seqClustR: An R Package for Sequence Clustering

Aditya Sharma
Playpower Labs
Gujarat
aditya@playpowerlabs.com

ABSTRACT

In this paper, we're going to describe the core features of the R package **seqClustR** [14] dedicated to sequence clustering. Sequence clustering is a data mining technique that groups similar sequences into clusters based on their similarities. Sequence clustering is useful when there are unknown number of similar sequences that need to be identified to gain valuable insights. The main feature of this package is that it provides easy access to different algorithms such as Edit Distance with Hierarchical Clustering, Markov Model-Based Clustering, Dynamic Time Warping, and K-Means to perform sequence clustering. We find that different algorithms can create very different clusters and lead us to very different conclusions. So to get a reliable understanding of the sequences, we need to apply various sequence clustering algorithms and explore the data from multiple points of view. This paper illustrates how you could create different clusters from different algorithms, extract event log data for each cluster and visualize them. We have provided an example in section 4 showing step by step procedure to run sequence clustering on the National Assessment of Educational Progress (NAEP) dataset.

Keywords

Sequence Clustering, Sequence Data, Event Logs, Visualization

1. INTRODUCTION

In this paper, we'll present the seqClustR package, which implements different clustering algorithms on sequence data in R. Sequence data contains information about the different activities performed over time. Working with sequence data can be hard sometimes; but it can probably provide better insights by making use of the temporal dimension [1]. When sequence data contains a lot of activities which lead to different types of sequences, the process model on the complete data can become too complex to interpret. To get better insights from the sequence data, we can cluster

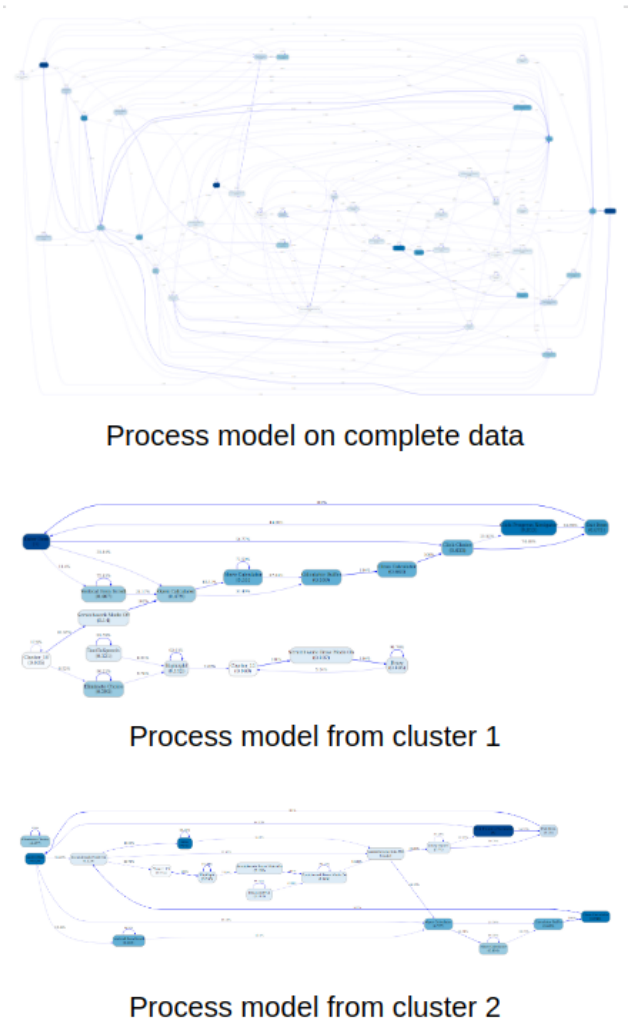

Figure 1: Process model on complete data Vs clustered data

similar sequences together and analyse them; sequence clustering can provide better explainable models than a general model fitted on complete data [2].

In process mining, sequence clustering plays an important role by grouping similar sequences and providing helpful insights, especially when we have little knowledge about various types of processes hidden in the data. For example, a business process data might have different versions of a single process within it. Sequence clustering is used in different domains: in bioinformatics to group similar biological sequences; in marketing to identify different purchasing behaviors of customers; in education to identify different learner behaviors. In education, while using digital content learners exhibit different types of behaviors: gaming the system behavior; off-task behavior; carelessness behavior. Performing sequence clustering on educational sequence data can help us group similar learners together and then we can compare their performance to understand which learner behavior models have a positive impact on learners' growth.

