FADML Scribe
Types of learning techniques and clusterting

Soumya Bhattacharya
24CS92P02
Indian Institute Of Technology, Kharagpur

April 15, 2025

1 Types of learning techniques

A concise tabular explanation of the types of learning that are present:

Learning Type	Definition	Examples	Popular Algorithms
Supervised Learning	Learns from labeled data to make predictions or classifications.	Email spam detection, disease diagnosis, stock price prediction	- Linear Regression - Logistic Regression - Decision Trees - Random Forest - Support Vector Machines (SVM) - k-Nearest Neighbors (k-NN) - Neural Networks
Unsupervised Learning	Finds hidden patterns or intrinsic structures in data without labeled outputs.	Customer segmentation, anomaly detection, topic modeling	- K-Means Clustering - Hierarchical Clustering - DBSCAN - Principal Component Analysis (PCA) - Autoencoders
Semi-Supervised Learning	Combines small amounts of labeled data with large amounts of unlabeled data.	Web content classification, medical imaging, speech analysis	- Semi-Supervised SVM - Label Propagation - Self-training - Graph-based methods

Table 1: types of learning and their examples

1.1 Active learning

A type of supervised learning where the model actively selects the most informative data points to be labeled, aiming to improve learning with fewer labeled examples.

Aspect	**Details**
Definition	A type of supervised learning where the model actively selects the most informative data points to be labeled, aiming to improve learning with fewer labeled examples.
Goal	Minimize labeling cost while maximizing model performance.
How It Works	The model queries a human (oracle) to label uncertain or ambiguous data points.
Use Cases	Medical diagnosis, text classification, fraud detection.
Query Strategies	- Uncertainty Sampling: Pick samples where the model is least confident. - Query-by-Committee: Use multiple models to find disagreement. - Expected Model Change: Select samples that would most change the model.

Table 2: Active Learning

Uncertainty Sampling often uses entropy: Entropy(x) = -$\sum_i P(y_i|x) \log P(y_i|x)$

1.2 Reinforcement Learning

An area of ML where an agent learns by interacting with an environment, receiving rewards for good actions and penalties for bad ones.

Aspect	Details
Definition	An area of ML where an agent learns by interacting with an environment, receiving rewards for good actions and penalties for bad ones.
Goal	Learn a policy that maximizes cumulative reward over time.
Key Concepts	- Agent: Learner - Environment: Where it acts - State: Current situation - Action: Choice made - Reward: Feedback - Policy (\pi): Strategy the agent follows
Use Cases	Game playing (e.g., AlphaGo), robotics, recommendation systems, self-driving cars
Popular Algorithms	- Q-Learning - Deep Q-Network (DQN) - SARSA - Policy Gradient Methods - Actor-Critic Models

Table 3: Reinforcement learning

Q(s,a) $\leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] Where :$
$s : currentstate$
$a : actiontaken$
$r : rewardreceived$
$s : nextstate$
$\alpha : learningrate$
$\gamma : discountfactor$
$This formularepresents the core Q-learning update rule in reinforcement learning, where the Q-value for a state-action pair is updated based on the immediate reward and the maximum expected future reward.$

2 Clustering Overview

Clustering is an **unsupervised learning** technique that aims to group similar data points into **clusters** such that

- Data points in the **same cluster** are similar (high cohesion).
- Data points in **different clusters** are dissimilar (high separation).

Used in:

- Customer segmentation
- Document/topic grouping
- Image segmentation
- Anomaly detection

2.1 Key ideas

1. **Cohesion (Intra-cluster similarity)**

   - **Definition**: Measures how close the data points in the **same cluster** are to each other.
   - **Goal**: Minimize cohesion (i.e., the tighter the cluster, the better).

   Formula (for cluster C_k with centroid μ_k):

   $$\text{Cohesion}(C_k) = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \tag{1}$$

   Lower cohesion = better compactness.

2. **Separation (Inter-cluster dissimilarity)**

   - **Definition**: Measures how far apart the **clusters** are from each other.
   - **Goal**: Maximize separation (i.e., more distinct clusters).

   Formula (between clusters C_i and C_j):

   $$\text{Separation}(C_i, C_j) = \|\mu_i - \mu_j\|^2 \tag{2}$$

   Higher separation = better distinctiveness between clusters.

3. **Scatter Coefficient (Cluster Evaluation Metric)**

   - **Definition**: Combines cohesion and separation to evaluate clustering quality.
   - Also related to **Dunn Index**, **Silhouette Score**, or **Davies-Bouldin Index**.

