Classifying Event Sequences using R Clustering Algorithms

by Jayson M. Webb, PhD

Abstract Data simulation experiments in 'R' (R Core Team, 2016) are used to compare the performance of 2 clustering algorithms, 'pam' and 'diana', from the **cluster** package (Maechler et al., 2016). Clustering is used to classify letter sequences that vary in their between group similarity as measured by the Jaccard index (Levandowsky and Winter, 1971). A method is proposed for improving classification accuracy that works with both clustering methods. One application of the method for improving classification accuracy is classifying clickstreams, the sequence of visits to pages or other events that result from the use of a mobile or web application.

Motivation

The reasons for doing these data simulation experiments were:

- Identify a simple method for grouping similar sequences of events so that the differences between groups can be easily understood.
- Develop a deep understanding of the factors that affect the accuracy of classifying sequences.

Event Sequences and Event Spaces

These simulations assume that an event sequence (an observation) is sampled from an event space (a hypothetical construct). In the context of clickstreams, an event space could be thought of as a tendency to do something or a characteristic pattern of behavior or the set of possible states related to some goal.

Suppose that our hypothetical event space E_1 consists of the first 10 letters of the alphabet, A through J, where each letter represents a type of event. Let's then suppose that an event sequence S_1 is sampled from E_1. So, S_1 = 'F,J,B,C,G,A' could be an event sequence generated by sampling from the event space E_1 = 'A,B,C,D,E,F,G,H,I,J'. If an event space is a hypothetical construct, where does it come from in observed data? What the simulation experiments will show is that event spaces can be recovered from the observed data and are the key to improving classification of observed event sequences.

The Jaccard Index of Similarity

The Jaccard index can be used to calculate the similarity between two event sequences, $J(S_1, S_2)$, two event spaces, $J(E_1, E_2)$, or between an event sequence and an event space, $J(S_i, E_i)$. The Jaccard index for two sequences can be calculated as the size of the intersection of the sequences divided by the size of the union of the two sequences.

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}, \text{ where } 0 \leq J(S_1, S_2) \leq 1$$

The simulations described in this paper used the following 'R' code for calculating the Jaccard index. A custom function was used to ensure a deep understanding of what was happening and to allow for customizations in the future.

```
  jaccard <- function(S1,S2) {
  x <- length(intersect(S1,S2)) / length(union(S1,S2))
  return(x)
}
```

For example, for the sequences S_1 = 'F,J,B,C,G,A' and S_2 = 'H,N,M,J,I,O', the Jaccard index $J(S_1, S_2) = 0.091$. The intersection of S_1 and S_2 is 'J' which has a size of 1. The union is 'A,B,C,F,G,H,I,J,M,N,O' which has a size of 11. For an event space example, $J(E_1, E_2) = 0.333$ for E_1 = 'A,B,C,D,E,F,G,H,I,J' and E_2= 'F,G,H,I,J,K,L,M,N,O'.

Sequence information is lost using the Jaccard index. However, it is very straightforward and can still yield insights based on the 'vocabulary' of the sequences, if not the 'syntax' and could be used as a first stage analysis.

Data Simulation and Clustering

The simulation experiments were intended to test the following hypotheses about what affects how accurately event sequences can be classified. Event sequences should be more easily classified when:

1. Sequences are similar to the event space they are sampled from and different from other event spaces. Event sequences become more similar to their event space when they are long relative to the event space.
2. Sequences are similar to other sequences from their event space and different from sequences sampled from other event spaces. This is also more true when event sequences are long. For example, for the event space consisting of the letters A through J, the short sequences 'A,B' and 'C,D' don't overlap (have 0 intersection), so have 0 similarity. Sequences that are longer than 1/2 the length of the event space (even if the letters are not sequential) must have some overlap. For example, the sequence consisting of the letters A through F (length 6) and the sequence E through J (length 6), both sampled from the space consisting of the letters A through J, have an intersection of length 2, consisting of the letters E and F.
3. Event spaces are different from each other. When event spaces have a large intersection, the sampled sequences can be equally similar to both event spaces. This is again more true for shorter sequences.