2. SEQUENCE CLUSTERING IN EDUCATION

In recent years, a lot of educational data has been collected through digital learning platforms at an increasing rate. With more learners joining the digital platforms and more types of content being created for them, we are observing a lot of clickstream data. Clickstream data shows us the sequence of activities that learners went through while being on the platform. Gaining insights from such data could help us identify different learner behaviors [16, 6]; improve the platform design based on learner's interactions on the platform [18, 9]; and provide adaptive content to individual learners to better support their learning [7, 8]. To gain insights from sequence data a few techniques have been used in the education data mining community such as Association Rule Mining [5], Sequential Pattern Mining [19], Process Mining [17], Graph-Based Analysis [10], and Curriculum Pacing [12].

Understanding a learner behavior from process models using sequence data can be complex with high-dimensional data. The sequence data can contain a high number of distinct activities and all the activities need not be equally important or have enough learner data; we could perform exploratory data analysis on the data to understand which activities are important. Activities that don't provide much information can make the models more complex and harder to interpret [4]. We should try to reduce the complexity of the sequence data by preprocessing the data before doing any sequence data analysis. To illustrate this by an example, Figure 1 shows three different process maps of learner action sequence data from the NAEP assessment. The complex process map at the top is made from all of the sequences in the dataset (N = 1009), and the two process maps below it are made from two distinct clusters of sequences (N1 = 726, N2 = 283). We can observe that the complexity of the process maps of clustered sequences is lower than the process map of complete data; it is much easier now to analyse the process map and make hypotheses about learner behavior.

3. seqClustR FRAMEWORK

This package is designed to cluster sequence data. The package uses event log [3] as an input for the data. Event log are very commonly used to store the user behavior data. They indicate the sequence of actions a user takes over time, along with added metadata of the event. For process analysis, event log is an essential data format. Event log contain three major attributes: case, activity, and timestamp. A case can be defined as a sequence of activities performed over time, and an event log represents one or more cases.

Once we have the data in event log format, we can perform sequence clustering by passing it into one of the sequence clustering functions. The output of the function would be a list containing the fitted model and a data frame having the case to cluster assigned mapping. To run analysis on individual clusters, we need event logs for each cluster for which we have written a function `split_event_log` which takes event logs and the clusters assigned data frame as inputs, and returns a list of event log by cluster. To visually observe the differences between sequence clusters and comparing different clustering algorithms we've used the fuzzymineR package [11] as it is readily available in R. There are some other tools available as well for visualizing sequence data, like ProM and Disco.

3.1 Clustering Algorithms

3.1.1 Edit Distance Clustering

(`seq_edit_distance_clustering`) Edit Distance between two sequences can be defined as the least number of activities required to be added, subtracted, or substituted to convert one sequence into another. For example, with 1 addition we can convert the sequence (a, b, c) into another sequence (a, b, c, d), so the Edit Distance between them is 1.

For clustering the sequence data with the help of Edit Distance, we chose Hierarchical Clustering Algorithm. To make it convenient to choose the number of clusters, we plot the Hierarchical Clustering tree for the users so that they can make an informed decision about it.

3.1.2 Markov Model Based Clustering

(`seq_markov_clustering`) Markov Model-based clustering is a probabilistic model-based approach to cluster sequence data. The Markov Model-based clustering method is similar to K-Means, where the cluster centroids in Markov Model are Markov transition probability matrices and the data points are Markov transition probability matrices for the sequences. An entry A_{ab} in a Markov transition probability matrix A can be interpreted as the conditional probability that a learner will do activity b after doing the activity a, independent of any previous activities.

3.1.3 Dynamic Time Warping (`seq_dtw_clustering`)

Dynamic Time Warping is a distance-based clustering algorithm that measures similarity between two temporal sequences, which can vary in speed. In the context of educational data, we can say that two learners that perform a sequence of activities in the same order but vary in the number of times each activity was done would have 0 Dynamic Time Warping distance. For example, two learners with sequences (a, b, c) and (a, b, b, c, c) would have 0 Dynamic Time Warping distance. Dynamic Time Warping can

be used to analyze any data that contain a sequence of actions over time, it has also been used to cluster educational sequence data previously [15]. We've used Dynamic Time Warping clustering algorithm from the R package Dynamic Time Warping [13] in our seqClustR package.

3.1.4 K-Means Clustering

(`seq_kmeans_clustering`) K-Means Clustering Algorithm identifies K number of centroids and allocates each data point to the nearest cluster to minimize the within-cluster sum of squares.

