Optimizing K-Means Tuning, Fall 2023

Bryan Huckleberry

October 22, 2023

1 Memory Efficiency

For background on the algorithm being implemented, see reference [1].

My version of similarityRcpp only has to store numerical doubles and the given membership vectors. This is because Ben-Hurs correlation metric for similarity is only based on 3 dot products. The calculation for each dot product shown on page 3 of [1] is only a sum of the product of two boolean values. Instead of storing each boolean in a qxq C matrix, we can just calculate every term of each dot product directly. Additionally, each C matrix, by definition, is symmetric with values of 0 on the diagonal. Thus, we only need to calculate two booleans for every i $<$ j and that is enough to update each dot product needed for the similarity calculation.

2 Speed Test for Different Values of M

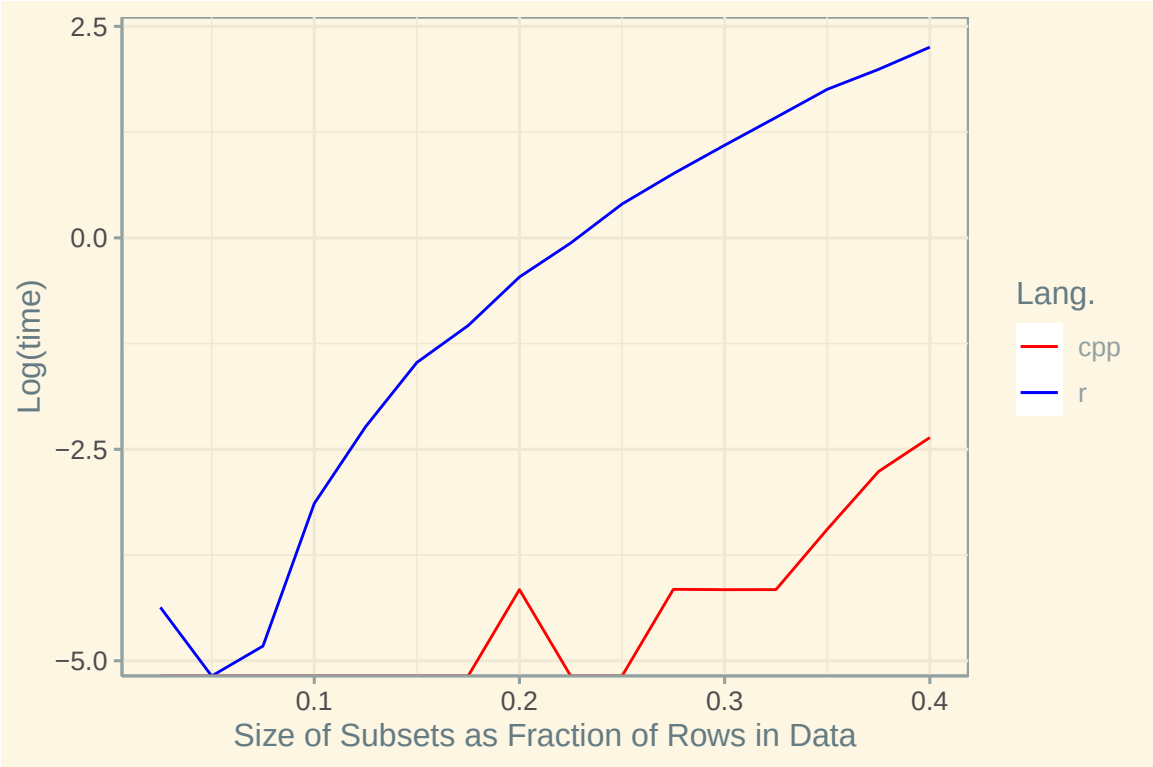

Figure 1: Comparison of Computing Time for Each Implementation of Similarity

```
## [1] "The average ratio between the time of my r implementation vs my cpp implementation is 147.4"
```

The time of my c++ implementation is faster to the point where I needed to use a logarithmic scale in fig.1. There is inherent variability due to the randomness of subset selection from LingBinary. For smaller values of m there is a lower likelihood of a large intersection between each subset from the table. From this seed the general trend of c++'s efficiency is visible.

