CS-E4650 Methods of Data mining

Project Work

November-December 2020

Alex Herrero Pons 918697
Pablo Rosales Rodríguez 914769

Contents

1 Introduction

The present project has as goals clustering two data sets with different methods and comparing the obtained results making use of the provided metric, normalized mutual information, which has the following equation:

Given clustering $C_1, \ldots, C_k$ and classification $D_1, \ldots, D_q$

$$NMI = \frac{I(C, D)}{\sqrt{H(C)H(D)}}$$

where $I = \sum_{C_i \in C} \sum_{D_j \in D} P(C_i, D_j) \log \frac{P(C_i, D_j)}{P(C_i)P(D_j)}$ is mutual information and $H(C) = \sum_{C_i \in C} P(C_i) \log P(C_i)$ and $H(D) = \sum_{D_i \in D} P(D_i) \log P(D_i)$ are entropies.

2 Methods

2.1 First data set: genedata.csv

This initial data set consists of 7000 features and 694 items. In order to perform proper clusterization, the optimal number of clusters for the chosen method is selected based on the results that the normalized mutual information yields. This process is performed by a for loop that computes the clusterization following the given method and iterates the number of clusters storing the greatest result. Five different approaches have been considered, hence, a function called `clustering` that receives the data set without the first two columns (id and class), previously transformed into a *numpy* array, and the desired method among the four contemplated: k-means and four agglomerative variants (complete, average, single and ward).

Agglomerative clustering methods are based on starting from a number of clusters equal to the number of features. Later, by computing the pairwise distances, an aggregation process is performed and clusters are combined. Four distance metrics were considered, as stated in the Scikit Sklearn documentation:

- Complete: considers the maximum of the distances between the observations in the two groups.
- Average: considers the average of the distances between the observations in the two groups.
- Single: considers the minimum of the distances between the observations in the two groups.
- Ward: minimizes the variance of the two clusters.

K-means starts by selecting random centroids and computing the distance from the observations to those centroids, creating groups with the nearest points. Secondly, the algorithm computes the average of the observations in the same group, yielding a new centroid. Hence, by iterating these two steps, the clusters are defined.

The resulting values of the normalized mutual information metric without preprocessing operations are not good. The provided data set contains a large amount of features,

therefore, it seems reasonable that some of them could be providing similar information and, hence, may not be essential. In statistical terms, this refers to the correlation between features. The correlation matrix yields the result of the pairwise correlation between features. In the designed program, a function called `filter_data` was defined. The mentioned function receives the initial data set and a coefficient. It computes the correlation matrix and removes from those features having a high pairwise correlation, one of them. The threshold for removing the features is the coefficient `corr_threshold`. A coefficient of 0.8, which is acceptable (it represents highly correlated variables), gives the best result.

To summarize, the algorithm for getting satisfactory results measured with the normalized mutual information metric is based on removing those features that have high correlation and choosing the optimal number of clusters.

2.2 Second data set: msdata.csv

The data set is composed by 5000 features and 695 items. The same procedure for clusterization with an optimal number of clusters that was used in the previous data set was implemented in this program. Hence, there is a for loop that iterates the number of clusters and stores that one providing the best result for the normalized mutual information metric. The same five clusterization methods were performed.

Without preprocessing the data set, the obtained results after performing clusterization are not good, according to the suggested metric. Hence, principal component analysis was carried out. Principal component analysis is a dimensionality reduction method based on choosing those directions in which the variation of the data set values is larger. In the given case, to optimize the clusterization technique, two principal components are considered. With the data frame composed by those two principal components, the results are more appropriate.

3 Results

3.1 First data set: genedata.csv

Initially, the clusterization methods were performed on the data set, without any preprocessing operations. Since the results are not as good as desired, the data is previously filtered based on the correlation values, yielding better values for the normalized mutual metric. The results for each of the clustering methods and using the processed and non-processed data are shown in 1. The best result is obtained making use of hierachical agglomerative method based on *ward* (explained in the previous section). The normalized mutual information reaches a value of 0.9307 and the optimal number of clusters is 6. In the appendix, the figures showing the metric as a function of the number of clusters for each of the considered methods are shown. In figures 1 to 5 the results for the non-filtered data are presented. In the following figures, those for the preprocessed data set. Finally, in figure 10, the best result is introduced.