In our approach, we calculate the number of times each activity was done for each case and then normalize it by the sequence length. These features were then passed to K-Means Clustering Algorithm.

4. EXAMPLE

In this section, we'll explain how practitioners can use the package to perform sequence clustering on their data. We'll show an example of how to prepare the sequence data that can be passed to the clustering algorithms and then how we can visualize the clusters.

4.1 Data

We've used NAEP's Process Data for Math test from 2017 which included data of 2500 learners. NAEP test is used by the US government to measure learner knowledge across the country. This data was part of the NAEP Data Mining Competition 2019 whose goal was to find effective and ineffective learner test-taking behaviors. The test was divided into two blocks of 30 min where Block A's data was supposed to be used to predict learner's performance on Block B. There were 8 different question types but for the analysis, we're only working with data from Block A and considering Multiple-Choice Questions as it had the majority of the questions in the test.

Variable	Description	Attributes of Event Logs
STUDENTID	Unique identifier of the learner	Case ID
Block	Block of the NAEP test, A or B	-
Accession Number	Unique question identifier	-
ItemType	Type of Question	-
Observable	learner Activity	Activity ID
ExtendedInfo	Metadata of learner's Activity	-
EventTime	Event Timestamp	Timestamp

Table 1: Columns of the NAEP Process Data with Attributes of Event Logs

Before applying any clustering algorithm to the data, we need to perform data preprocessing steps to reduce data complexity. Our data preprocessing steps included removing activities that didn't have enough learner events and removing events where data for any of the event log attributes mentioned in table 1 were missing.

Figure 2: Hierarchical Cluster Plot

For clustering, we need to convert the sequence data into event log format. We have used STUDENTID as Case ID, Observable as Activity ID, and EventTime as Timestamp. For the missing attributes, we can manually add them [3]: Lifecycle ID was added as complete, Activity Instance ID was added as row number, and Resource was added as NA.

```
library(tidyverse)
library(bupaR)

event_log <- sequence_data %>%
 arrange(EventTime) %>%
 mutate(lifecycle_id = 'complete',
        resource = NA,
        row_num = 1:nrow(.)) %>%
 eventlog(case_id = "learnerID",
          activity_id = "Observable",
          activity_instance_id = "row_num",
          lifecycle_id = "lifecycle_id",
          timestamp = "EventTime",
          resource_id = "resource")
```

4.2 Clustering

After converting the sequence data into event logs format, we can start applying clustering algorithms on it. As an example, we've performed Edit Distance Based Hierarchical Clustering on the event log data that we prepared in Section 4.1. When we pass the event log into the respective function, we are shown a Hierarchical Cluster Tree (Figure 2) in the graphics window, and we get a user prompt to select whether we would like to cut the tree by number of groups or by height. On selecting one of the options we need to enter the value corresponding to it. The output of the function would be a list containing the fitted model and the cluster assignments by **case_id**.

```
library(seqClustR)
```

```
cluster <- seq_edit_distance_clustering(
           event_log)
```

4.3 Visualizing Clusters

Visualization is a powerful tool that can help us explore common behavioral patterns exhibited by learners. We've got two clusters from the event log in Section 4.2, to find out how both of them differ from each other based on common learner behavior patterns we'll make visualizations for these clusters (Figures 3 and 4). We've made process models for both the clusters using fuzzymineR package [11].

```
# Get event logs by cluster as a list.

event_log_2 <- split_event_log(
                  eventlog,
                  cluster$cluster_assignment)

library(fuzzymineR)

# Process Model for Cluster 1

metrics <- mine_fuzzy_model(
                  event_log_2[["1"]])

viz_fuzzy_model(metrics = metrics,
                  node_sig_threshold = 0.1,
                  edge_sig_threshold = 0.3,
                  edge_sig_to_corr_ratio = 1)
```

4.4 Discussion

Performing sequence clustering using hierarchical clustering with edit distance method on the NAEP data provided us two clusters that exhibit different behaviors. Cluster 1 (Figure 3) shows that for 56% of the times learners directly click on one of the choices after they have presented a multiple choice question; 32% of the times when learners are presented with a multiple-choice question they first use the calculator and then click on one of the choices. The different behaviors exhibited by learners in cluster 1 could be because of varying item difficulties, where an easy item might not require the use of the calculator but for more difficult item learners might need it.

In cluster 2 (Figure 4), learners show a different behavioral pattern, they tend to go through different steps before clicking on one of the choices; 39% of the times when learners are presented with a multiple-choice question they first use the calculator but unlike cluster 1, they do some scratchwork before clicking on one of the choices; learners tend to click on different choices a significant number of times before submitting their final answer and moving on to the next item; learners also use the eliminate answer choice tool significantly in cluster 2.

