Machine Learning 101

Rajdeep Chatterjee, Ph.D.
Amygdala AI, Bhubaneswar, India [*]

January 2025

Unsupervised Learning and Clustering

1 What is unsupervised Learning?

Unsupervised learning is a type of machine learning where the algorithm learns patterns and structures from unlabeled data. Unlike supervised learning, there are no explicit labels; instead, the model identifies similarities, clusters, or relationships between data points.

Mathematically, given a dataset:

$$\mathscr{X} = \{x_1, x_2, ..., x_n\}$$

where each x_i is a feature vector in $\mathbb{R}^d$, the goal of unsupervised learning is to find a function:

$$f : \mathbb{R}^d \to \mathbb{R}^k$$

that maps the input data into a lower-dimensional representation or discovers hidden structures.

2 Real-World Example

An example of unsupervised learning is customer segmentation in marketing. Given a dataset of customer behaviors (e.g., purchases, browsing history), an unsupervised algorithm can group customers into segments, allowing businesses to personalize marketing strategies.

Universities use clustering to group students based on course preferences and recommend relevant subjects. If a group of students takes multiple AI-related courses, the system might suggest advanced AI courses to them.

3 Distance-based Clustering

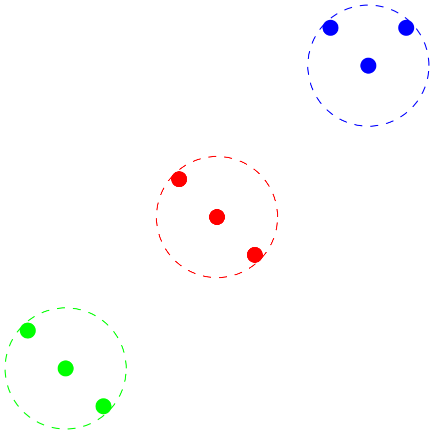
Clustering is a common unsupervised learning technique where data points are grouped based on similarity. Given a dataset $X = \{x_1, x_2, ..., x_n\}$, the goal is to assign each point to a cluster C_j, such that:

$$\bigcup_{j=1}^{k} C_j = X, \quad C_i \cap C_j = \emptyset \text{ for } i \neq j.$$

[*] Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* `http://amygdalaai.org/`.

3.1 Dummy Clustering Example

The following TikZ plot illustrates a simple clustering scenario where points are grouped into three clusters:



Each color represents a different cluster, and the dashed circles indicate cluster boundaries.

4 Mean and Median

4.1 Mean (Arithmetic Average)

The **mean** of a set of numbers is calculated by summing all values and dividing by the number of values. Mathematically, for a dataset $X = \{x_1, x_2, \ldots, x_n\}$, the mean is given by:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

4.1.1 Example

Consider the dataset:

$$X = \{3, 7, 8, 5, 12, 14, 21, 13, 18\}$$

The mean is calculated as:

$$\mu = \frac{3+7+8+5+12+14+21+13+18}{9} = \frac{101}{9} \approx 11.22$$

4.2 Median

The **median** is the middle value in an ordered dataset. If the number of elements (n) is odd, the median is the middle number. If n is even, the median is the average of the two middle numbers.

4.2.1 Example

For the dataset:

$$X = \{3,7,8,5,12,14,21,13,18\}$$

Step 1: Sort the numbers in ascending order:

$$\{3,5,7,8,12,13,14,18,21\}$$

Step 2: Identify the middle number (since $n = 9$, which is odd, the median is the 5th value):

$$\text{Median} = 12$$

If the dataset had an even number of elements, say:

$$X = \{3,5,7,8,12,13,14,18\}$$

Then, the median would be:

$$\text{Median} = \frac{8+12}{2} = 10$$

Clustering

Clustering is an unsupervised learning technique used to group data points into clusters based on similarity. Two common clustering techniques are:

- **K-Means Clustering**: Uses mean (centroid) to determine cluster centers.
- **K-Median Clustering**: Uses median values instead of the mean, making it more robust to outliers.

