards analyzing learning sequences, which helps in understanding variation within different but similarly evolving learning behaviors. MOOcad implements an unsupervised stage analysis algorithm [1] that segments learning sequences into stages with semantic meaning through three critical steps: (1) Estimation of event representation, which converts each event of a sequence into a vector representation by employing a word embedding model [5]. In our application, a learning event/activity is represented by H-code. For example, if the event means a learner watching lecture video in module 1 => section 2 => lecture 3, then we encode the H-code as "V1-2.3". (2) Alignment of event sequences, in which a Dynamic Time Warping (DTW) algorithm [2] is used to align and group event sequences with variable lengths and event orders. (3) Segmentation of sequence, as shown in Fig. 1(b), where aligned learning sequences are segmented into latent stages using an optimization-based algorithm. In our application, there is a coherence score within each segment. The greedy algorithm is used to maximize the total coherence score of all the segments in each iteration. We set a threshold value th for the optimization such that the algorithm will stop automatically when total score increase that is smaller than the threshold.

B. Identification of Anomalous Groups in Each Stage

Although the progression stages are formulated, the learning behaviors in each stage are of great variation, which requires a method to identify the abnormal instances from them. However, identifying the anomalous groups–between the identification of predominant groups and individual outliers– is more meaningful for instructors and learners because they care more about the group behaviors when analyzing the learning sequence data. Therefore, we implemented a LOF-based rare category detection algorithm [4] to cluster the segmented learning sequences into different groups in each stage. The detailed steps are: (1) in each stage, we transfer the sub-learning learning sequence to strings, and then calculate its similarity with all the other sequences by Levenshtein distance. Thereby, for each sub-sequence s, we can construct a representative vector:

$$a = [l_1, l_2, ..., l_m]$$

where l_i is the Levenshtein distance between sub-sequence s and sub-sequence i. (2) The representative vectors are fed into the rare category detection algorithm. We modified this algorithm by setting a threshold value of confidence, such that the groups with an outlier score larger than the threshold will be identified as anomalous groups. The rest of learners are aggregated into a "normal" group. For example, as shown in Fig. 1(c), the learning sequences in the first stage are clustered into two groups, with the top one as normal group and the bottom one as anomalous one. Specifically, the LOF-based rare category detection algorithm detects rare categories by iteratively enlarging the k-neighborhood of an instance and examine the trend of the corresponding LOF scores. The boundary of a rare category is determined by locating the inflection point corresponding to the minimum value of the LOF score. Instances with high confidence of representing an unknown category are chosen as the center of a potential rare category. The thresholds of confidence is set as the as the 80% of the largest confidence value of the most rare category.

C. Extraction of Frequent Patterns

At the end, the classified groups in each stage usually involve thousands of learning sequences with similar patterns. To improve the interpretability of each group and facilitate the comparison between groups, we employ a technique based on Gap-BIDE [3] to mine the frequent patterns in each group of segmented learning sequences. We employ this technique mainly because it introduces wild-cards when matching potential subsequences, which is consistent with our idea of allowing individual variance in each sequence. The most frequent patterns are extracted and displayed in the visualization designs (Fig. 1(d)). In this way, users can have a comprehensive understanding of the learning activities, as well as a contextualized focus on comparing different groups of behaviors.

References

[1] S. Guo, Z. Jin, D. Gotz, F. Du, H. Zha, and N. Cao. Visual progression analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics (Early Access)*, 2018.

[2] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[3] C. Li and J. Wang. Efficiently mining closed subsequences with gap constraints. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 313–322. SIAM, 2008.

[4] H. Lin, S. Gao, D. Gotz, F. Du, J. He, and N. Cao. Rclens: Interactive rare category exploration and identification. *IEEE Transactions on Visualization and Computer Graphics*, 24(7):2223–2237, 2018.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

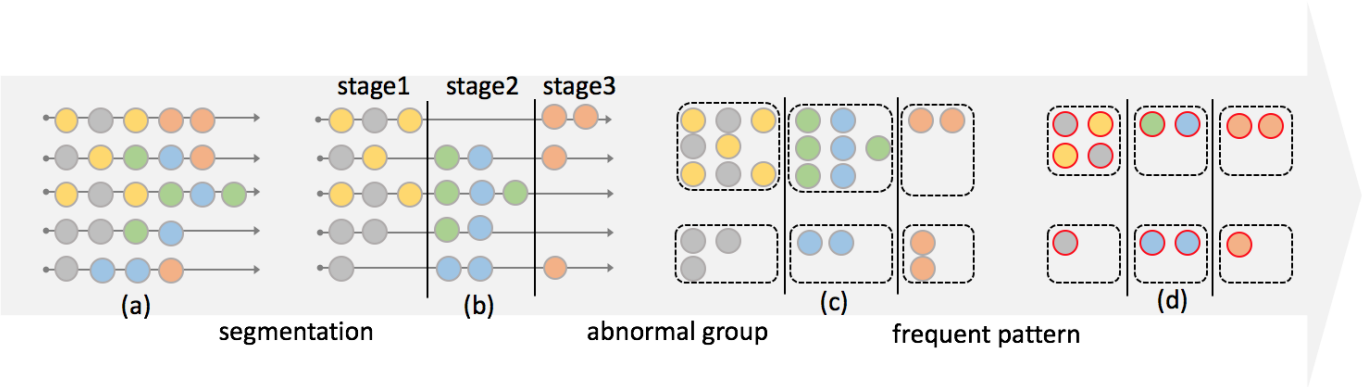

Fig. 1. The visual anomaly detection of learning sequence data includes three major steps: (1) sequence segmentation, (2) anomalous group identification, and (3) frequent pattern extraction.