K-means Clustering -Example Notes (Practice, Tutorial)

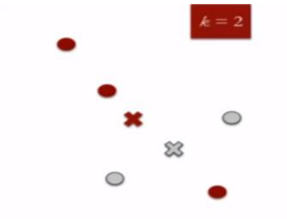
K Means Clustering

K means is an iterative clustering algorithm that aims to find local maxima in each iteration. This algorithm works in these 5 steps:

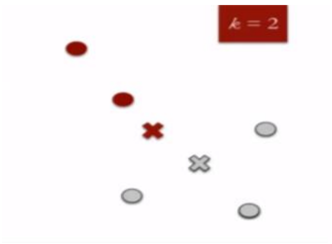
1. **Specify the desired number of clusters K**: Let us choose k=2 for these 5 data points in 2-D space.

2. **Randomly assign each data point to a cluster**: Let's assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.

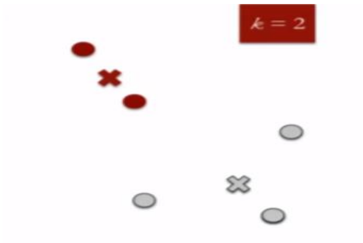
3. **Compute cluster centroids**: The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.



4. **Re-assign each point to the closest cluster centroid**: Note that only the data point at the bottom is assigned to the red cluster even though it's closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster.



5. **Re-compute cluster centroids**: Now, re-computing the centroids for both the clusters.



Repeat steps 4 and 5 until no improvements are possible

Similarly, we'll repeat the 4th and 5th steps until there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

Determining the Number of Clusters (K-means)

Elbow method

The Elbow method looks at the total within-cluster sum of square (WCSS) as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't improve the total WCSS. The optimal number of clusters can be defined as follow:

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
2. For each k, calculate the total within-cluster sum of square (wss).
3. Plot the curve of wss according to the number of clusters k.
4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

Average silhouette method

Briefly, it measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering. The algorithm is similar to the elbow method and can be computed as follow:

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
2. For each k, calculate the average silhouette of observations (*avg.sil*).
3. Plot the curve of *avg.sil* according to the number of clusters k.
4. The location of the maximum is considered as the appropriate number of clusters.

Gap statistic method

The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be value that maximize the gap statistic (i.e that yields the largest gap statistic). This means that the clustering structure is far away from the random uniform distribution of points. The algorithm works as follow:

1. Cluster the observed data, varying the number of clusters from k = 1, …, k_{max}, and compute the corresponding total within intra-cluster variation W_k.
2. Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters k = 1, …, k_{max}, and compute the corresponding total within intra-cluster variation W_{kb}.
3. Compute the estimated gap statistic
4. Compute also the standard deviation of the statistics

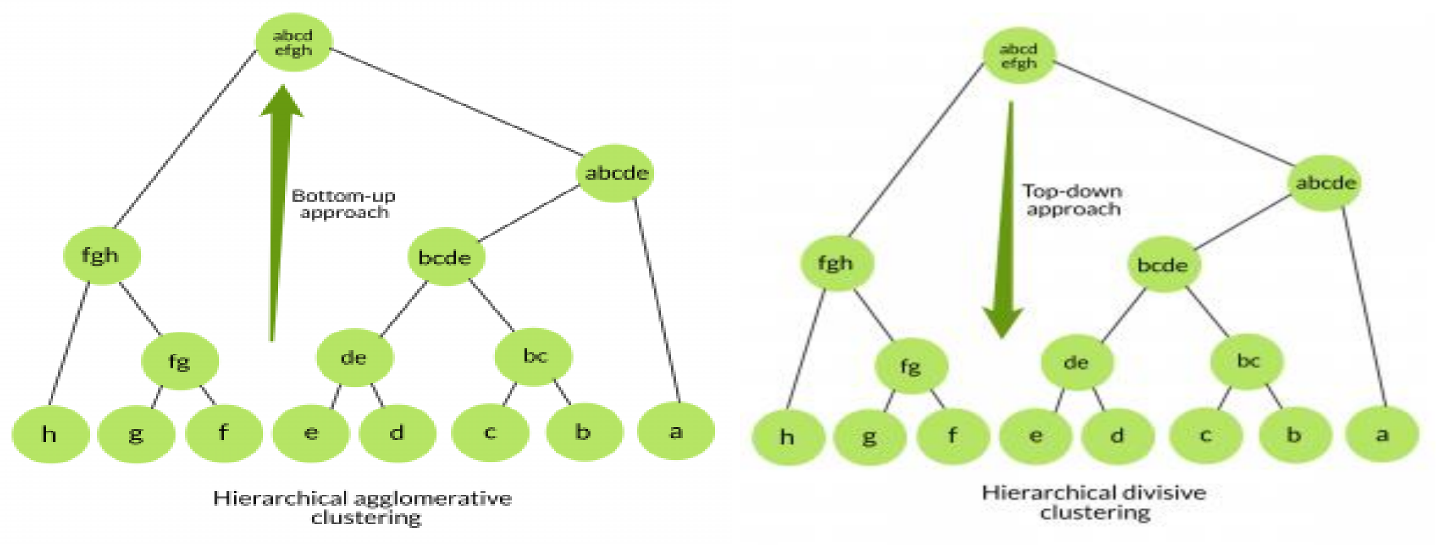
5. Choose the number of clusters as the smallest value of k

Hierarchical Clustering

Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. **Basically, there are two types of hierarchical cluster analysis strategies:**

1. **Agglomerative Clustering:** Also known as bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

2. **Divisive clustering:** Also known as top-down approach. This algorithm also does not require to prespecify the number of clusters. Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been splitted into singleton cluster.