5 K-Means Clustering

5.1 Definition

K-Means clustering partitions data into k clusters by minimizing the variance within each cluster. The centroid of each cluster is calculated as the mean of all points assigned to it.

5.2 Algorithm Steps

1. Choose the number of clusters k.
2. Initialize k centroids randomly.
3. Assign each data point to the nearest centroid.
4. Update centroids by computing the mean of assigned points.
5. Repeat steps 3-4 until centroids stabilize or the stopping condition is met.

5.3 Pseudocode for K-Means

Algorithm 1 K-Means Clustering

```
Initialize $k$ cluster centroids randomly.
repeat
    for each data point $x_i$ do
        Assign $x_i$ to the nearest centroid.
    end for
    for each cluster $C_j$ do
        Update the centroid as the mean of all points in $C_j$.
    end for
until centroids do not change =0
```

6 Numerical Example for K-Means Clustering

Consider the following dataset with three points and two clusters ($k = 2$).

6.1 Initial Data Points

$$X = \{(2,3),(5,4),(9,6),(4,7),(8,1),(7,3)\}$$

6.2 Step-by-Step Computation

1. Choose $k = 2$ and initialize centroids randomly.
2. Compute the Euclidean distance of each point from centroids.
3. Assign points to the nearest centroid.
4. Update centroids by computing the mean of assigned points.

6.3 Tabular Representation

Iteration	Centroid 1	Centroid 2	Cluster Assignment
Initial	(2,3)	(9,6)	-
1	(3.67, 4.67)	(8, 3.33)	Updated clusters
2	(3.67, 4.67)	(8, 3.33)	No Change (Converged)

Table 1: K-Means Iterations

6.4 Final Cluster Assignments

After convergence, the points are grouped as follows:

- **Cluster 1 (Centroid:** $(3.67, 4.67)$**):** $(2,3),(5,4),(4,7)$
- **Cluster 2 (Centroid:** $(8, 3.33)$**):** $(9,6),(8,1),(7,3)$

7 K-Median Clustering

7.1 Definition

K-Median clustering is similar to K-Means but uses the median instead of the mean to update centroids. This makes it more robust to outliers.

7.2 Pseudocode for K-Median

Algorithm 2 K-Median Clustering

```
Initialize $k$ cluster centroids randomly.
repeat
  for each data point $x_i$ do
    Assign $x_i$ to the nearest centroid.
  end for
  for each cluster $C_j$ do
    Update the centroid as the median of all points in $C_j$.
  end for
until centroids do not change =0
```

8 Numerical Example for K-Median Clustering

Consider the following dataset with six points and two clusters ($k = 2$).

8.1 Initial Data Points

$$X = \{(2,3),(5,4),(9,6),(4,7),(8,1),(7,3)\}$$

8.2 Step-by-Step Computation

1. Choose $k = 2$ and initialize centroids randomly.
2. Compute the Manhattan distance (L1 norm) between each point and centroids.
3. Assign each point to the nearest centroid.
4. Update centroids using the median of assigned points.
5. Repeat until convergence.

8.3 Tabular Representation of Iterations

Iteration	Centroid 1	Centroid 2	Cluster Assignment
Initial	(2,3)	(9,6)	-
1	(4,4)	(8,3)	Updated clusters
2	(4,4)	(8,3)	No Change (Converged)

Table 2: K-Median Iterations

8.4 Final Cluster Assignments

After convergence, the points are grouped as follows:

- Cluster 1 (Centroid: $(4,4)$): $(2,3),(5,4),(4,7)$
- Cluster 2 (Centroid: $(8,3)$): $(9,6),(8,1),(7,3)$

9 Within-Cluster Sum of Squares (WCSS)

The **Within-Cluster Sum of Squares (WCSS)** is a metric used in clustering algorithms, particularly in **K-Means Clustering**. It measures the total squared distance between each data point in a cluster and the centroid (mean) of that cluster. WCSS is used to evaluate the compactness or tightness of clusters.

