Unit-5: Clustering

1. Clustering Basics

- **Classification vs Clustering**
  - *Classification* → Supervised learning (uses labeled data, e.g., Apple/Banana).
  - *Clustering* → Unsupervised learning (no labels; group data by similarity).
- **Cluster**: A group of data objects that are **similar within group** and **dissimilar across groups**.
- Clustering = **finding hidden patterns** and groups in data.
- Used also for **Outlier detection** (objects that don't fit into any cluster).

---

2. Applications of Clustering

- **Marketing** → Group customers by buying habits.
- **Biology** → Classify plants & animals.
- **Insurance** → Identify policyholders with high claim cost.
- **City Planning** → Group houses by type, value, location.
- **Libraries** → Group books by topics.
- **Earthquake studies** → Identify risky areas.
- **Fraud detection** → Detect abnormal activities.

---

3. Good Clustering Algorithm

- Produces **high-quality clusters**:
  - **High intra-class similarity** (members inside a cluster are close).
  - **Low inter-class similarity** (clusters are well separated).

---

4. Requirements for Cluster Analysis

1. **Scalability** – Should handle large datasets.
2. **Different types of attributes** – Numeric, categorical, binary, ordinal.
3. **Arbitrary shapes** – Not only circular clusters (must detect any shape).
4. **Input parameters** – Like number of clusters (k); sometimes difficult to choose.

5. **Deal with noisy data/outliers** – Should be robust.
6. **Incremental clustering** – Should handle new data without re-computing all.
7. **High-dimensional data** – e.g., documents with thousands of keywords.
8. **Constraint-based clustering** – E.g., ATM locations must consider geography.
9. **Interpretability** – Results should be easy to understand.

---

5. Basic Clustering Methods

(A) Partitioning Methods

- Divide data into **k clusters**.
- Each object belongs to **exactly one cluster**.
- Example algorithms: **k-Means, k-Medoids**.
- Works well for small–medium datasets.

(B) Hierarchical Methods

- Creates a **tree-like hierarchy of clusters**.
- **Agglomerative (Bottom-up)**: Start with single objects → merge step by step.
- **Divisive (Top-down)**: Start with all objects together → split step by step.
- Result shown in **Dendrogram** (tree diagram).

(C) Density-Based Methods

- Form clusters as **dense regions** separated by low-density areas.
- Can find clusters of **arbitrary shape** and handle **noise**.
- Example: **DBSCAN, OPTICS, DENCLUE**.

(D) Grid-Based Methods

- Divide data space into a **grid of cells**.
- All clustering done on grid, not data directly → **fast**.
- Time depends only on number of cells, not number of data points.

---

6. Partitioning Algorithms

(a) k-Means (Centroid-based)

- Select **k centroids randomly**.

- Assign each point to nearest centroid.
- Update centroids = mean of cluster points.
- Repeat until clusters don't change.
- **Weakness** → Sensitive to outliers.

(b) k-Medoids (Representative object-based)

- Similar to k-Means, but uses **medoid (most central object)** instead of mean.
- Algorithm: **PAM (Partitioning Around Medoids)**.
- More robust to outliers than k-Means.

---

7. Hierarchical Clustering

- **Agglomerative (AGNES)**: Merge closest clusters step by step.
- **Divisive (DIANA)**: Split clusters step by step.
- **Distance measures**:
  - *Single link* → min distance between two clusters.
  - *Complete link* → max distance.
  - *Average link* → average distance.
  - *Centroid link* → distance between centroids.
- **Weakness** → Cannot undo steps, $O(n^2)$ time.
- **Improved methods**:
  - **BIRCH** → Uses CF-tree, good for large numeric data.
  - **CHAMELEON** → Uses dynamic modeling (interconnectivity + proximity).

---

8. Density-Based Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- Parameters:
  - **Eps** → neighborhood radius.
  - **MinPts** → minimum points in neighborhood.
- Concepts:
  - **Core point** → has at least MinPts within Eps.

- Density-reachable → can be reached via chain of core points.
- Density-connected → both reachable from some other point.

- Handles noise and arbitrary shapes.

OPTICS (Ordering Points To Identify the Clustering Structure)

- Improvement over DBSCAN.
- Produces **cluster ordering** instead of single clustering.
- Less sensitive to parameter choice.
- Concepts: **Core-distance** and **Reachability-distance**.

---

9. Outliers

- **Definition** → Data object that deviates a lot from others, generated by different mechanism.
- Different from **noise** (random error).
- **Applications**: Fraud detection, medical analysis, customer segmentation.

Types of Outliers

1. **Global (Point anomaly)** → Entirely different from rest.
2. **Contextual (Conditional anomaly)** → Abnormal only in certain context (e.g., 28°C in winter).
3. **Collective outliers** → Group of data objects deviating together.

---

10. Outlier Detection Methods

- **Supervised** → Use labeled data, train classifier. (Problem: rare outliers).
- **Unsupervised** → Assume normal data form clusters, outliers don't.
- **Semi-supervised** → Limited labeled data + unlabeled data.

Categories:

- **Statistical methods** → Assume normal data follows distribution (e.g., Gaussian).
- **Proximity-based methods** → Outliers are far from neighbors.
- **Clustering-based methods** → Outliers don't belong to any cluster or form very small clusters.

---

11. Important Exam Questions

- What is clustering? Difference with classification.
- Supervised vs Unsupervised learning.
- Requirements for cluster analysis.
- Explain clustering methods: Partitioning, Hierarchical, Density-based, Grid-based.
- k-Means and k-Medoids with examples.
- Agglomerative vs Divisive clustering.
- Explain BIRCH and CHAMELEON.
- DBSCAN algorithm.
- OPTICS in detail.
- Outliers: definition, types, methods of detection.