Lab 3 - Parallelizing k-means
Stat 215A, Fall 2017

October 23, 2017

1 Introduction

Clustering is a widely used exploratory analysis method for learning about how observations can be grouped together. When we have a prior knowledge about how many groups are present in the data and which observations correspond to a certain group, clustering can be used as a way of verifying the quality of data. If we are ignorant about the structure of the data, clustering can be used as a way to discover latent groupings.

However, there is no fixed definition of what a latent grouping is. Ben-Hur et al proposes the use of stability analysis as a way of detecting the presence of clusters in data. The idea is that a true and meaningful clustering structure should be robust to perturbations or the addition of noise to data. The proposed method is to subsample two sets of data, obtain a cluster solution in each subsample for varying number of cluster sizes, and compute similarity of the two cluster solutions. The process is repeated many times until we get a distribution of the similarity for each candidate cluster size. The optimal number of clusters is the one that yields the most stable similarity result.

In this report, we use the linguistic data from lab 2 and apply the stability based method for learning the structure of dialect patterns across respondents in the United States. In the next section, we explain a metric that was used to compare similarity of cluster solutions. We will also compare the algorithm written in R and C++ to obtain the similarity matrix and discuss the stability of the result.

2 Clustering similarity measures

There are multiple ways to measure similarity between two cluster solutions. One that we use in this report is the *matching coefficient*.

Let $X = x_1, ..., x_n$ and $x_i \in R^d$ be the dataset to be clustered. A L is a partition of dataset X into k groups. Then C_{ij} can be defined as:

$$C_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ belong to the same cluster and } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

Given two matrices $C^{(1)}$ and $C^{(2)}$, we can consider N_{ij} for $i, j \in 0, 1$ as the number of observations on which $C^{(1)}$ and $C^{(2)}$ have values i and j, respectively. The *matching coefficient* is defined as the proportion of observations on which the two cluster solutions C_{ij} agree:

$$M(L_1, L_2) = \frac{N_{00} + N_{11}}{N_{00} + N_{01} + N_{11} + N_{10}}.$$

Other similarity measures include cosine similarity measure and the *Jaccard coefficient*. The *Jaccard coefficient* is similar to the *matching coefficient*, except the "negative matches" (i.e. N_{00}) are ignored. The cosine similarity measure involves a dot product, which computes the number of pairs of points clustered together:

$$cor(L_1, L_2) = \frac{\langle L_1, L_2 \rangle}{\sqrt{\langle L_1, L_2 \rangle \langle L_1, L_2 \rangle}},$$

where

$$\langle L_1, L_2 \rangle = \langle C^{(1)}, C^{(2)} \rangle = \sum_{i,j} C_{ij}^{(1)} C_{ij}^{(2)}.$$

3 The model explorer algorithm

To compare stability of cluster solutions across different cluster sizes, we need a distribution of similarity for each cluster size. This can be achieved by the following algorithm:

1. Choose a cluster size k to consider
2. Draw two sets of subsamples of size n from original data
3. For each subsample, perform K-means clustering
4. Extract q observations that are common to both subsamples
5. Generate matrices $C^{(1)}$ and $C^{(2)}$ for the subsamples
6. Compute matching coefficient between $C^{(1)}$ and $C^{(2)}$
7. Repeat steps 1 to 6 for t times

Cluster size k we consider is from 2 to 10. As a rule of thumb, the proportion of observations we subsample should be between 0.2 and 0.8. If the proportion is too small, all clusters might not be represented in the subsample. In this case, we set proportion to be 0.5, resulting in 22,576 observations. The number of iterations t was 100.

K means clustering was conducted based on the raw binary data. Clustering solutions may be different and possibly more stable if we perform the analysis on a less sparse data projected on the first few principal components (e.g. data that aggregates the binary data by zipcode). However, we proceed with using the given binary data as the focus of the lab is to conduct stability analysis.