Method	raw data		filtered data	
	K	NMI	K	NMI
kmeans	6	0.8923	6	0.9076
agg complete	12	0.7162	11	0.6724
agg average	15	0.6258	18	0.3899
agg single	19	0.0582	19	0.0582
agg ward	7	0.8821	6	**0.9307**

Table 1: NMI results for the different methods with the optimal number of clusters K in the first data set.

3.2 Second data set: msdata.csv

As in the previous case, without preprocessing operations, the results are not good. In the figures 11 to 15 , the normalized mutual information was plotted as a function of the number of clusters. As it can be depicted, the values for the metric are extremely low. Hence, principal component analysis was performed on the data set, yielding the analogous results shown in figures 16 to 20. Particularly, in figure 20, the best result is shown. It was obtained by performing principal component analysis with two principal components on the data set and clustering the resulting reduced data set using a hierarchical agglomerative method based on *ward*, which was previously explained. The obtained value for the normalized mutual information was 0.9037, with a number of clusters equal to 5. The results for both cases, with and without preprocessing phase are presented in the table 2.

Method	raw data		filtered data	
	K	NMI	K	NMI
kmeans	18	0.1598	5	0.8026
agg complete	9	0.2784	9	0.7755
agg avg	19	0.0565	13	0.7853
agg single	19	0.0565	19	0.0634
agg ward	5	0.4739	5	**0.9037**

Table 2: NMI results for the different methods with the optimal number of clusters K in the second data set.

4 How to run the program

To run the programs, execute the following: `python clustering_genedata.py` for the first data set and `python clustering_msdata.py` for the second.

In both programs there are some lines that can be commented or uncommented in order to run them with or without filtering. The programs will run with filtering operations by default. Due to stochasticity components, it might be necessary to execute the programs several times to obtain the presented results. The program for the first data set takes some time to run (over 5 minutes).

5 Appendix

5.1 Plots for first data set (genedata.csv) without filtering

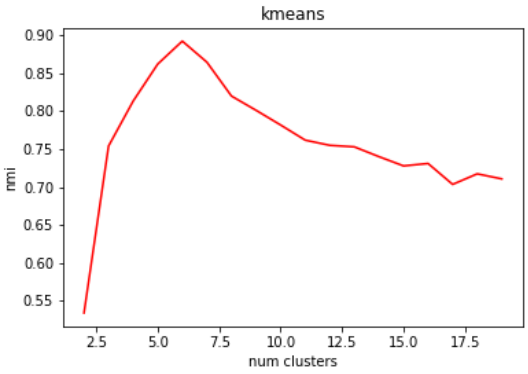

Figure 1: NMI in function of the number of cluster for kmeans method.

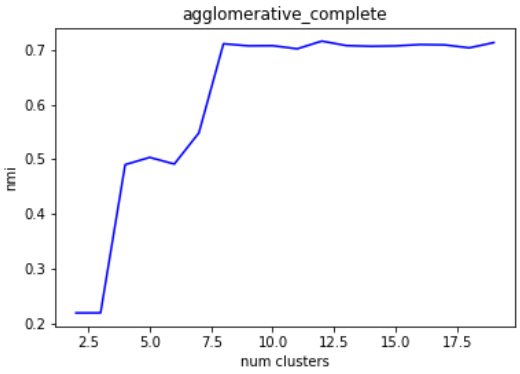

Figure 2: NMI in function of the number of cluster for agglomerative complete method.

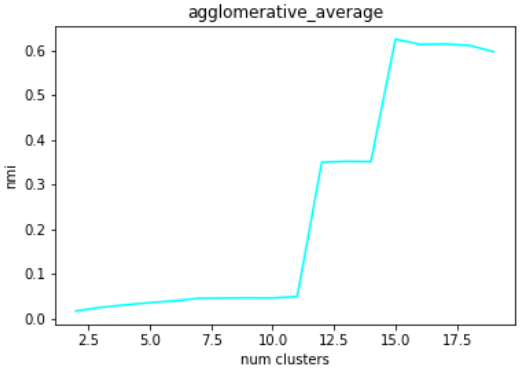

Figure 3: NMI in function of the number of cluster for agglomerative average method.

