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 and higher than 1000 times (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 clustering.

After the step 5 on the pre-processing, we make a histogram of the words, which shows that the major part of words has a low frequency, and some words have a high frequency. The important words are those which appear 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.

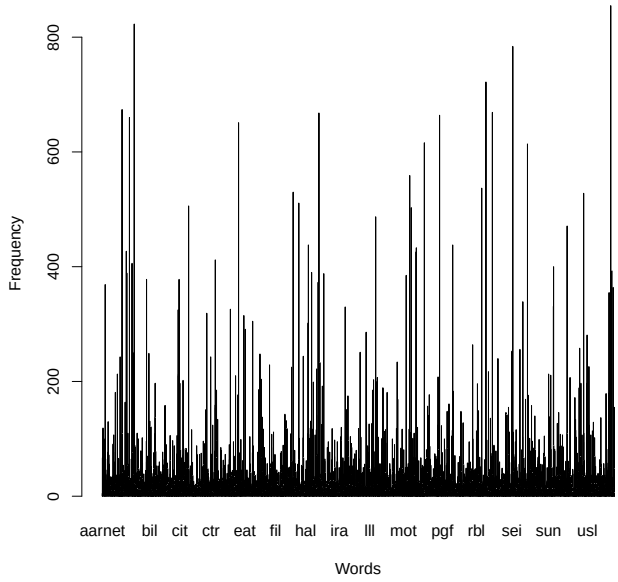


Figure 1: Histogram word frequency in a collection of documents

3 Clustering

An algorithm that clusters 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 the point belongs to the cluster with that center. After we recalculate the centers and restart. We stop if the recalculates centers are unchanged.

Thus, we executed k-means (k = 3) on our corpus 3 times, each one with different initial centers. After we used the parameter n from the bult-in function k-means which receives the number of different inicial centers, we used the value of ten different starts, and as a result the k-means give the best solution found. The results are showed in table 1:

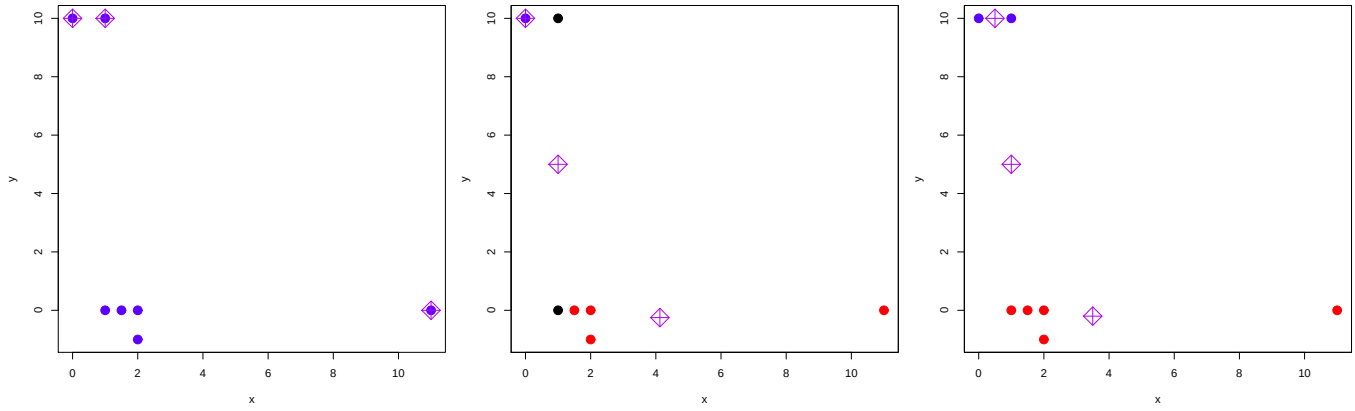
Table 1: Clustering Results

K-means (k = 3) (n = number of different starts)				
	SSE	Distribution		
Try 1 (n = 1)	1000.39	389	303	429
Try 2 (n = 1)	1000.63	610	388	123
Try 3 (n = 1)	1005.97	126	386	609
Try 4 (n = 10)	999.91	508	224	389
Try 5 (n = 10)	999.91	508	389	224

The different values achivied by k-means shows that the inital centers have a important role on the result, because for each chosen centers we got a different clustering. Also, if we execute the algorithm with $n = 10$ we arrive to stable result each time we execute kmeans for $k = 3$, each means that we should use a great number of starts to get this stability of results.

Unfortunately, the k-means algorithm can produce an empty cluster. To show an example of empty cluster, we chose some points and centers so that in some 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 the three initial 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 [Robert Tibshirani, 2000], silhouette validation technique [ROUSSEEUW, 1986] and the Hartigan calculus[Hartigan, 1975]. 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 gives 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 of the average s_i which means that the points were more near to a well clustering than lower values.