The items above boil down to this: classification should be better when event sequences are longer and event spaces are less similar. This seems pretty intuitive, but let's look at some data.

Experiment 1

In this experiment, we examine the effects on classification accuracy of the similarity between an event sequence and its event space, $J(S_i, E_i)$, and the similarity between two event spaces, $J(E_1, E_2)$. Simulations were done for 16 combinations of $J(S_i, E_i)$ (0.2, 0.4, 0.6, 0.8) and $J(E_1, E_2)$ (0, 0.11, 0.33, 0.67) and were replicated 32 times at each level. The average classification accuracy was calculated for the 32 replications for each of the 16 combinations of $J(S_i, E_i)$ and $J(E_1, E_2)$. Each replication consisted of 200 simulated event sequences from each of two event spaces. Each event space was 10 letters long and event sequences of length 2, 4, 6, and 8 were randomly sampled, without replacement. All of this was done with both divisive hierarchical clustering and partitioning around medoids, as implemented in the packages 'diana' and 'pam', respectively, from the **cluster** package (Maechler et al., 2016).

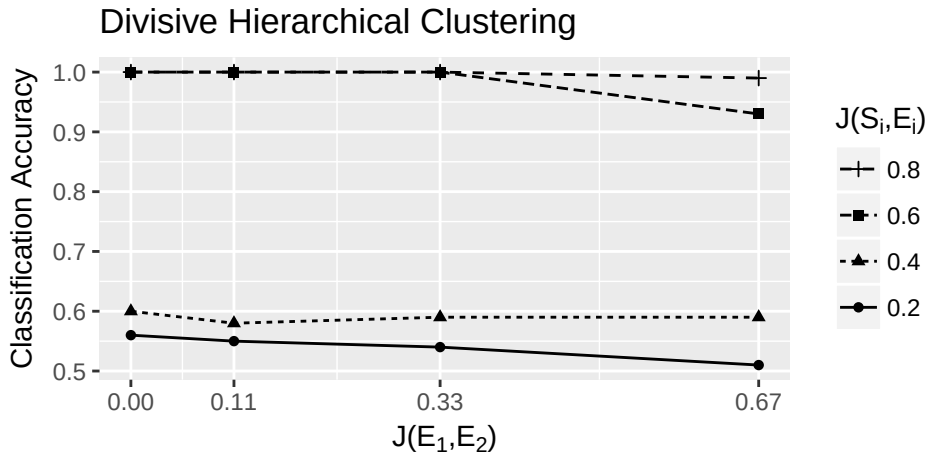

Figure 1: The effects on classification accuracy of the similarity between a sequence and its event space $J(S_i, E_i)$ and the similarity between two event spaces $J(E_1, E_2)$. Similarity is measured by the Jaccard index. The 4 levels of $J(S_i, E_i)$ are shown by 4 different lines. The 4 levels of $J(E_1, E_2)$ are the x-axis values.

As we hypothesized, classification accuracy is better when event spaces are not similar to each other and when event sequences are longer. This is evident in both Figure 1 and Figure 2.

Figure 1 shows, for divisive hierarchical clustering, that when $J(S_i, E_i)$ is greater than 0.5 (the sequences are long relative to the event spaces) and $J(E_1, E_2)$ is less than 0.5 (the event spaces are not very similar), our classification accuracy is perfect. When the event spaces are very similar, $J(E_1, E_2)$=0.67, classification accuracy falls for all levels of $J(S_i, E_i)$, but longer sequences are always classified more accurately than shorter sequences.