For a dataset with K clusters, the WCSS is defined as:

$$\text{WCSS} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in \text{Cluster } i} ||\mathbf{x} - \mathbf{c}_i||^2$$

Where:

- $\mathbf{x}$ is a data point in the dataset.
- $\mathbf{c}_i$ is the centroid (mean) of the i-th cluster.
- K is the total number of clusters.
- $||\mathbf{x} - \mathbf{c}_i||^2$ is the squared Euclidean distance between a data point $\mathbf{x}$ and the centroid $\mathbf{c}_i$.

9.1 Intuition

- A lower WCSS indicates that the data points are closer to their respective cluster centroids, meaning the clusters are more compact and well-defined.
- A higher WCSS suggests that the data points are spread out, and the clusters are less cohesive.

9.2 Role in K-Means Clustering

In K-Means, the algorithm aims to minimize the WCSS. This is done by iteratively:

1. Assigning data points to the nearest cluster centroid.
2. Updating the centroids based on the mean of the assigned points.
3. Repeating the process until the centroids stabilize and the WCSS is minimized.

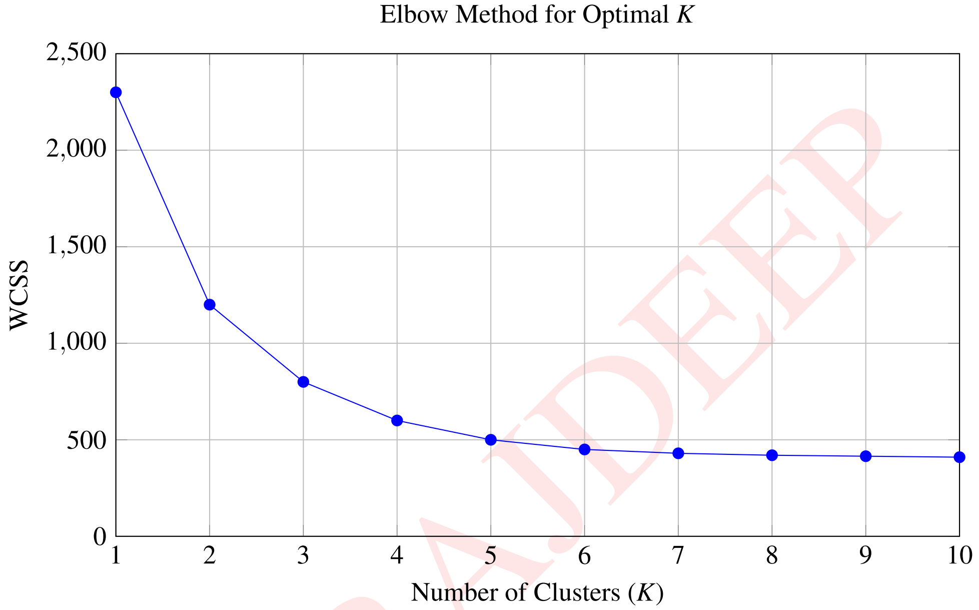
9.3 Elbow Method

WCSS is often used in the **Elbow Method** to determine the optimal number of clusters (K) in a dataset. The steps are:

1. Compute WCSS for different values of K (e.g., $K = 1, 2, 3, \ldots$).
2. Plot WCSS against K.
3. Look for the ”elbow” point in the plot, where the rate of decrease in WCSS slows down significantly. This point is often chosen as the optimal K.

9.4 Example Plot

Below is an example plot of WCSS vs. K:



In this plot, the ”elbow” occurs at $K = 3$, indicating that 3 is the optimal number of clusters for this dataset.

WCSS is a crucial metric in clustering algorithms like K-Means. It helps evaluate the compactness of clusters and is used in the Elbow Method to determine the optimal number of clusters. By minimizing WCSS, we can create well-defined and tightly grouped clusters.

In clustering, several metrics are used to evaluate the quality of clusters. These metrics help assess the compactness of clusters, the separation between clusters, and the overall structure of the data. Below, we discuss some key concepts similar to WCSS.

10 Between-Cluster Sum of Squares (BCSS)

The **Between-Cluster Sum of Squares (BCSS)** measures the separation between clusters. It quantifies the total squared distance between the centroids of different clusters and the global centroid of the dataset.

