Clustering: K-Means and Hierarchical Clustering

Clustering is an unsupervised machine learning technique used to group similar data points into clusters. Unlike supervised learning, clustering does not require labeled data. The primary goal of clustering is to identify hidden patterns, structures, or relationships within a dataset.

Data points within the same cluster are more similar to each other than to data points in other clusters. Clustering is widely used in data mining, pattern recognition, image processing, market segmentation, recommendation systems, and bioinformatics.

Some popular clustering algorithms include K-Means Clustering, Hierarchical Clustering, DBSCAN, and Mean Shift Clustering. Among these, K-Means and Hierarchical Clustering are the most commonly used techniques.

K-Means Clustering

Definition

K-Means Clustering is a partition-based clustering algorithm that divides a dataset into K predefined clusters. The algorithm groups similar data points by minimizing the distance between data points and the cluster centroid.

The value of K represents the number of clusters to be formed and is specified before the algorithm begins.

Working of K-Means Clustering

The K-Means algorithm follows these steps:

Step 1: Choose Number of Clusters (K)

Select the desired number of clusters.

Step 2: Initialize Centroids

Randomly select K points as initial cluster centroids.

Step 3: Assign Data Points

Each data point is assigned to the nearest centroid based on distance measures such as Euclidean distance.

Step 4: Recalculate Centroids

The centroid of each cluster is recalculated by taking the mean of all data points within that cluster.

Step 5: Repeat

Steps 3 and 4 are repeated until cluster assignments no longer change or maximum iterations are reached.

Mathematical Formula

The centroid of a cluster is calculated as:

[Centroid = \frac{\sum_{i=1}^{n}x_i}{n}]

Where:

- (x_i) = Data points in the cluster
- (n) = Number of data points

The objective function minimized by K-Means is:

[J=\sum_{i=1}^{K}\sum_{x \in C_i}|x-\mu_i|^2]

Where:

- (C_i) = Cluster i
- (\mu_i) = Centroid of cluster i

Example of K-Means Clustering

Suppose a company wants to group customers based on purchasing behavior.

1. Select K = 3 clusters.
2. Choose three initial centroids.
3. Assign customers to the nearest centroid.
4. Recalculate cluster centers.
5. Repeat until stable clusters are formed.

The result is three groups of customers with similar purchasing patterns.

Advantages of K-Means

1. Easy to understand and implement.
2. Fast and computationally efficient.
3. Suitable for large datasets.
4. Produces well-defined clusters.
5. Scalable for big data applications.

Disadvantages of K-Means

1. Number of clusters (K) must be specified beforehand.
2. Sensitive to initial centroid selection.
3. Affected by outliers and noise.
4. Works best with spherical clusters.
5. May converge to local optima.

Applications of K-Means

- Customer segmentation
- Image compression
- Recommendation systems
- Document clustering
- Market analysis
- Pattern recognition

Hierarchical Clustering

Definition

Hierarchical Clustering is a clustering technique that builds a hierarchy of clusters. Unlike K-Means, it does not require the number of clusters to be specified initially.

The algorithm creates a tree-like structure called a Dendrogram, which represents the relationships among clusters.

Hierarchical clustering is widely used when the natural grouping of data is unknown.

Types of Hierarchical Clustering

1. Agglomerative Hierarchical Clustering

Also known as the bottom-up approach.

- Each data point starts as an individual cluster.
- The closest clusters are merged repeatedly.
- The process continues until a single cluster remains.

2. Divisive Hierarchical Clustering

Also known as the top-down approach.

- All data points begin in one cluster.
- The cluster is repeatedly divided into smaller clusters.
- The process continues until each data point forms its own cluster.

Agglomerative clustering is more commonly used than divisive clustering.

Working of Agglomerative Hierarchical Clustering

Step 1

Treat each data point as a separate cluster.

Step 2

Calculate distances between all clusters.

Step 3

Merge the two nearest clusters.

Step 4

Update the distance matrix.

Step 5

Repeat until all points belong to a single cluster.

Distance Measures

Hierarchical clustering uses different methods to calculate distances between clusters.

Single Linkage

Distance between the nearest points of two clusters.

Complete Linkage

Distance between the farthest points of two clusters.

Average Linkage

Average distance between all pairs of points.

Ward's Method

Minimizes variance within clusters.

Dendrogram

A dendrogram is a tree-like diagram used to visualize hierarchical clustering.

Features of a dendrogram:

- Displays cluster formation process.
- Helps determine the optimal number of clusters.
- Represents similarities among data points.

The height of branches indicates the distance at which clusters are merged.

Example of Hierarchical Clustering

Suppose a university wants to group students based on academic performance.

1. Each student starts as an individual cluster.
2. Similar students are merged.
3. Larger groups are formed progressively.
4. A dendrogram is generated.
5. Clusters are selected by cutting the dendrogram at a suitable level.

Advantages of Hierarchical Clustering

1. No need to specify the number of clusters beforehand.
2. Produces a clear visual representation through dendrograms.
3. Suitable for small and medium-sized datasets.
4. Captures nested cluster structures.
5. Flexible with different distance metrics.

Disadvantages of Hierarchical Clustering

1. Computationally expensive for large datasets.
2. Sensitive to noise and outliers.
3. Once clusters are merged, they cannot be separated.
4. Requires more memory compared to K-Means.

Applications of Hierarchical Clustering

- Gene sequence analysis
- Social network analysis
- Document classification
- Customer segmentation
- Image processing
- Biological taxonomy

Comparison Between K-Means and Hierarchical Clustering

Feature	**K-Means Clustering**	**Hierarchical Clustering**
Learning Type	Unsupervised	Unsupervised
Number of Clusters	Must be specified	Not required initially
Speed	Faster	Slower

Feature	K-Means Clustering	Hierarchical Clustering
Scalability	Suitable for large datasets	Suitable for small datasets
Output	Cluster assignments	Dendrogram
Memory Usage	Lower	Higher
Flexibility	Less flexible	More flexible