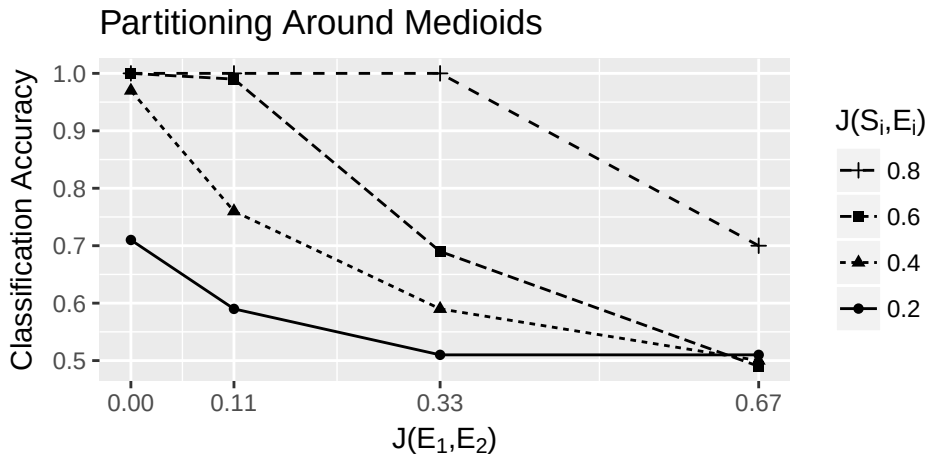

Figure 2: The same as Figure 1, except using partitioning around medoids instead of divisive hierarchical clustering.

Figure 2 leads to the same overall conclusions as Figure 1, but with some interesting differences. When $J(E_1, E_2)$ is 0.67, the best classification accuracy for partitioning around medoids was 0.70 for $J(S_i, E_i)$ = 0.8. Classification accuracy in the same condition using divisive hierarchical clustering was 0.99. Figure 2 shows much better classification accuracy overall for the two lower values of $J(S_i, E_i)$ than we saw in Figure 1. So, the two clustering algorithms certainly have different characteristics.

Experiment 1 provided support for our initial intuitions. Classification accuracy is better when event spaces are not similar and when event sequences are longer. In real life, $J(S_i, E_i)$ is a random variable, not a constant as it was in the different conditions of experiment 1. Experiment 2 will repeat experiment 1 except that $J(S_i, E_i)$ will be a random variable ranging between two values.

Experiment 2

This experiment will be just like experiment 1, except the $J(S_i, E_i)$ similarity will vary randomly between 0.8 and one of 0.6, 0.4, or 0.2. We'll call this random variable $J(S_i, E_i)$_range. For divisive hierarchical clustering shown in Figure 3, the pattern for experiment 2 is similar to that for experiment 1. Once again, we got perfect classification accuracy when $J(S_i, E_i)$_range values are all greater than 0.5 and $J(E_1, E_2)$ is less than 0.5. The other conditions also show a similar pattern to experiment 1. Of particular interest is that when $J(S_i, E_i)$ varied randomly between either 0.2 or 0.4 and 0.8, classification accuracy was only a little better than experiment 1 when the values were constant at 0.2 or 0.4.

Figure 4 shows that partitioning around medoids performs better overall than divisive hierarchical clustering, except when $J(S_1, S_2)$ is highest (0.67). This pattern is also similar to experiment 1.

We have learned 3 things so far about what affects classification accuracy when using either divisive hierarchical clustering or partitioning around medoids based on a Jaccard index similarity measure.

Classification accuracy is better when:

1. event spaces don't overlap very much (smaller value of the $J(E_1, E_2)$ variable).
2. event sequences are longer (larger value of the $J(S_i, E_i)$ variable).
3. event sequence variability is lower (a smaller $J(S_i, E_i)$_range).

We have also seen that divisive hierarchical clustering and partitioning around medoids have different characteristics under the conditions that we have simulated.

Let's assume for now that in real data, the event space is something we can't control. To control event sequence length and length variability, we could sample only the longest sequences from the data and create clusters based on those. This would address the finding that classification with longer sequences should work better and it will also lower the variability of sequence lengths. After the longer sequences have been classified, it should then be possible to extract the event spaces from the clusters created from the longest sequences. Then, the remaining sequences could be compared to the event spaces.

