UNIVERSITY OF LIVERPOOL

COMP 527 Data Mining and Visualisation

Assignment 1 Report

Submitted To: Dr Viktor Zamaraev, Joshua Murphy, Mehdi Mohamed Pierre Anhichem

Submitted By: Balkrishna Bhatt (201673048)

Abstract:

In this report, we describe several clustering algorithms like k-means, k-means++ & Bisecting k-Means that have been explained with pseudo code and implemented using Python as a programming language and how they have been applied to cluster a dataset that we have given.

Q1:

Ans: K-means clustering is a popular unsupervised learning algorithm used for partitioning a dataset into k clusters based on the similarity of their data points. The algorithm aims to minimize the distance between the data points and the centroids (mean) of their corresponding clusters. [1] The algorithm is iterative and involves two main steps, i.e., assigning the data points to their nearest centroid (assignment phase) and then computing the mean of all data points assigned to each centroid to obtain new centroids (optimization phase). The process is repeated until the centroids no longer change significantly, or the maximum number of iterations is reached [2].

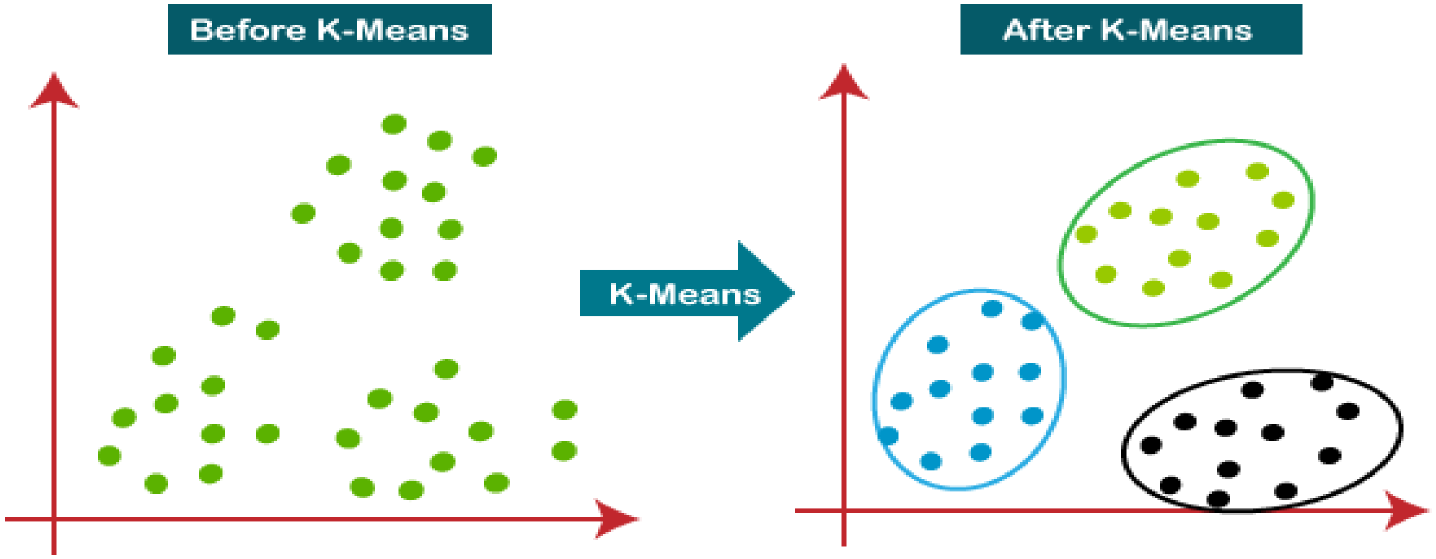

Figure 1.1: K-means Algorithm

Pseudo-Code:

```
k-MeansClustering (Number of clusters k, Dataset {x1, x2, ..., xn})

1. Initialisation phase:
  a. Choose k random centroids from the dataset as initial cluster representatives.
  b. Initialize an empty dictionary to store the clusters and their assigned data points.
2. Assignment phase:
  a. For each data point xi, compute the squared Euclidean distance to each centroid yj.
  b. Assign xi to the cluster with the closest centroid, i.e., argmin($||xi - yj||^2$)
  c. Update the dictionary of clusters with the new assignments.
```

3. Optimisation phase:
   a. For each cluster, compute the mean of its assigned data points.
   b. Set the new centroid of the cluster as the mean.
   c. Repeat steps 2 and 3 until convergence (i.e., the centroids no longer change significantly, or the maximum number of iterations is reached)
4. Return the dictionary of clusters and their assigned data points

Q2:

Ans: K-means is a widely used clustering algorithm that partitions a given dataset into k clusters based on the similarity of the data points. However, the standard k-means algorithm can suffer from the initialization problem where the choice of initial cluster centres can significantly impact the resulting clusters. K-means++ is an improved version of the standard k-means algorithm that is addressing this initialization problem. It aims to maximize the quality of the clusters produced by a k-means clustering algorithm by optimizing the initialization of the centroids in order to achieve a more accurate clustering process. The K-means++ algorithm selects the centroids in a way that ensures a more even distribution across the dataset instead of randomly selecting the centroids from the dataset [3].