5. CONCLUSION

The R package seqClustR provides the means to perform different clustering algorithms on sequence data by reducing the complexity of preparing data for each algorithm in a different way, by just converting the sequence data into event logs we can run multiple clustering algorithms and compare them. In this paper, we've shown how to prepare sequence data, perform sequence clustering, and visualize the clusters. In future, we would like to add more clustering algorithms and add a functionality in the package to do qualitative comparisons of clusters as well.

6. ACKNOWLEDGMENTS

We would like to thank the organizers of the 2019 NAEP Data Mining Challenge for providing the data. We hope that the seqClustR package would support and inspire more research on sequence data.

7. REFERENCES

[1] H.-P. Blossfeld, T. Schneider, and J. Doll. Methodological advantages of panel studies. designing the new national educational panel study (neps) in germany. *Journal for Educational Research Online*, 1(1):10–32, 2009.

[2] A. Bogarín, C. Romero, R. Cerezo, and M. Sánchez-Santillán. Clustering for improving educational process mining. In *Proceedings of the fourth international conference on learning analytics and knowledge*, pages 11–15, 2014.

[3] bupaR. *Creating Event Logs*. `https://www.bupar.net/creating_eventlogs.html`.

[4] A. H. Cairns, B. Gueni, M. Fhima, A. Cairns, S. David, and N. Khelifa. Process mining in the education domain. *International Journal on Advances in Intelligent Systems*, 8(1):219–232, 2015.

[5] E. García, C. Romero, S. Ventura, C. de Castro, and T. Calders. Association rule mining in learning management systems. *Handbook of educational data mining*, pages 93–106, 2010.

[6] C. Hansen, C. Hansen, N. Hjuler, S. Alstrup, and C. Lioma. Sequence modelling for analysing student interaction with educational systems. *arXiv preprint arXiv:1708.04164*, 2017.

[7] M. Köck and A. Paramythis. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*, 21(1):51–97, 2011.

[8] J. D. Lomas, J. Forlizzi, N. Poonwala, N. Patel, S. Shodhan, K. Patel, K. Koedinger, and E. Brunskill. Interface design optimization as a multi-armed bandit problem. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 4142–4153, 2016.

[9] R. Luckin et al. Modeling learning patterns of students with a tutoring system using hidden markov models. *Artificial intelligence in education: Building technology rich learning contexts that work*, 158:238, 2007.

[10] N. Patel, C. Sellman, and D. Lomas. Mining frequent learning pathways from a large educational dataset. *arXiv preprint arXiv:1705.11125*, 2017.

[11] N. Patel and T. Shah. *fuzzymineR*. `https://github.com/nirmalpatel/fuzzymineR`.

[12] N. Patel, A. Sharma, C. Sellman, and D. Lomas. Curriculum pacing: A new approach to discover instructional practices in classrooms. In *International Conference on Intelligent Tutoring Systems*, pages 345–351. Springer, 2018.

[13] A. Sarda-Espinosa. *dtwclust: Time Series Clustering Along with Optimization for the Dynamic Time Warping Distance*, 2019. R package version 5.5.6.

[14] A. Sharma and N. Patel. *seqClustR*. `https://github.com/aditya9352/seqClustR`.

[15] S. Shen and M. Chi. Clustering student sequential trajectories using dynamic time warping. *International Educational Data Mining Society*, 2017.

[16] B. Shih, K. R. Koedinger, and R. Scheines. Discovery of student strategies using hidden markov model clustering. In *the Proceedings of the 6th International Conference on Educational Data Mining*. Citeseer, 2010.

[17] N. Trcka, M. Pechenizkiy, and W. van der Aalst. Process mining from educational data. *Handbook of educational data mining*, pages 123–142, 2010.

[18] O. Zaiane. Web usage mining for a better web-based learning environment. 2001.

[19] M. Zhou, Y. Xu, J. C. Nesbit, and P. H. Winne. Sequential pattern analysis of learning logs: Methodology and applications. *Handbook of educational data mining*, 107:107–121, 2010.

