Clustering

Luu Minh Sao Khue

December 16, 2025

1. Unsupervised Learning

Unsupervised learning refers to a class of machine learning methods where **no labeled outputs are provided**. The goal is to discover hidden structure, patterns, or regularities directly from the data. Unlike supervised learning, where the model learns a mapping $(\mathbf{x}, y)$, unsupervised learning operates only on input data $\mathbf{x}$ and answers exploratory questions such as:

- Are there natural groups in the data?
- Can the data be represented using fewer dimensions?
- What is the underlying distribution of the data?
- Which variables tend to occur together?

1.1 Main Tasks in Unsupervised Learning

Unsupervised learning encompasses several distinct tasks:

- **Clustering:** grouping similar data points together.
- **Dimensionality Reduction:** projecting data into a lower- dimensional space while preserving important structure (e.g., PCA).
- **Density Estimation:** modeling the probability distribution of the data (e.g., Gaussian Mixture Models).
- **Association Rule Mining:** discovering frequent co-occurring patterns (e.g., market basket analysis).
- **Topic Modeling:** uncovering latent themes in text data.

Among these tasks, **clustering** is one of the most widely used and intuitive unsupervised learning techniques.

2. Clustering

Clustering aims to partition a dataset into groups, called *clusters*, such that:

- data points within the same cluster are highly similar,
- data points in different clusters are dissimilar.

Importantly, clusters are formed **without any prior labels**, and the structure emerges solely from the data.

2.1 Intuition

Geometrically, clustering can be viewed as grouping points in a feature space based on proximity:

- points that are close together belong to the same cluster,
- points that are far apart belong to different clusters.

The definition of "close" depends on how similarity (or distance) is measured, which is a central concept in clustering.

2.2 Types of Clustering Methods

Clustering algorithms can be broadly categorized as follows:

- **Partition-based methods:** divide data into a fixed number of clusters (e.g., K-Means).
- **Hierarchical methods:** build nested clusters using a tree structure (agglomerative or divisive clustering).
- **Density-based methods:** identify clusters as dense regions separated by sparse areas (e.g., DBSCAN).
- **Model-based methods:** assume data is generated from a mixture of probabilistic models (e.g., Gaussian Mixture Models).

3. Similarity Between Data Points

At the core of clustering lies the notion of **similarity** (or dissimilarity) between data points. Most clustering algorithms rely on a distance or similarity function to compare samples.

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ be two data points.

3.1 Euclidean Distance

The most common distance measure is the Euclidean distance:

$$d_{\text{Euc}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2}.$$

Euclidean distance measures straight-line distance in feature space and is widely used in algorithms such as K-Means.

3.2 Manhattan Distance

The Manhattan (or ℓ_1) distance is defined as:

$$d_{\text{Man}}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} |x_j - y_j|.$$

It measures distance along coordinate axes and is often more robust to outliers.

3.3 Cosine Similarity

Cosine similarity measures the *angle* between two vectors rather than their absolute distance:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|_2 \, \|\mathbf{y}\|_2}.$$

It is particularly useful when the magnitude of vectors is not informative, such as in text or embedding-based representations.

A corresponding cosine distance can be defined as:

$$d_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \cos(\mathbf{x}, \mathbf{y}).$$

4. Feature Scaling

Distance-based clustering algorithms are highly sensitive to feature scale. When features have different units or numeric ranges, those with larger scales dominate distance computations.

4.1 Standardization

A common scaling method is standardization:

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j},$$

where μ_j and σ_j are the mean and standard deviation of feature j.

Standardization ensures that all features contribute comparably to distance calculations.

4.2 Min–Max Scaling

Alternatively, features can be scaled to a fixed interval:

$$x'_{ij} = \frac{x_{ij} - \min_j}{\max_j - \min_j}.$$

This maps feature values to $[0, 1]$ and preserves relative ordering.

5. Sample Normalization for Cosine Similarity

Feature scaling should not be confused with **sample normalization**. Normalization operates on individual samples rather than features.

5.1 L2 Normalization

For cosine-based similarity, each sample vector is normalized to unit length:

$$\mathbf{x}' = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}, \quad \text{so that } \|\mathbf{x}'\|_2 = 1.$$

This removes magnitude information and ensures that similarity depends only on direction.

5.2 Relation to Clustering

When all samples are L2-normalized, squared Euclidean distance becomes proportional to cosine distance:

$$\|\mathbf{x} - \mathbf{y}\|_2^2 = 2(1 - \cos(\mathbf{x}, \mathbf{y})).$$

As a result, applying K-Means to normalized data effectively performs cosine-based clustering, often referred to as *Spherical K-Means*.