4 Computation in R and C++

Similarity matrix is a 100 by 9 matrix showing 100 iterations of similarities between two cluster solutions across 9 different cluster sizes. Perhaps the most straightforward way to generate the similarity matrix is to generate C_{ij}, which is a q by q matrix of 0 and 1 entries indicating whether observation i belongs to the same cluster as observation j. Thus, C_{ij} is a symmetric matrix by definition – if we let c_{ij} be an element of the matrix C_{ij}, then $c_{ij} = c_{ji}$, and diagonal entries are 1. This means that we do not need to construct an entire matrix to compute the matching coefficient. Instead, we only need to know the upper triangular part of the matrix, which contains just $\frac{2(q-1)}{2}$ entries. If we let w denote the number of entries that are common in the upper triangular part of the two matrices, then the matching coefficient can be computed as:

$$M(L_1, L_2) = \frac{2w + q}{q^2}.$$

The algorithm was written both in R and in C++. C++ is a programming language that has imperative and generic programming features and has no memory overhead. We compare the time required in R and C++ to compute the matching coefficient given the upper triangular part of the binary matrix, for one iteration and a fixed k. For $k = 2$, R took approximately 6.7 seconds to run, whereas the time was 2.0 seconds for C++. For other k, C++ was also significantly faster than R.

Because the amount of computation is large, we also rely on the use of parallel processing in R, supported by the `doParallel` package and `foreach` package. In this case, we run the algorithm for 9 different cluster sizes ($k = 2$ to 10) in parallel.

The distribution of the similarities is compared for different values of k (Figure 1, 2). Variability is high for distributions of $k = 2$, 4 5, and 6. Cluster sizes $k = 7$ to 10 have relatively smaller variability but the peaks of the distributions appear to lie below 0.9. However, the histogram and cumulative distribution of $k = 3$ mostly take values above 0.9, suggesting that it has both low variability and high similarity. 3 clusters appears to be an optimal number.

