K-Means Clustering

Juan David Rodríguez Cuervo

March 2018

The **K-Means** algorithm is a method to train an artificial unsupervised model on a dataset. The basic concept behind it, is to cluster or group the points in the given space of the input in a certain number of clusters.

Each point is said to be classified in a category, or belonging to a certain cluster, is the euclidean distance from that certain point to the centroid of the cluster is the minimum compared with the distance to the other centroids. That said, it can be seen that a cluster is basically defined by a centroid, and a belonging relationship is given by the distances with respect to each cluster.

In other words, let p be a certain point in an $f-$dimensional dataset D, and predefined cluster centroids $C = \{c_1, c_2, ..., c_k\}$. As it can be seen, the vector space of the input data depends on the number of features, f, there are to consider. Each cluster, or category T is related with its centroid $T(c_i)$. Then, it is said that for every $j \neq i \in [1, k]$:

$$p \in T(c_i) \leftrightarrow d(p, c_i) < d(p, c_j) \tag{1}$$

Here, $d(p, q)$ is the distance function. In the euclidean space, it is the difference function defined as:

$$d(p, q) = \sqrt{\sum_{n=1}^{f} (p_n - q_n)^2} \tag{2}$$

Moreover, each cluster's size (at least during the learning process) is determined by the number of data points that belong to the category.

$$T(c_i) = \{p \in D : d(p, c_i) < d(p, c_j), \forall j \neq i \in [1, k]\} \tag{3}$$

On each learning iteration, the definition of the new locations for the centroids, depend on the data points in the given category. That is to say, the new value for each centroid is the arithmetic mean of the points classified in the category defined by the centroid in question. Hence,

$$c_i' = mean(T(c_i)) = \frac{\sum_{n=1} T_n}{|T(c_i)|} \tag{4}$$

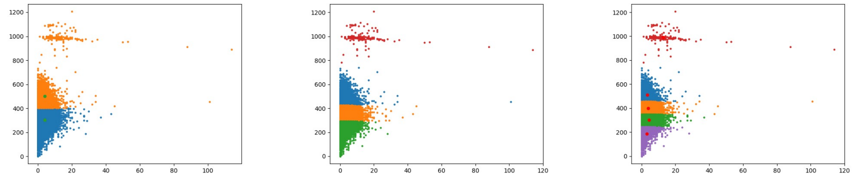
This would be a learning iteration, composed by a classification (or grouping) step, and then a centroid definition, based on the mean point of each cluster.

This process is repeated until some criterion is met. In the present implementation this is the convergence criterion. A convergence state is reached when the centroids position between one iteration and the next is 0. In general, a convergence state is reached if:

$$d(c_i, c_i') = 0 \tag{5}$$

1 Results

The algorithm was executed on a sample dataset, included in the project files. At first, the dataset was unclassified. Thus, the code was run with different number of target clusters. This is shown in Figure 1, where each color represent a cluster in the dataset.



(a) k=2

(b) k=4

(c) k=5

Figure 1: K-Means with different numbers of clusters

Up to this point, the number of clusters is arbitrary. Given this, it becomes relevant to introduce *The Elbow Method* as the way to determine analytically the number of clusters that may best describe the dataset.

The particular objective is to determine the least value that is sufficient to describe the set. That is, to find the optimal number of clusters to find. This is because when selected a very high number of clusters, the model would tend to overfit the training set, but having very few clusters, would imply that each cluster has a very high data dispersion. The optimal value would be one high enough to ensure each cluster is ”compact”, but low enough to avoid overfitting.

The Elbow Method is one way to determine this optimal value. It is a graphical method, which relies in the plotting of the total sums of each point with respect to its nearest centroid. All these distances are summed up, giving the total sum of distances for a given k value of clusters. As one would expect, for very high values of k, the sum of distances become very small, whilst if selected 1 cluster, for example, a very high value would be given, as the dispersion is the highest and that centroid would not describe correctly the dataset. Fig 2 shows the plots of the total sum of distances for each k value.

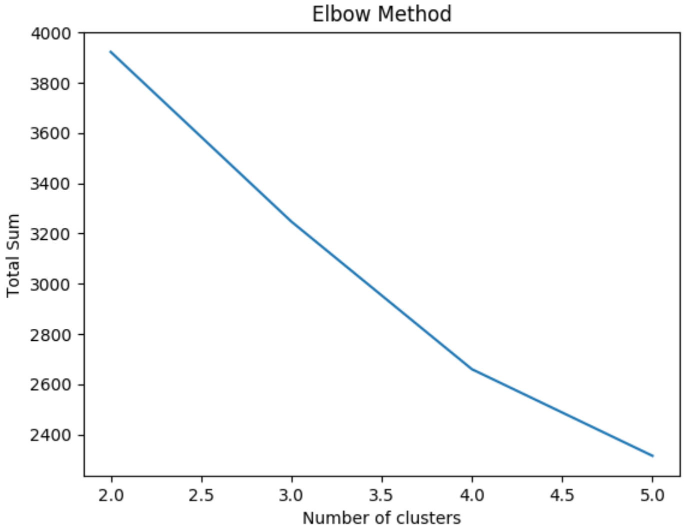


Figure 2: Elbow Method graph

As it can be shown in the graph, there is a "change direction" point in the plot (see plot for k=4). From that point comes the method's name. Therefore, that direction change states the optimal value for k, as it avoids the overfitting (higher values change the distance sum value in a despicable rate), but also keeping the clusters dispersion values at its lowest.

That said, it can be determined that $k = 4$ is the optimal cluster number, as it keeps a balance in compactness maximization (for each cluster) and overfitting minimization.