The k-means++ algorithm consists of three main steps:

1. **Initialization phase:** Choose the first cluster centre Y1 uniformly at random from the dataset D.
2. **Iterative phase:** For every i=2,...,k, select the next cluster centre Yi based on the probability distribution defined as:

   Yi = argmax X∈$\mathcal{D}$ min(j<i){distance $(\mathrm{X}, \mathrm{Yj})^2$}/ $\sum_{\mathrm{X}\in\mathcal{D}} \min(\mathrm{j} < \mathrm{i})$\{distance$(\mathrm{X}, \mathrm{Yj})^2$\}

   where distance(X, Yj) is the Euclidean distance between X and the j-th cluster centre.

   This selection process ensures that the next cluster centre is far from the previous ones and is likely to represent a new cluster.
3. **Assignment and optimization phase:** Proceed with the standard k-means using Y1,...,Yk as the initial cluster centres. Assign each data point to the closest cluster centre and compute the new cluster centres as the mean of the assigned data points. Repeat the assignment and update steps until convergence. [4]

Pseudo-Code:

```
k_means_pp(D, k):
  # Initialization phase
  Y = [random.choice(D)]

  # Iterative phase
  for i in range(2, k+1):
    dist = [min([distance(X, Yj)**2 for Yj in Y]) for X in D]
    probs = dist / np.sum(dist)
    Y.append(D[np.random.choice(len(D), p=probs)])
```

```
 # Assignment and optimization phase
 kmeans = KMeans(n_clusters=k, init=np.array(Y), n_init=1)
 kmeans.fit(D)

 return kmeans.labels_, kmeans.cluster_centers_
```

Q3:

Ans: Bisecting k-means is a hybrid approach between Divisive Hierarchical Clustering (top-down clustering) and K-means Clustering. Instead of partitioning the data set into K clusters in each iteration, bisecting k-means algorithm splits one cluster into two sub clusters at each bisecting step (by using k-means) until k clusters are obtained. [5]

- Overall, the Bisecting k-Means algorithm divides the data into a hierarchical structure of clusters, with the number of clusters decreasing at each level.
- At each level, the algorithm selects the largest cluster, bisects it, and adds the resulting clusters to the hierarchy.
- This process continues until the desired number of clusters is reached.

Pseudo-Code:

```
Input: dataset D, number of clusters s
1. Initialise a tree T with a single vertex containing all points in D
2. While the number of leaf clusters in T is less than s:
   a. Select a leaf node L in T that has the largest sum of square distance $\sum_{(x,y)\in L} dist(x,y)^2$
   b. Split L into two clusters L1 and L2 using the k-means algorithm.
   c. Add L1 and L2 as children of L in T
3. Return the leaf clusters of T
```

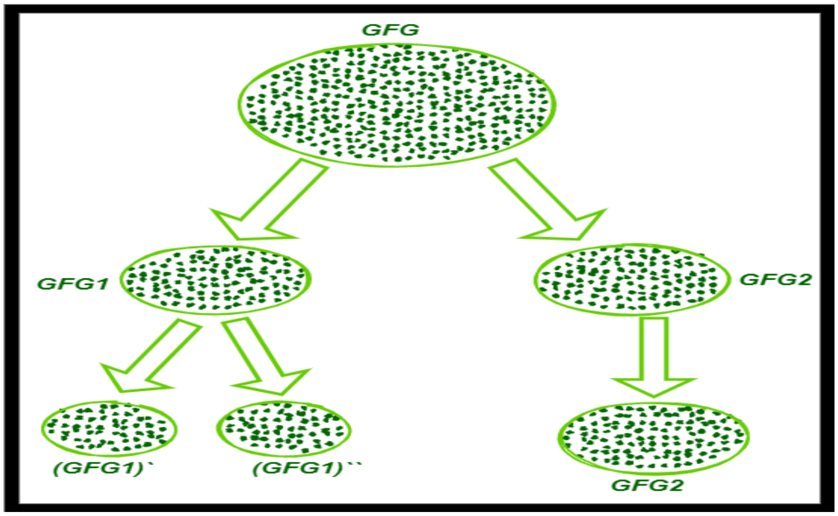

Figure 4.1: Bisecting K-means Algorithm

Q7:

Ans: When clustering a huge word embedding dataset with varying values of k, it is important to consider the quality of the clusters produced by the different algorithms. In this case, we can compare the k-means, k-means++, and bisecting k-means hierarchical algorithms based on their performance in terms of the Silhouette coefficient.