Source: geeksforgeeks.org

Difference between K Means and Hierarchical clustering

- Hierarchical clustering can't handle big data well but K Means clustering can.
- In K Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.

- K Means is found to work well when the shape of the clusters is hyper spherical (like circle in 2D, sphere in 3D).
- K Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into. But, you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram

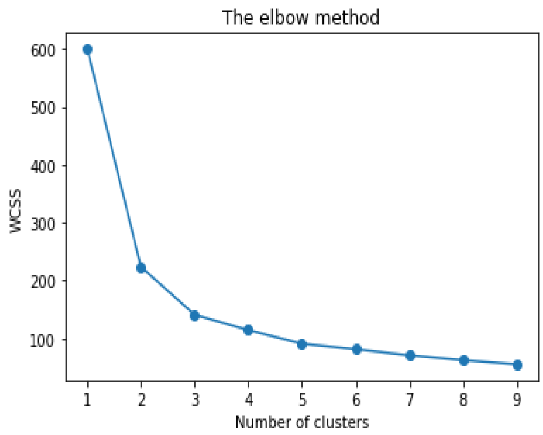
Python Practice

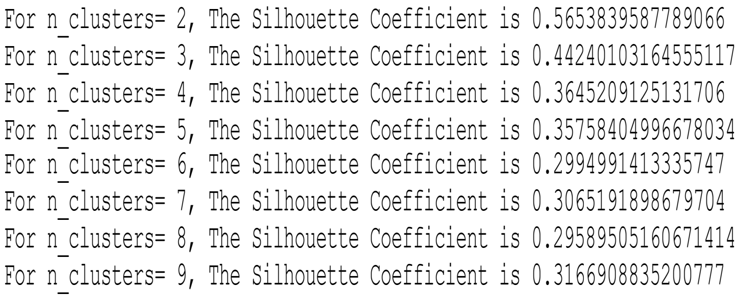
1. You will be using well known **iris.csv** dataset

   Please check the KM-Example file and practice accordingly

Evaluating the Model

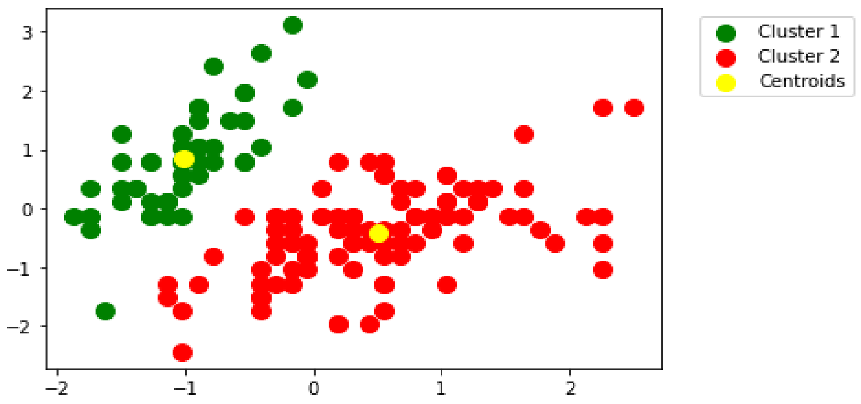
Elbow Method and Silhouette Method





```
For n_clusters= 2, The Silhouette Coefficient is 0.5653839587789066
For n_clusters= 3, The Silhouette Coefficient is 0.44240103164555117
For n_clusters= 4, The Silhouette Coefficient is 0.3645209125131706
For n_clusters= 5, The Silhouette Coefficient is 0.35758404996678034
For n_clusters= 6, The Silhouette Coefficient is 0.2994991413335747
For n_clusters= 7, The Silhouette Coefficient is 0.3065191898679704
For n_clusters= 8, The Silhouette Coefficient is 0.29589505160671414
For n_clusters= 9, The Silhouette Coefficient is 0.3166908835200777
```

Looking at both methods we observe that **n_clusters should be 2**. Below is a graphical representation of the clusters.



Python Tutorial

1. You will be using **bodyfatanalysis.csv**
   Follow the "Tutorial-KMeans-BodyFatAnalysis.html" file

Let's look at the variables in the dataset

Independent Variables:

Triceps: circumference (cm)

Thigh: circumference (cm)

Midarm: circumference (cm)

Dependent Variable:

Bodyfat: % of Body Fat