ASSIGNMENT 10

PROBLEM STATEMENT

Perform the data clustering algorithm using any Clustering algorithm.

OBJECTIVE

Students should be able understand the concept of data clustering and different data clustering algorithm.

THEORY

Data clustering

- Cluster analysis or simply clustering is the process of partitioning a set of data objects (or observations) into subsets.
- Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters.
- The set of clusters resulting from a cluster analysis can be referred to as a clustering. In this context, different clustering methods may generate different clustering on the same data set. The partitioning is not performed by humans, but by the clustering algorithm.
- Cluster analysis has been widely used in many applications such as image pattern recognition, Web search, biology, and security.

Partitioning methods

- Given a set of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and $k \leq n$.
- That is, it divides the data into k groups such that each group must contain at least one object. In other words, partitioning methods conduct one-level partitioning on data sets. The basic partitioning methods typically adopt exclusive cluster separation. That is, each object must belong to exactly one group.
- Most partitioning methods are distance-based. Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are “close” or related to each other, whereas objects in different clusters are “far apart” or very different.
- Most applications adopt popular heuristic methods, such as greedy approaches like the k-means and the k-medoids algorithms, which progressively improve the clustering quality and approach a local optimum.

Hierarchical methods

- A hierarchical method creates a hierarchical decomposition of the given set of data objects.
- A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.
- The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group.
- It successively merges the objects or groups close to one another, until all the groups are merged into one (the topmost level of the hierarchy), or a termination condition holds.
- The divisive approach, also called the top-down approach, starts with all the objects in the same cluster.
- In each successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition holds. Hierarchical clustering methods can be distance-based or density- and continuity based.

Method	General Characteristics
Partitioning methods	– Find mutually exclusive clusters of spherical shape – Distance-based – May use mean or medoid (etc.) to represent cluster center – Effective for small- to medium-size data sets
Hierarchical methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object "linkages"

CONCLUSION

In this way we have explored the concept of data clustering and implemented the data clustering algorithm.

ORAL QUESTION

1. What is the primary objective of clustering algorithms?
2. What are some common types of clustering algorithms?
3. Describe the process of preparing data for clustering.
4. Can you provide an example of a real-world application where clustering algorithms are commonly used?

```
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt


# Load the iris dataset
iris = load_iris()


# Features
X = iris.data


# Initialize KMeans with the number of clusters (3 for the iris dataset)
kmeans = KMeans(n_clusters=3)


# Fit KMeans to the data
kmeans.fit(X)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
  warnings.warn(
```

KMeans

KMeans(n_clusters=3)

```
# Get the cluster centroids and labels
centroids = kmeans.cluster_centers_
labels = kmeans.labels_


# Visualizing the clusters (assuming 2D data for simplicity)
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
plt.scatter(centroids[:, 0], centroids[:, 1], marker='X', s=200, c='red')
plt.title('K-Means Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```