3 Algorithm

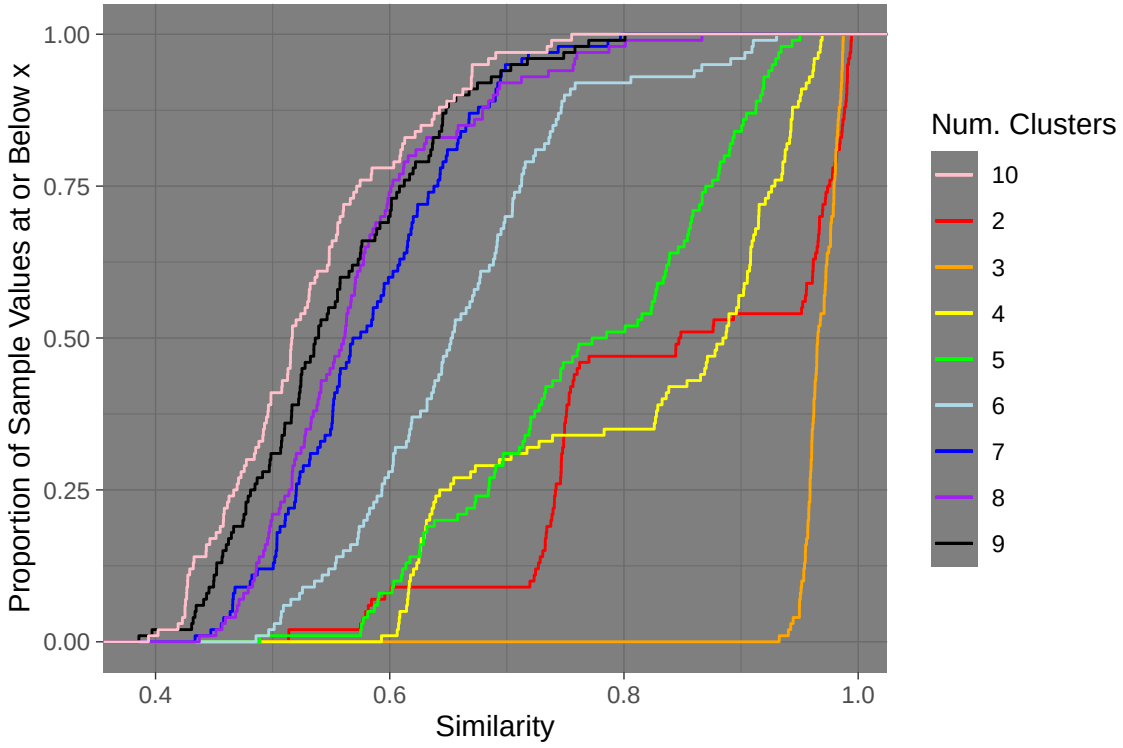

Figure 2: Empirical CDF of Similarity Values in Results for Each K

Based on the outcome of my algorithm, I would choose the optimal value of k for the LingBinary dataset to be 3. This is due to the observation that - more so than any other value of k - the overwhelming majority of similarity measurements are greater than 90%. This means using k = 3 is producing clusters of high similarity across many perturbations of samples from the dataset. Observing the other options, each choice has at least 25% of similarity values being less than 80%. Clearly there is a huge gap in the average similarity in clusters when selecting 3 groupings versus any other choice.

There are degrees to which I would have faith in this methodology. I believe the algorithm does a good job of identifying the optimal number of clusters when the relative size of clusters in the dataset doesn't differ tremendously. The issue with subset selection is that smaller, outlier clusters which may be present in the data are going to be more difficult to identify across subsets since it is more likely a small subset will have a larger majority of points from the larger clusters. Here the algorithm shows the most stability for 3 clusters, but it is very possible that a really compact, yet smaller, cluster exists which is not being captured in the intersection between each random subset. Therefore, most of the subsets compared will try to force points from the 3 dominating clusters into 4 clusters and will therefore produce lower similarity. Stability and robustness come at a cost of obscuring abnormalities in the dataset. Subsets of data are less likely to contain outlier values because these values represent a much smaller proportion of the entire dataset.

To summarize, while I have trust in this method I do not consider it to be a universally optimal choice for tuning the cluster parameter. I have confidence that the Ben-Hur algorithm will identify the larger groupings within a given dataset for a sufficient choice of N. Therefore, I believe it is a good choice for looking at large-scale partitions of the dataset. However, the algorithm is less reliable for identifying relatively small, yet potentially very compact, groups.

4 Academic Integrity Statement

All work presented in this report was conducted by myself. This includes the creation of all plot, analysis of findings, and written information. Any information collected from external sources has been cited accordingly.

5 Collaborator

Worked without collaborators for this report

6 Bibliography

[1] Asa Ben-Hur, André Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In Pacific symposium on biocomputing, volume 7, pages 6–17, 2001.