Based on empirical studies, k-means++ is generally considered to be superior to regular k-means in terms of both convergence speed and quality of the solution. Bisecting k-means hierarchical can produce more accurate clusters than k-means, but it can be more computationally expensive.

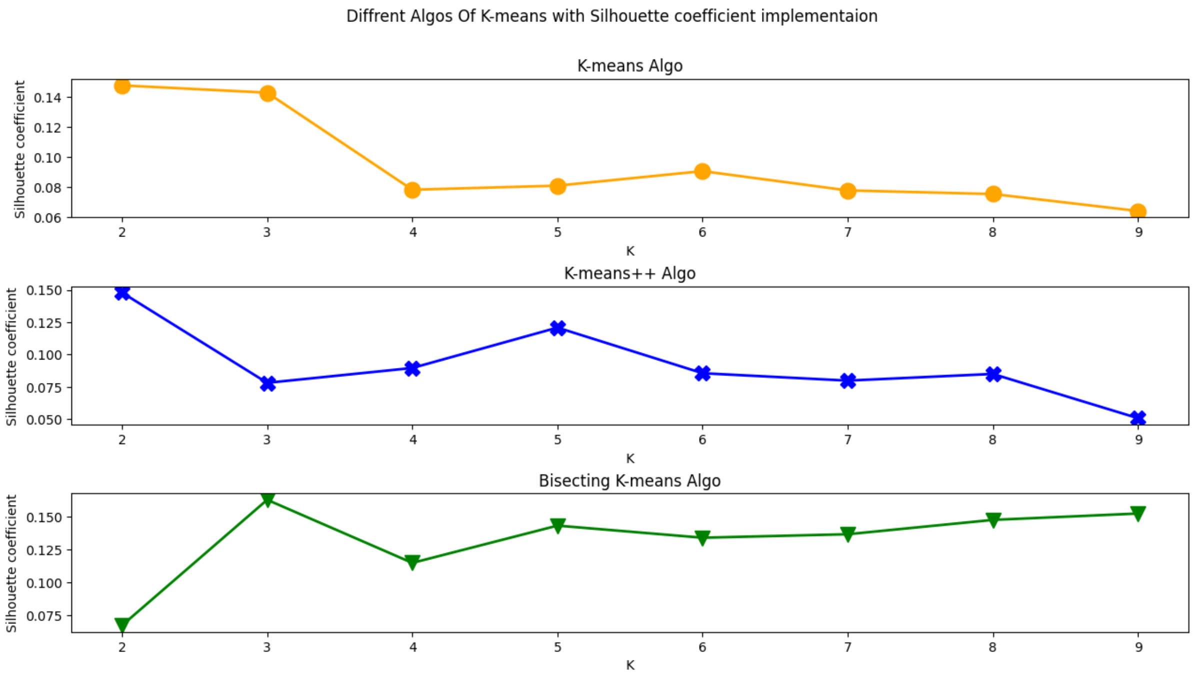

Figure 7.1: All K-means Algorithm Plotting

In order to compare the different clustering obtained in (4)-(6), we can analyse the Silhouette coefficients for each value of k or s. For the k-means algorithm, the Silhouette coefficients increase as k increases, but after k=3, the increase becomes less pronounced. The highest Silhouette coefficient is obtained when k=3, which suggests that three clusters are the best for this dataset using k-means.

For the k-means++ algorithm, the Silhouette coefficients decrease as k increases, The highest Silhouette coefficient is obtained when k=5, which suggests that five clusters are the best for this dataset using k-means++.

For the Bisecting k-Means algorithm, the Silhouette coefficients decrease as s increases. The highest Silhouette coefficient is obtained when s=3, which suggests that three clusters are the best for the top level of the hierarchy. However, when we refine the clustering to obtain more clusters, the Silhouette coefficients decrease, which suggests that the Bisecting k-Means algorithm may not be the best option for this dataset.

Based on the analysis of Silhouette coefficients, we can conclude that the k-means++ algorithm provides the best clustering for this dataset, with four clusters being the optimal number of clusters.

References

[1] "sklearn.cluster.KMeans," Scikit Learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Berkeley Symp. on Math. Statist. and Prob,* vol. I, pp. 281-297 , 1967.

[3] S. Khosla, "ML | K-means++ Algorithm," geeksforgeeks, 13 July 2021. [Online]. Available: https://www.geeksforgeeks.org/ml-k-means-algorithm/.

[4] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms,* p. 1027–1035, 2007.

[5] A. Firdaus, "Bisecting Kmeans Clustering," Medium, 17 March 2020. [Online]. Available: https://medium.com/@afrizalfir/bisecting-kmeans-clustering-5bc17603b8a2.