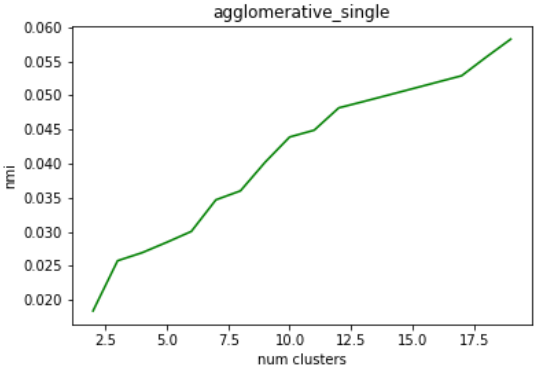

Figure 4: NMI in function of the number of cluster for agglomerative single method.

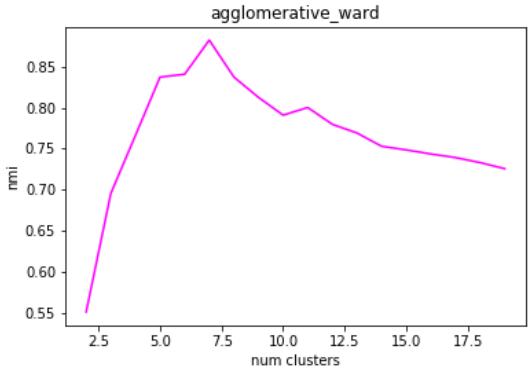

Figure 5: NMI in function of the number of cluster for agglomerative ward method.

5.2 Plots for first data set (genedata.csv) with filtering

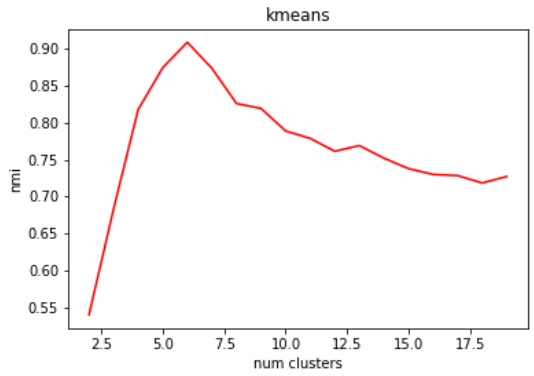

Figure 6: NMI in function of the number of cluster for kmeans method with the filtered data.

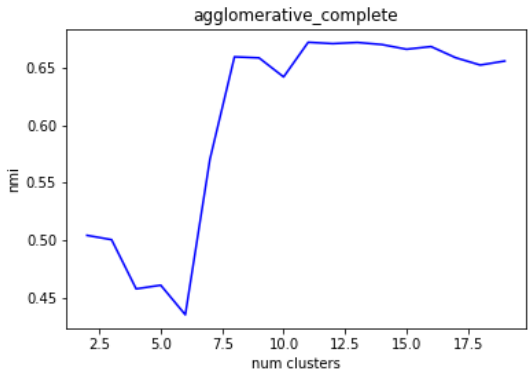

Figure 7: NMI in function of the number of cluster for agglomerative complete method with the filtered data.

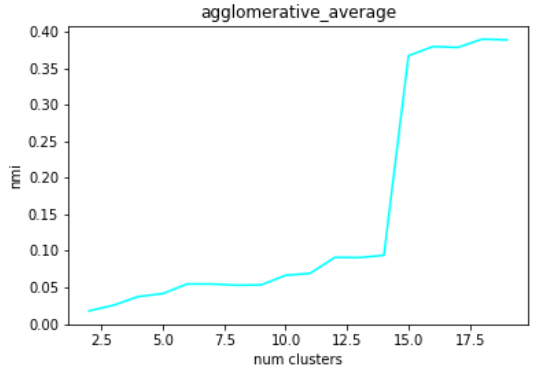

Figure 8: NMI in function of the number of cluster for agglomerative average method with the filtered data.

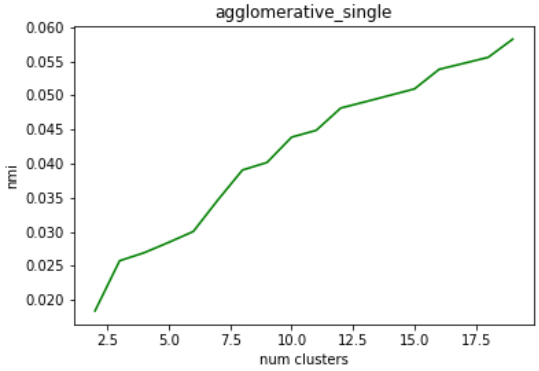

Figure 9: NMI in function of the number of cluster for agglomerative single method with the filtered data.

