INFMDI348

Project on Data Mining

Clustering

Professor: Mauro Sozio

Tiago CHEDRAUOI SILVA

June 22, 2012

Contents

List of Figures

1 Introduction

The main goal of this project is to clustering a collection of documents so that documents dealing with a same topic belong to a same cluster. To this purpose, we chose to use the k-means algorithm.

2 Pre-processing

As input for the clustering we have 1000 documents that were taken from blogs. Some steps to remove noisy are applied: 1. Remove non-alphabetic characters 2. Remove white spaces 3. Make every word lower case 4. Remove stopwords 5. Remove Words with frequency lower than 5 times and higher than 1000 (ex: object, message, from, etc.), because they are considered noises as they cannot give a meaningful value to our clustering. 6. Normalize the matrix: That gives the same importance to all documents: either a big or a small document will have the same influence during a cluterization.

After the step 5 on the pre-processing, we make a histogram of the words, which shows that the major part of words have a low frequency, and some words have a high frequency. The important words are those which appears in a great part of documents with high frequency, which could induce us to a cluster. For example, if my files talk about sports and are recent maybe the word Eurocup and Olympic games have a high frequency and appears in most part of documents with sports subject.

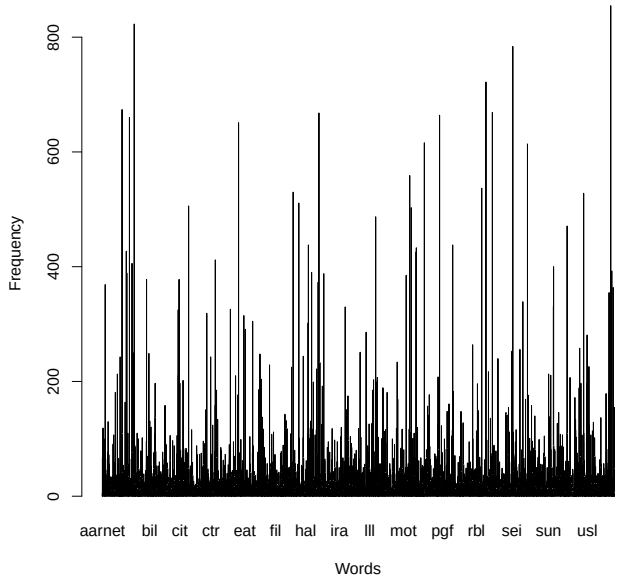


Figure 1: Histogram word frequency in a collection of documents

3 Clustering

An algorithm that clusters a given points is the k-means algorithm, which needs as input a number of clusters and the data. The steps are: find to each point the nearest center, so that my point belongs to the cluster with that center. After we recalculate the centers and restart. We stop if the recalculates centers are unchanged.

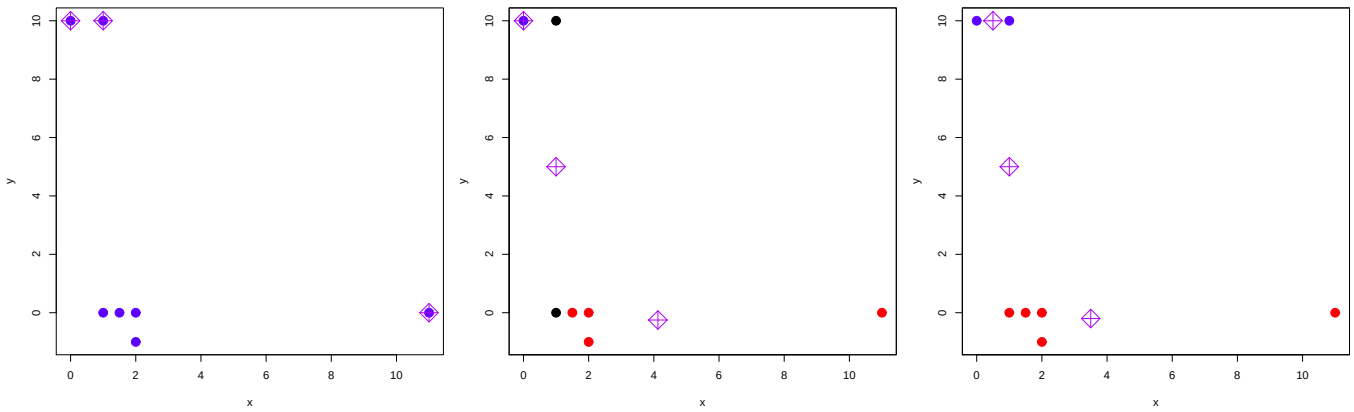
So, we executed k-means (k = 3) on our corpus 20 times, each one with different initial centers. The best result is showed in table 1:

Table 1: Clustering Results

K-means (k = 3)	
Size Group 1	389
Size Group 2	224
Size Group 3	508
SSE	999.9103

Unfortunately, the k-means algorithm can produce an empty cluster. To show an example of empty cluster, we have chosen some points and centers so that in an iteration, a recalculated center is more far from one point in its cluster than the center of another cluster. If all points of a cluster are closer to other cluster centers, then the next iteration the cluster will be emptied.