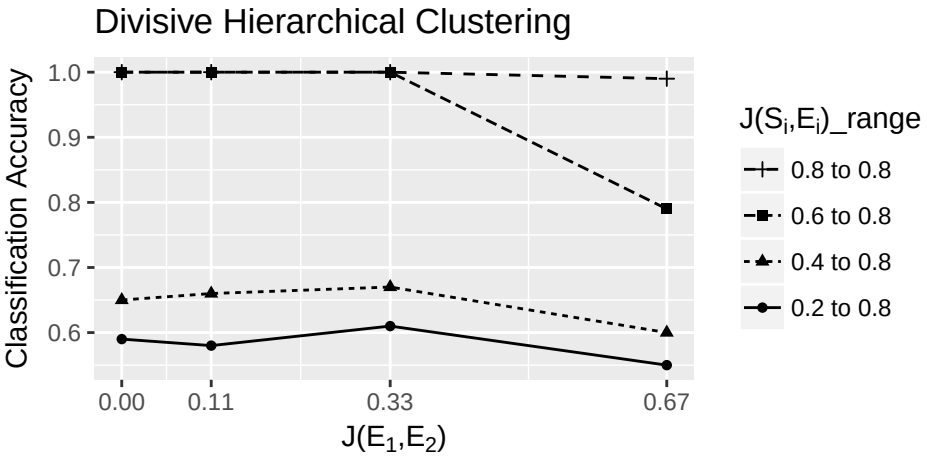

Figure 3: Effect of the and the similarity between two event spaces $J(E_1, E_2)$, and the similarity between an event sequence and event space as it varies randomly in $J(S_i, E_i)$_range. The four different lines show 4 $J(S_i, E_i)$_range values: 0.8 to 0.8 (no variation), 0.6 to 0.8, 0.4 to 0.8 and 0.2 to 0.8.

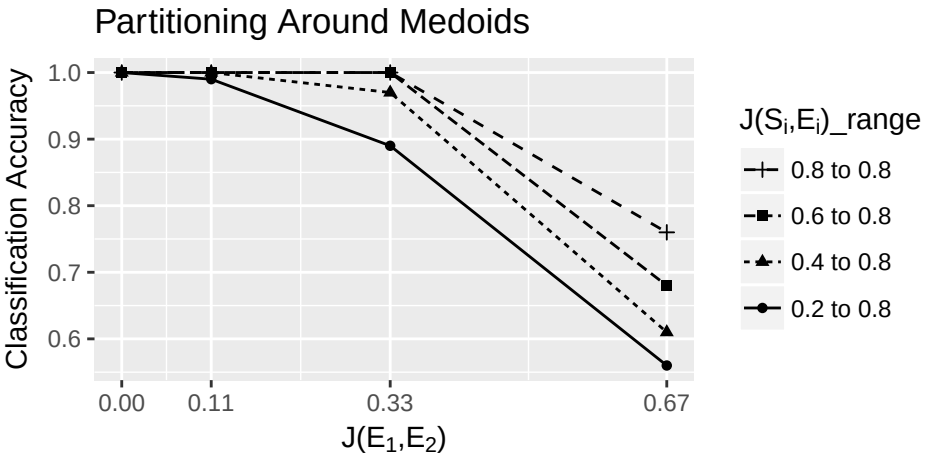

Figure 4: Same as Figure 3 except using partitioning around medoids.

Experiment 3

We will use a three stage approach to clustering that does the following:

1. Create clusters using only the longest sequences, defined as being at or above the 80th percentile in length.
2. Extract the event space for each cluster by taking the union of the events across all the sequences in the cluster. This process will be repeated 20 times in order to average out any small variations in the extracted clusters. In each of the 20 repetitions, clustering will be done with a random sample of 80% of the sequences to further minimize the effects of outliers. An event is included in the event space definition if it appears in at least 18 of the 20 repetitions.
3. Compare the remainining sequences to each extracted event space. The outcome will be either (a) the sequence is more similar to one of the event spaces, in which case it will be put in that cluster, or (b) the sequence will have equal similarity to both clusters, in which case it would be undefined and not put in any cluster.