   One common scatter-based metric is the **Silhouette Coefficient** for a point i:

   $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3}$$

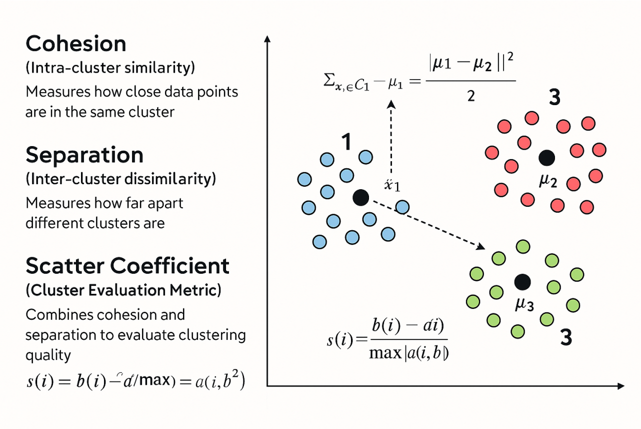
   Where:

   - $a(i)$: average distance between i and other points in the **same** cluster (cohesion)
   - $b(i)$: minimum average distance between i and points in **other** clusters (separation)

   $-1 \leq s(i) \leq 1$

   - $s(i) \approx 1$: well-clustered

- $s(i) \approx 0$: on the boundary
- $s(i) < 0$: likely in the wrong cluster



Cohesion
(Intra-cluster similarity)
Measures how close data points are in the same cluster

Separation
(Inter-cluster dissimilarity)
Measures how far apart different clusters are

Scatter Coefficient
(Cluster Evaluation Metric)
Combines cohesion and separation to evaluate clustering quality

$s(i) = b(i) - d/\text{max}) = a(i, b^2)$
$\Sigma_{x, \in C_1} - \mu_1 = \frac{|\mu 1 - \mu_2||^2}{2}$
3
1
$\dot{x}1$
μ_2
μ_3
3
$s(i) = \frac{b(i) - ai)}{\max |a(i, b|)}$

Figure 1: Description of the clustering metrics visualization

2.2 Analyzing distance function and metrics between two points

When analyzing distance functions or metrics between two points, there are key properties we often check: reflexivity, symmetry, and the triangle inequality. These are crucial in metric spaces and clustering algorithms like K-Means, DBSCAN, etc.

Let $x, y, z \in R^n$ be points (or vectors), and let $d(x, y)$ be the **distance** between x and y.

◇ 1. **Reflexivity**

- **Definition**: A point is always at zero distance from itself.

$$d(x, x) = 0 \tag{4}$$

- This means the distance between a point and itself is always zero.

◇ 2. **Symmetry**

- **Definition**: The distance from x to y is the same as from y to x.

$$d(x, y) = d(y, x) \tag{5}$$

- This ensures that the direction of comparison doesn't matter.

◇ 3. **Triangle Inequality**

  - **Definition**: The direct distance from x to z is never greater than going through an intermediate point y.

$$d(x,z) \leq d(x,y) + d(y,z) \tag{6}$$

  - This property ensures consistency and prevents “shortcut violations.”

Together, these properties define a **metric** or **distance function**:
A function $d : R^n \times R^n \to R$ is a **metric** if it satisfies :

1. **Non-negativity**: $d(x,y) \geq 0$
2. **Identity of indiscernibles**: $d(x,y) = 0 \iff x = y$
3. **Symmetry**: $d(x,y) = d(y,x)$
4. **Triangle inequality**: $d(x,z) \leq d(x,y) + d(y,z)$

Example: Euclidean Distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{7}$$

3 Clustering Concepts

Clustering algorithms generally rely on the idea of grouping data based on their similarities. The goal is to partition a set of data points into clusters where points within a cluster are more similar to each other than to those in other clusters.

3.1 Dendrogram

A *dendrogram* is a tree-like diagram used to visualize the process of hierarchical clustering. It records the steps of merging (in agglomerative clustering) or splitting (in divisive clustering) clusters at each stage. The height of the tree represents the distance at which the clusters are merged or split.
Key Concepts:
Leaves: Represent individual data points.
Branches: Represent clusters formed by merging.
Height (y-axis): Represents the distance (dissimilarity) between merged clusters.
Example: Dendrogram Using 5 Points.
Let’s say we have 5 points: A, B, C, D, E

And their pairwise distances are as follows:

	A	**B**	**C**	**D**	**E**
A	0	2	6	10	9
B	2	0	5	9	8
C	6	5	0	4	5
D	10	9	4	0	3
E	9	8	5	3	0

Table 4: dendrogram example (pairwise distances)

Clustering Process (Agglomerative)

- Merge A & B → distance = 2
- Merge D & E → distance = 3
- Merge C & (D,E) → distance = 5
- Merge (A,B) & (C,D,E) → distance = 6

Dendrogram visualization:

