Student ID – 201669264
Student Name – Rahul Nawale

Data Mining and Visualization - CA2 – Data Clustering

Question 1: Answer

K-means algorithm:

The k-means clustering algorithm is an unsupervised machine learning algorithm used for clustering data points into K number of clusters based on their features. The algorithm works as follows:

The algorithm works in three phases: initialization, assignment, and optimization.

1. In the initialization phase, K random data points are chosen as initial centroids, and a distance metric is defined.

2. In the assignment phase, each data point is assigned to the nearest centroid, forming K clusters.

3. In the optimization phase, the mean of each cluster is computed to obtain new centroids, and the assignment phase is repeated until convergence is achieved.

4. Convergence is achieved when the centroids no longer change or a maximum number of iterations is reached.

Pseudo Code of K-means algorithm:

kMeansClustering (Number of clusters: k, Dataset: {X1,.....Xn})

1. Randomly initialize k cluster representatives Y1,.....Yk from the dataset

2. Repeat below steps until convergence or maximum iterations reached:

   a. For each data point Xi, calculate its distance to each representative Yj, using Euclidean distance

   b. Assign each data point Xi to the nearest representative Yj, forming clusters C1,.....Ck

   c. For each cluster Cj, calculate the mean of all data points assigned to that cluster, set the new representative Yj to the mean

3. Output - clusters C1,.....Ck and their representatives Y1,.....Yk

Distance function (Euclidean distance):

distance(Xi, Yj) = sqrt(sum((Xi - Yj)^2))

Note: The convergence criterion can be either "no objects have moved among clusters" or "maximum number of iterations reached", as specified by the user.

Question 2: Answer

K-means ++ algorithm:

K-means++ is an improved version of the standard k-means clustering algorithm which focuses on improving the initial centroid selection process. K-means++ addresses the issue of inaccurate clustering results or slow convergence that can arise from random centroid selection by using a more intelligent approach to select the initial centroids.

The algorithm chooses the first centroid randomly from the dataset and then selects subsequent centroids based on the nearest distance to the previously selected centroids with probability proportional to its squared distance. By choosing centroids that are far apart from each other in the data space, k-means++ can lead to better clustering results and faster convergence.

The k-means++ algorithm works as follows:

1. Choose the first centroid randomly from the data points.
2. For each data point, calculate its distance to the nearest centroid that has already been chosen.
3. Choose the next centroid randomly from the remaining data points, with probability proportional to its distance to the nearest centroid.
4. Repeat step 2 and 3 until k centroids have been chosen.
5. Use the k centroids obtained in step 4 as the initial centroids for the standard k-means algorithm.

Overall, k-means++ is a simple and effective algorithm that can significantly improve the performance of k-means clustering, especially when the number of clusters is large or the data has complex structures.

KMeans++ Pseudo Code:

kMeansPlusPlusClustering (Number of clusters: k, Dataset: {X1,.....Xn})

1. Randomly choose the first representative Y1 to a randomly chosen data point $\overline{X}$from the dataset {X1,.....Xn}

2. For j = 2 to k:

a. For each data point Xi, compute its distance to the nearest representative that has already been chosen

b. Choose the next representative, Yj, randomly from the remaining data points, with probability Yj = $\overline{x}$ proportional to the distance from each point to its nearest representative $\frac{R\bar{x}^2}{\Sigma x \in Dataset\ R\bar{x}^2}$

3. Run the standard k-means algorithm with the initial representatives Y1,.....Yk

4. Output the resulting clusters C1,.....Ck and their representatives Y1,.....Yk

Standard k-means algorithm:

- Same as in the standard k-means algorithm, using the k initial representative chosen by k-means++

Note: Step 2b involves selecting the next representative with probability proportional to the squared distance from each point to its nearest center

Question 3: Answer

Bisecting K-means:

Bisecting k-Means is a hierarchical clustering algorithm that works by repeatedly dividing clusters into two until the desired number of clusters is obtained. The algorithm starts with all data points in a single cluster, and then iteratively splits the largest cluster into two using the standard k-means algorithm.

Bisecting KMeans Pseudo code:

bisectingKMeansClustering (Number of clusters: k, Dataset: {X1,.....Xn})

1. Start with all data points in a single cluster C1

2. While the number of clusters is less than k:

   a. Choose the largest cluster Cj i.e. Cluster with largest sum of square distance $\sum_{\bar{x},\, \bar{y} \in Cj} \text{dist}(\bar{x},\bar{y})^2$

   b. Apply the standard k-means algorithm to Cj to obtain two new clusters Cj1 and Cj2

   c. Replace Cj with Cj1 and Cj2

3. Output the resulting clusters C1,.....Ck

Standard k-means algorithm:

1. Randomly initialize k cluster representatives Y1,.....Yk from the dataset

2. Repeat until convergence or maximum iterations reached:

   a. For each data point Xi, calculate its distance to each representative Yj, using Euclidean distance

   b. Assign each data point Xi to the nearest representative Yj, forming clusters C1,.....Ck

   c. For each cluster Cj, calculate the mean of all data points assigned to that cluster, set the new representative Yj to the mean

3. Output the resulting clusters C1,.....Ck and their representatives Y1,.....Yk

Question 7: Answer

For Kmeans, the highest Silhouette coefficient is obtained for **K=2 (0.1473)**, which means the algorithm is able to separate the data points into two distinct clusters more effectively than for any other values of K.

For Kmeans++, the highest Silhouette coefficient is obtained for **K=2 (0.1478)**, which is the same as Kmeans, indicating that the initialization technique has not made a significant difference in this case.

For Bisecting Kmeans, the highest Silhouette coefficient is obtained for **K=2 (0.6902**), which is significantly higher than for any other values of K. However, as we increase K, the quality of clustering deteriorates significantly, as indicated by negative Silhouette coefficients for K>3.

Based on the silhouette coefficients, the best clustering for this dataset is obtained using KMeans with k=2. Both KMeans and KMeans++ give similar silhouette coefficients for k=2, but KMeans++ performs worse than KMeans for higher values of k.

Bisecting KMeans performs better than KMeans and KMeans++ for k=2, but it produces worse clustering results for higher values of k.

Overall, it seems that K=2 is the optimal number of clusters for all the three algorithms. However, it is worth noting that the Bisecting Kmeans algorithm has achieved significantly higher Silhouette coefficient compared to Kmeans and Kmeans++ for K=2, indicating that Bisecting Kmeans algorithm is more effective in clustering the given data.