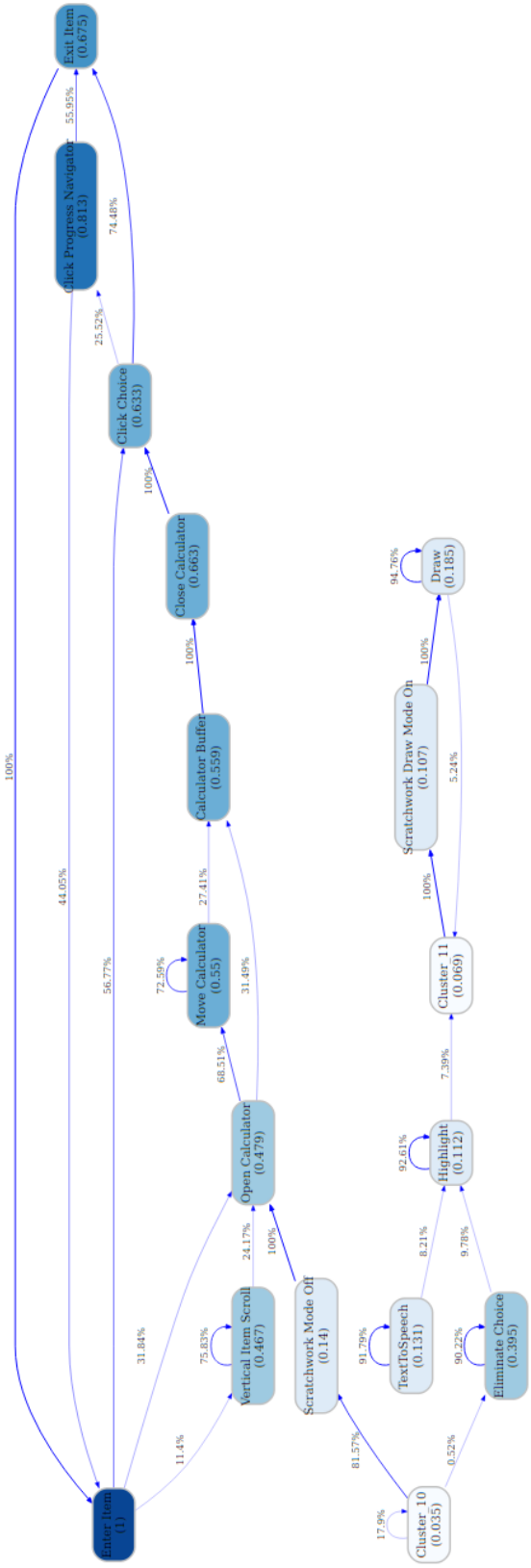

Figure 3: Process Model for Cluster 1

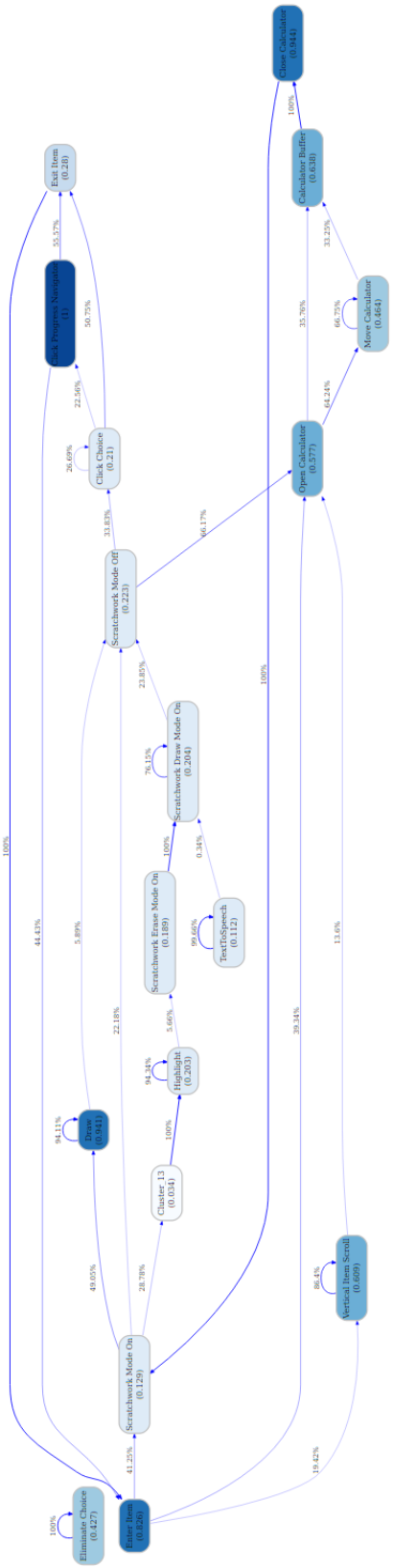

Figure 4: Process Model for Cluster 2