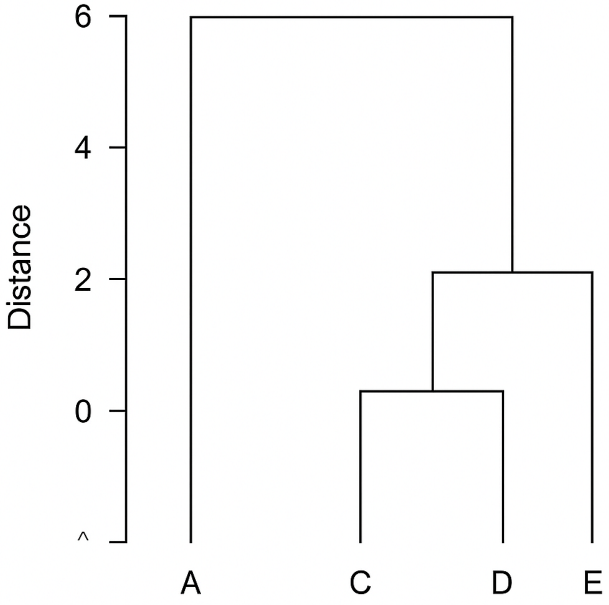


Figure 2: Dendrogram for hierarchical clustering.

4 Types of Hierarchical Clustering

There are two main types of hierarchical clustering:

4.1 Agglomerative Hierarchical Clustering (AHC)

Agglomerative clustering is a *bottom-up* approach. Initially, each data point is treated as a separate cluster. In each step, the two closest clusters are merged until all points belong to a single cluster.

4.1.1 Algorithm

1. Compute the pairwise distance matrix.
2. Treat each data point as a separate cluster.
3. Repeatedly merge the closest pair of clusters.
4. Stop when only one cluster remains.

4.1.2 Linkage Criteria

The distance between two clusters can be defined using different linkage criteria:

$$D(A,B) = \begin{cases} \min\limits_{x\in A, y\in B} d(x,y) & \text{(Single Linkage)} \\ \max\limits_{x\in A, y\in B} d(x,y) & \text{(Complete Linkage)} \\ \frac{1}{|A||B|}\sum_{x\in A}\sum_{y\in B} d(x,y) & \text{(Average Linkage)} \\ \|\mu_A - \mu_B\| & \text{(Centroid Linkage)} \end{cases}$$

where $d(x,y)$ represents the distance between two points x and y, and μ_A and μ_B are the centroids of clusters A and B, respectively.

4.2 Divisive Hierarchical Clustering (DHC)

Divisive clustering is a *top-down* approach. The algorithm starts with all data points in one large cluster and recursively splits the clusters into smaller ones until each data point is in its own cluster.

4.2.1 Algorithm

1. Start with the entire dataset as one cluster.
2. Split the cluster into two subclusters.
3. Recursively split the subclusters.
4. Continue until each data point is its own cluster.

5 Complexity Analysis

Let's discuss the time and space complexities of both agglomerative and divisive clustering.

5.1 Agglomerative Hierarchical Clustering

The naive time complexity of agglomerative clustering is derived from the following steps:

- Computing the distance matrix takes $O(n^2)$.
- Merging clusters takes $O(n^2)$ at each step, and there are $n - 1$ steps.

Thus, the total time complexity is:

$$T(n) = O(n^2) + (n - 1) \cdot O(n^2) = O(n^3)$$

Optimized versions, such as using priority queues, reduce the complexity to:

$$T(n) = O(n^2 \log n)$$

The space complexity is $O(n^2)$ due to the distance matrix.

5.2 Divisive Hierarchical Clustering

Divisive clustering has a worst-case complexity of $O(2^n)$, as it considers all possible splits of the data. However, if k-means or other clustering methods are used for splitting, the time complexity is reduced to:

$$T(n) = O(n^2 \log n)$$

The space complexity remains $O(n^2)$, as pairwise dissimilarities must be stored.

6 Merits and Demerits of Clustering

6.1 Merits of Clustering

Merit	**Explanation**
Unsupervised Learning	Clustering doesn't require labeled data, making it useful when labels are expensive or unavailable.
Data Exploration	It helps to uncover hidden patterns, structures, or relationships within the data.
Dimensionality Reduction	Clustering can act as a preprocessing step to reduce data complexity before applying other algorithms.
Scalability	Algorithms like k-means scale well for large datasets.
Adaptability	Clustering is used in various domains such as image processing, market segmentation, bioinformatics, etc.
Improves Interpretability	Clusters help organize data into meaningful categories, making it easier to interpret than raw data.

6.2 Demerits of Clustering

Demerit	Explanation
No Ground Truth	The evaluation of clustering results is challenging without predefined labels or clear metrics.
Choice of k	Some algorithms, like k-means, require the number of clusters k to be predefined.
Sensitivity to Initialization	Algorithms like k-means can converge to local optima based on initial cluster centroids.
Assumptions on Shape/Size	Many algorithms assume clusters are spherical or have similar sizes.
High Dimensionality Problems	In high-dimensional spaces, distance metrics lose their meaning (curse of dimensionality).
Noise and Outliers	Clustering algorithms are sensitive to outliers, especially centroid-based methods.

7 Conclusion

Clustering is a powerful tool in unsupervised learning, offering valuable insights into data structure. However, its performance and applicability depend on the specific problem at hand and the choice of algorithm. Understanding the complexities, merits, and limitations of various clustering techniques is crucial for selecting the best method for a given task.