We will use the toughest two conditions from experiments 1 and 2, $J(E_1, E_2) = 0.67$ (high event space overlap) and $J(S_i, E_i)$_range = 0.2 to 0.8. Everything else will be the same as it was in experiments 1 and 2:

- 2 events spaces, each of length 10
- 200 simulated sequences per event space
- classification accuracy averaged over 32 repetitions

Divisive Hierarchical Clustering Results

We had the simulation functions return the event spaces we recovered for each of the 32 repetitions so we could inspect them. We recovered the correct event spaces 31 out of 32 times (97%). The two event spaces were E_1 = the letters A to J and event space E_2 = the letters C to L.

When we look at the 31 cases where we correctly recovered the event space, our sequence classification accuracy was perfect, 100%. As a reminder, this is for the toughest two conditions from experiments 1 and 2, $J(E_1, E_2) = 0.67$ and $J(S_i, E_i)$_range = 0.2 to 0.8, which yielded a classification accuracy of 55% in experiment 2. To get this, we did have to throw out 1 of our replications because it didn't recover the correct event spaces. In real life, how would we know which ones to keep? Afterall, we wouldn't know the correct answer as we did in these experiments. We could do a large number of replications as we did here and use a 'majority wins' approach to select the results based on high agreement among recovered event space definitions. Understanding the events themselves would also help in applying top down reasoning to understand the nature of the event spaces and whether or not they make sense and could be helpful.

Another reason our classification was perfect is that we classified about 1/4 (27%) of the sequences as unclassifiable because they were equally similar to both event spaces. As an example of when that can happen, consider the event space E_1 as the letters A to J and event space E_2 as the letters C to L. The sequence '`F,G,H,I,J`' would be equally similar to both event spaces. Being able to inspect the set of unclassifiable sequences could be very useful. In the case of clickstreams, these might be the set of actions that don't distinguish groups, that is, things that everyone has to do or wants to do.

Partitioning Around Medoids Results

Only 5 of 32 repititions returned the correct event sequences (16%). Again, we picked the most difficult condition from the previous experiments, and the '`pam`' algorithm always performed worse than the '`diana`' algorithm under those conditions. Our conclusion is that for subsequent work, we're going to focus on the divisive approach. Wang et al. also came to the conclusion that a divisive hierarchical clustering approach was best for classifying clickstreams (Wang et al., 2016).

Experiment 4

Experiment 4 extends our method to 4 clusters. With 4 clusters, we recovered the correct event spaces 32 of 32 times and our classification accuracy was 100%. About 13% of the sequences were unclassifiable. In the Appendix, you can see the event space results from one of the 32 trials in experiment 4. It shows what the overlapping event spaces were. Each event space had a similarity $J(E_i, E_i + 1) = 0.43$ to the next most similar event space.

Conclusion

Data simulations show that the classification accuracy for event sequences can be quite high if the longest subsequences are classified first, then the event spaces are recovered, then the remaining event sequences are compared to the event spaces. In this paper, the event spaces were overlapping but always had some unique events. In the next sequence, event sequences will differ only by the sequential constraints among events.

Appendix

```
$event_space1
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"

$event_space2
 [1] "E" "F" "G" "H" "I" "J" "K" "L" "M" "N"

$event_space3
 [1] "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R"

$event_space4
 [1] "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V"
```

Bibliography

M. Levandowsky and D. Winter. Distance between sets. 1971. URL http://dx.doi.org/doi:10.1038/234034a0. [p1]

M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2016. R package version 2.0.5 — For new features, see the 'Changelog' file (in the package source). [p1, 2]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/. [p1]

G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao. Unsupervised clickstream clustering for user behavior analysis. 2016. URL http://dx.doi.org/10.1145/2858036.2858107. [p5]

Jayson M. Webb, PhD
4525 Navajo Pl, Boulder, CO, 80303
United States
jayson.m.webb@gmail.com