The figure 2(a) shows the initial points and centers. After the first interaction, we have 3 cluster (blue: 1 point, black: 2 point, red: 4 point) showed in the figure 2(b). Finally the figure 2(c) shows that after the recalculation of the centers, we got an empty cluster (blue: 2 point, black: empty, red: 5 points)



(a) Initial Points (b) Iteration 1 (c) Iteration 2

Figure 2: Kmeans: Example of empty cluster

3.1 Clustering validation

A great number of studies were made to validate a clustering such as Gap statistics, silhouette validation technique and the Hartigan calculus. These algorithms give us the best number of cluster to the data based on the solution for different number of clusters.

The average silhouette width could be applied for evaluation of clustering validity and also could be used to decide how good is the number of selected clusters. To construct the silhouettes S(i) the following formula is used:

$$S_i = \frac{b_i - a_i}{max(a_i, b_i)} \tag{1}$$

Where a_i is average distance of the sample to all other samples in the same cluster and b_i is the lowest average distance to the other clusters, that means, we get all points from other clusters and we make get the average distance to our sample point. As each different cluster give us a value, we get the lowest one.

So the idea is to maximize the distance between different clusters and minimize the distance of the points in the same cluster.

As a result we get a $-1 < s_i < 1$, if s_i is near 1 we have a well clustered point, if it is near -1 we have as misclassification, if it is near 0 the sample could be assigned to other cluster. We search for the maximum value o the average s_i which means that the points were more near to a well clustering than lower values.

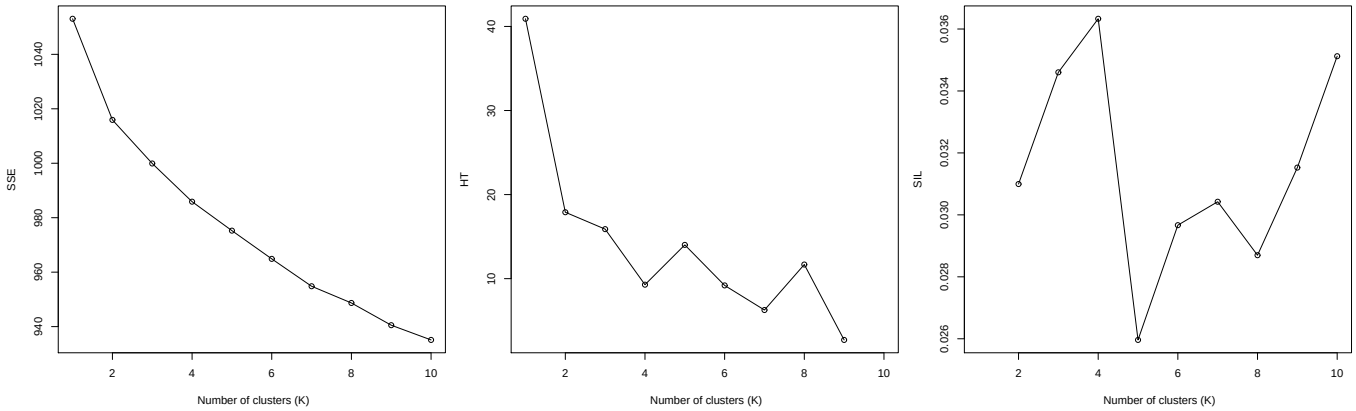
So, using the method of average silhouette coefficient which combines both cohesion and separation we got the graph 3(c), which gives $k_{good} = 4$.

Also Hartigan in 1975 used the SSE value as a value for comparison. Firstly, as the SSE value always decrease with the increase of k, Hartigan tried to give a penalty value for increasing the k, also he would compared the value of SSE of k and k+1, which give us the Hartigan's method equation:

$$H(k) = \left(\frac{W(k)}{W(k+1)} - 1 \right) * (n - k - 1)$$

where n is the number of instances being clustered and k is the number of sets. Hartigan suggests that if H(k) > 10 the cluster should be added, if H(k)<10, the cluster is not added and the algorithm is stop. Applying Hartigan, we got the graph 3(b) which gives $k_{good} = 4$

Finally, as both methods, which are a reference in the field of study, gave us $k_{good} = 4$, we will use this value for the next experiments. However, that doesn't mean that it is the real best number of clusters.



(a) SSE vs k (b) Hartigan Calculus (c) Silhouette Coefficient

Figure 3: Kmeans validation

3.2 Clustering Analyses

Table 2: Results k-means

	Time	SSE	Distrib				Purity				Entropy			
Fixed	0.028	987.42	367	381	218	155	0.59	1	0.94	0.98	1.34	0	-0.30	-0.99
Rand (nstart = 1)	0.044	988.28	301	222	434	164	0.98	1	0.74	1	0.11	0	1.08	0
Rand (nstart = 5)	0.273	986.35	383	219	422	97	1	0.94	0.65	0.98	0	0.30	1.23	0.08
Rand (nstart = 10)	0.323	986.35	383	97	422	219	1	0.99	0.65	0.94	0	0.08	1.23	0.30
Rand (nstart = 20)	0.57	985.88	77	384	219	441	0.99	1	0.94	0.66	0.1	0	0.3	1.2

4 References

Hartigan, J. (1975) Clustering Algorithms New York: Wiley
http://www.stanford.edu/ hastie/Papers/gap.pdf