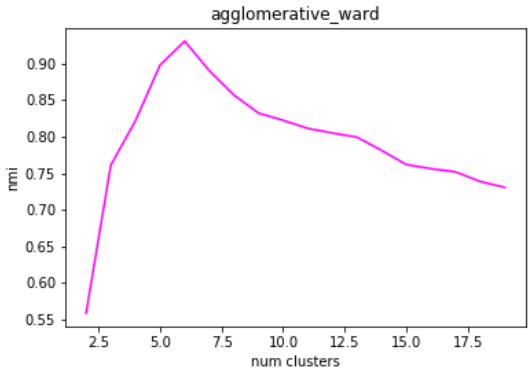

Figure 10: NMI in function of the number of cluster for agglomerative ward method with the filtered data.

5.3 Plots for second data set (msdata.csv) without PCA

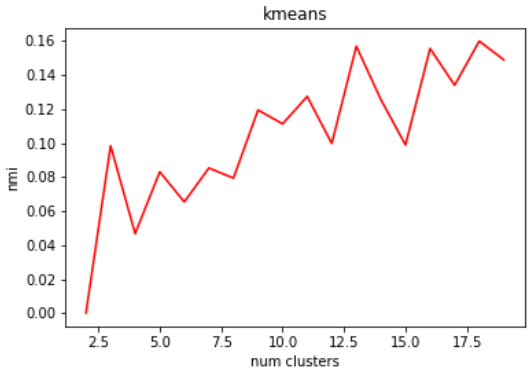

Figure 11: NMI in function of the number of cluster for kmeans method.

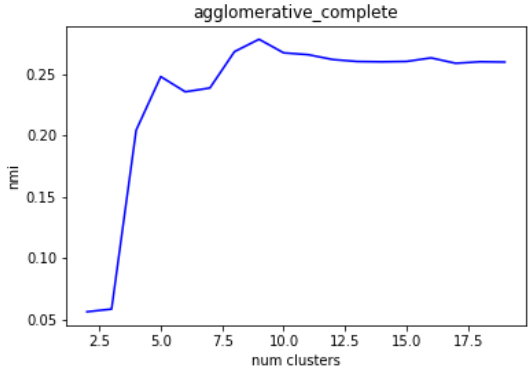

Figure 12: NMI in function of the number of cluster for agglomerative complete method.

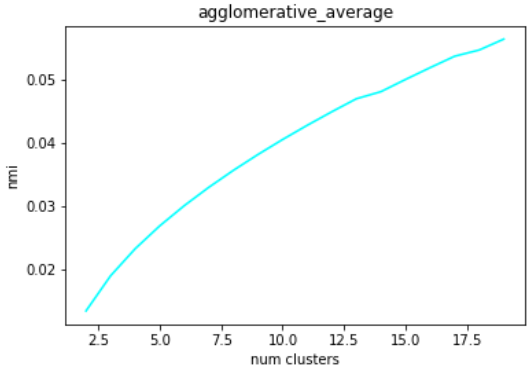

Figure 13: NMI in function of the number of cluster for agglomerative average method.

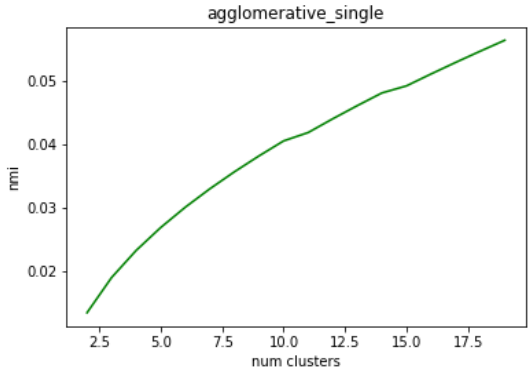

Figure 14: NMI in function of the number of cluster for agglomerative single method.