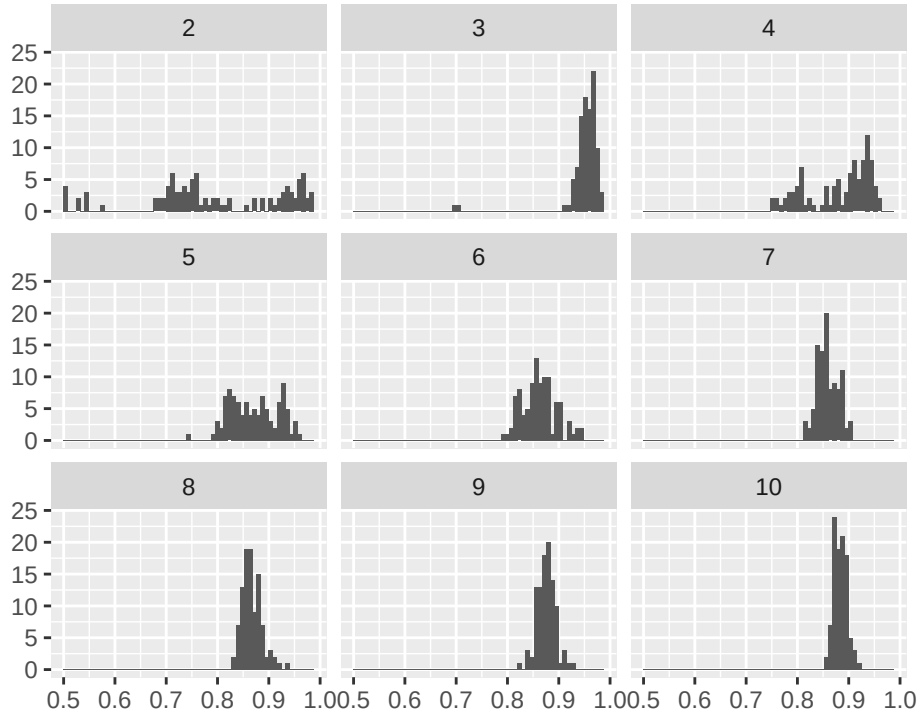

Figure 1: Histogram of the correlation similarity measure

5 Discussion

The stability based method indicates that 3 clusters is an appropriate solution. The method appears to be reasonable. For one, the cluster solution agrees with what was obtained by Calinski-Harabasz criterion, observed in lab 2, which is another method for selecting an optimum number of clusters. The criterion is computed as the ratio of overall between-cluster variance and overall within-cluster variance. The idea is that a natural cluster solution should have a high ratio, because each cluster is contained and well separated from each other. Thus, Calinski-Harabasz criterion is not a stability based method and represents the structure

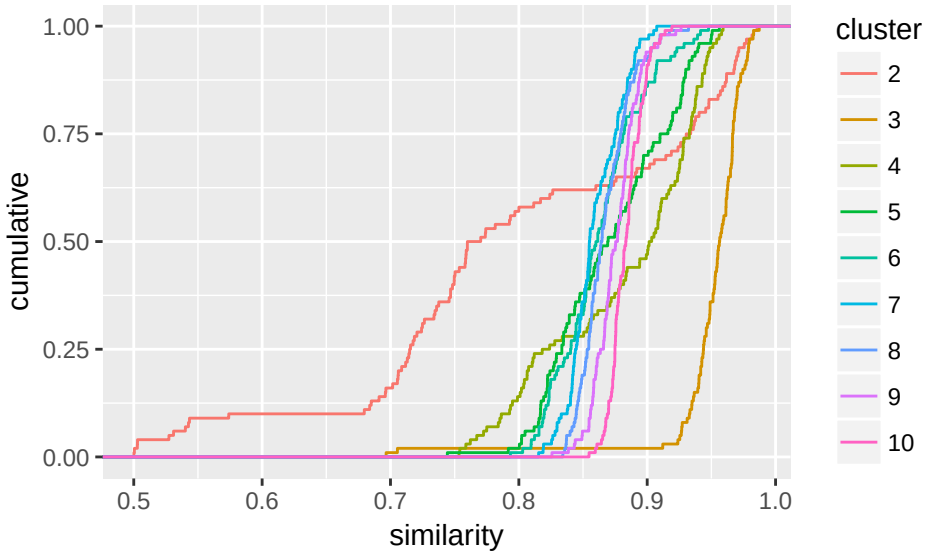

Figure 2: Overlay of the cumulative distributions for increasing values of k

of the cluster.

Another reason we consider the solution to be reasonable is because it agrees with our prior knowledge about how geographic location plays a major role in differentiating various dialects. In lab 2, we have learned that responses to dialect questions vary greatly across respondents in the south, midwest, and northeast.

One concern with the analyses is that we have used a sparse data to conduct K means clustering. While a sparse, binary data can technically be used to conduct cluster analysis, stability results may differ if we have used a less sparse data, such as a data that aggregates the binary responses by zipcode.

In this case, the distribution of the similarities showed that there is one cluster solution that has both high similarity and low variability. However, this may not always be the case. With some data, there may be multiple cluster solutions with the aforementioned features, and there may be a need to select the solution among these candidates based on methods that do not rely on stability analysis. Thus, the stability based method should be used as one of several methods for selecting an optimum number of clusters.

6 Conclusion

The report investigated a stability based method proposed by Ben-Hur et al. to select an optimum number of clusters. The method is an iterative process, where similarities are computed based on two subsamples of data, for different values of k. The algorithm was conducted using both R and C++. C++ showed significantly higher computation speed than R, even though the method for computing matching coefficient was the same in both languages. The distribution of similarities is then used as a guide for examining the stability of cluster solutions across different k. The optimum $k = 3$ yields both high stability and high similarities. While stability based approach for selecting k is effective, it should be used in combination with other methods, such as methods that rely on the structure of the clusters and prior knowledge about natural groupings.

References

[1] Asa Ben-Hur, Andre Elisseeff, Isabelle Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, 7:6-17, 2002.