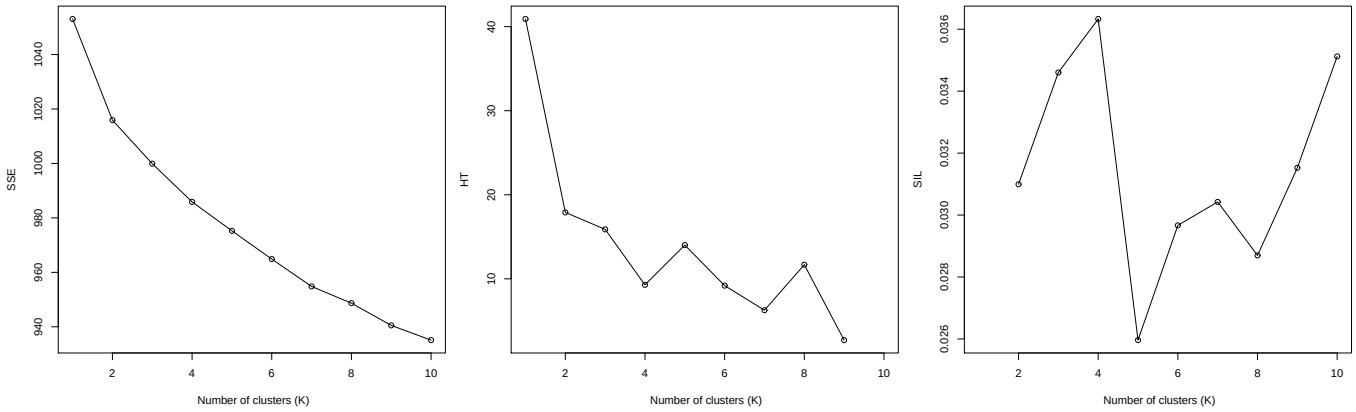
So, using the method of average silhouette coefficient which combines both cohesion and separation we got the graph 3(c), which gives $k_{good} = 4$.

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, moreover he 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 samples being clustered and k is the number of clusters. Hartigan suggests that if H(k) > 10 the cluster should be added, if H(k)$<$10, the cluster is not added and the evaluation ends. Applying Hartigan, we got the graph 3(b) which gives $k_{good} = 4$

Finally, as both methods, which are a reference in this 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

For the analyses of a clustering we fixed $k = 4$ and developed the functions that calculates the purity and entropy of each cluster. As k is fixed, the parameter that has a great influence in the k-means algorithm is the initial centers. The table below shows the result when the centers are fixed and when we increase the times of the random starts. This shows that greater is the number of starts the probability of a better result is increased.

For the choice of fixed centers, we divided our initial data in the 4 groups, and found the words that have 90% of presence in each group if compared with all corpuses. After, we searched for the documents that had the biggest number of words that would describe a group of clusters. We got a better result than a random start with one center. However, for five different starts we found a better result.

3.3 Improvements

As improvements to our algorithms we implemented the Tf-idf weight[Wikipédia, 2012], which gives a higher weight to rarely terms than frequently ones, that is made by multipling the frequency of the word by the idf weight showed below:

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2}$$

Table 2: Results k-means: k = 4 (n = number of different starts)

	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	0.1
Random (n = 1)	0.044	988.05	162	222	80	657	1	1	0.98	0.5	0	0	0.1	1.42
Random (n = 5)	0.273	986.35	383	219	422	97	1	0.94	0.65	0.98	0	0.30	1.23	0.08
Random (n = 10)	0.323	986.35	383	97	422	219	1	0.99	0.65	0.94	0	0.08	1.23	0.30
Random (n = 20)	0.57	985.88	77	384	219	441	0.99	1	0.94	0.66	0.1	0	0.3	1.2

where $|D|$ is the total number of documents in the corpus and $|\{d \in D : t \in d\}|$ is number of documents where the term t appears.

However, the result did not improve significantly (see table 3), this may have happened because the results are already good and improving them significantly is not very easy.

Table 3: Results k-means: k = 4 Tf-idf (n = number of different starts)

	Time	SSE	Distrib				Purity				Entropy			
Random (n = 10)	0.398	24372	444	218	384	75	0.67	0.95	1	0.97	1.2	0.3	0	0.2

3.4 Conclusion

The most important thing is the relation between the size of the cluster and its purity, so to evaluate our different approaches a weighted average using the size of the clusters and their purity should be done. The table below shows that the centers have a great influence in the k-means algorithm, as the normalization of data. Using the Tf-Idf approach we got a better result, however, not so far from the result using only a normalized approach.

Table 4: Results k-means: k = 4 (n = number of different starts)

	Weighted average purity
Fixed	0.851
Random (n = 1)	0.705
Random (n = 5)	0.855
Random (n = 10)	0.855
Random (n = 20)	0.854
TF-IDF Random (n = 10)	0.857

References

[Hartigan, 1975] Hartigan, J. (1975). Clustering algorithms. New York: Wiley.

[Robert Tibshirani, 2000] Robert Tibshirani, Guenther Walther, T. H. (2000). Estimating the number of clusters in a data via the gap statistic. Available at http://www.stanford.edu/ hastie/Papers/gap.pdf.

[ROUSSEEUW, 1986] ROUSSEEUW, P. J. (1986). Sillhouettes: a graphical aid to the interpretation and validation of cluster analysis. Available at ftp://adrem.ua.ac.be/pub/preprints/87/Silgra87.pdf.

[Wikipédia, 2012] Wikipédia (2012). Tf*idf. Available at http://en.wikipedia.org/wiki/Tf*idf.