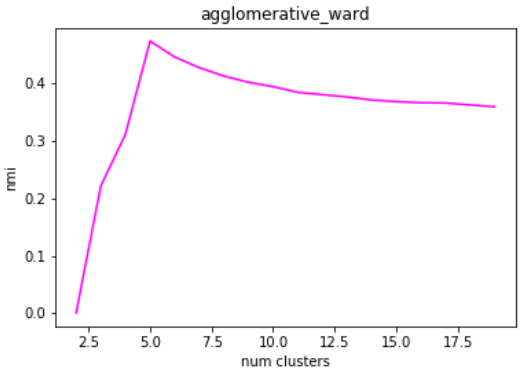

Figure 15: NMI in function of the number of cluster for agglomerative ward method.

5.4 Plots for second data set (msdata.csv) with PCA

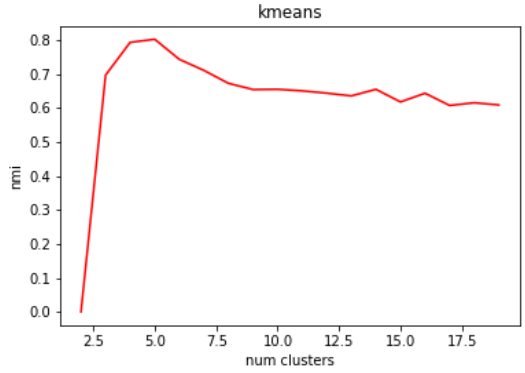

Figure 16: NMI in function of the number of cluster for kmeans method with the filtered data.

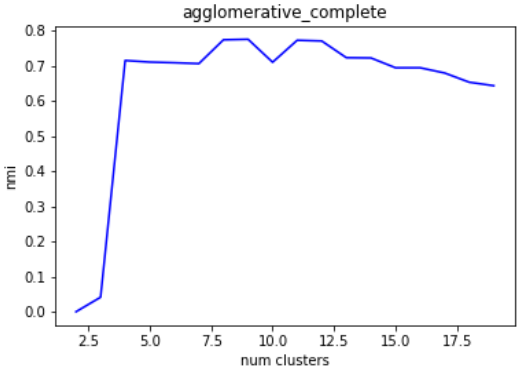

Figure 17: NMI in function of the number of cluster for agglomerative complete method with the filtered data.

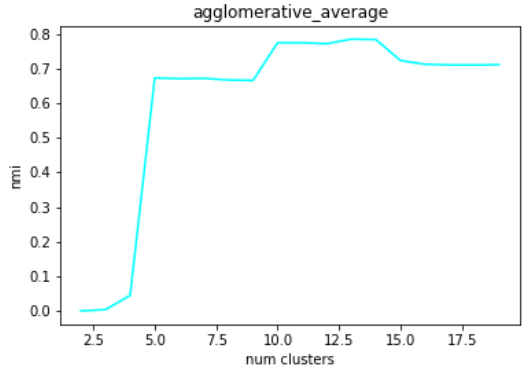

Figure 18: NMI in function of the number of cluster for agglomerative average method with the filtered data.

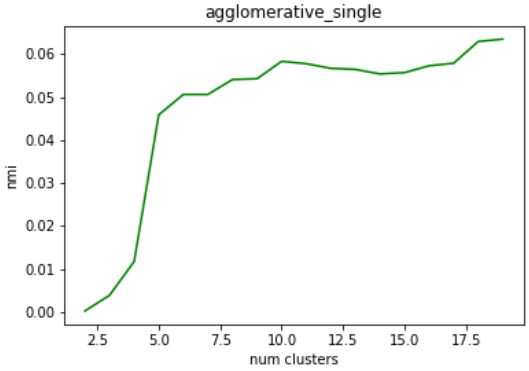

Figure 19: NMI in function of the number of cluster for agglomerative single method with the filtered data.

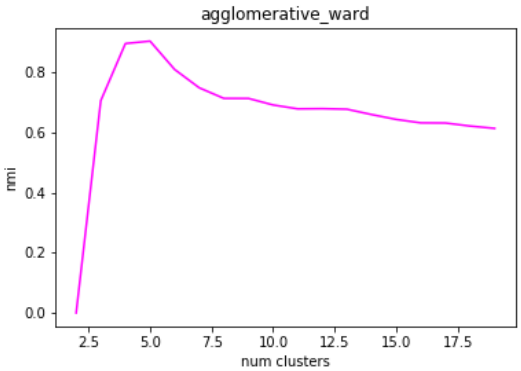

Figure 20: NMI in function of the number of cluster for agglomerative ward method with the filtered data.