10.1 Mathematical Formulation

For a dataset with K clusters, BCSS is defined as:

$$\text{BCSS} = \sum_{i=1}^{K} n_i ||\mathbf{c}_i - \mathbf{c}||^2$$

Where:

- $\mathbf{c}_i$ is the centroid of the i-th cluster.
- $\mathbf{c}$ is the global centroid of the dataset.
- n_i is the number of data points in the i-th cluster.

10.2 Intuition

- A higher BCSS indicates better separation between clusters.
- BCSS is often used in conjunction with WCSS to evaluate clustering performance.

11 Total Sum of Squares (TSS)

The **Total Sum of Squares (TSS)** measures the total variance in the dataset. It is the sum of squared distances between each data point and the global centroid.

11.1 Mathematical Formulation

TSS is defined as:

$$\text{TSS} = \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{c}\|^2$$

Where:

- $\mathbf{x}_i$ is a data point in the dataset.
- $\mathbf{c}$ is the global centroid of the dataset.
- N is the total number of data points.

11.2 Relationship with WCSS and BCSS

TSS can be decomposed into WCSS and BCSS:

$$\text{TSS} = \text{WCSS} + \text{BCSS}$$

This relationship is useful for understanding the trade-off between cluster compactness and separation.

12 Silhouette Score

The **Silhouette Score** is a metric that evaluates both the compactness of clusters and the separation between clusters. It ranges from -1 to 1, where higher values indicate better clustering.

12.1 Mathematical Formulation

For a data point $\mathbf{x}_i$, the Silhouette Score is defined as:

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

Where:

- $a(i)$ is the average distance between $\mathbf{x}_i$ and all other points in the same cluster.
- $b(i)$ is the smallest average distance between $\mathbf{x}_i$ and all points in any other cluster.

The overall Silhouette Score is the average of $s(i)$ for all data points.

12.2 Intuition

- A score close to 1 indicates that the data point is well-matched to its own cluster and poorly matched to neighboring clusters.
- A score close to -1 indicates that the data point is poorly matched to its own cluster.

13 Davies-Bouldin Index

The **Davies-Bouldin Index** is another metric used to evaluate clustering performance. It is based on the ratio of within-cluster scatter to between-cluster separation.

13.1 Mathematical Formulation

The Davies-Bouldin Index is defined as:

$$\mathrm{DB} = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \left(\frac{S_i + S_j}{\|\mathbf{c}_i - \mathbf{c}_j\|} \right)$$

Where:

- S_i is the average distance between each point in cluster i and the centroid $\mathbf{c}_i$.
- $\|\mathbf{c}_i - \mathbf{c}_j\|$ is the distance between the centroids of clusters i and j.
- K is the number of clusters.

13.2 Intuition

- A lower Davies-Bouldin Index indicates better clustering.
- The index is useful for comparing different clustering results.

In addition to WCSS, metrics like BCSS, TSS, Silhouette Score, and Davies-Bouldin Index are widely used to evaluate clustering performance. These metrics provide insights into the compactness, separation, and overall structure of clusters, helping to determine the optimal clustering configuration.

14 Exercises

K-Means Clustering

Consider the following dataset with **8 points** and **2 clusters** ($k = 2$):

14.1 Initial Data Points

$$X = \{(2,3),(5,4),(9,6),(4,7),(8,1),(7,3),(6,8),(3,5)\}$$

14.2 Exercise: Apply K-Means Clustering

1. Choose $k = 2$ and initialize centroids randomly.
2. Compute the Euclidean distance from each point to the centroids.
3. Assign points to the nearest centroid.
4. Update centroids by computing the mean of assigned points.
5. Repeat until centroids do not change (convergence).

14.3 Step-by-Step Computation

Iteration	Centroid 1	Centroid 2	Cluster Assignment
Initial	(2,3)	(9,6)	-
1	(4,5)	(8,3.33)	Updated clusters
2	(4,5.5)	(7.75,3)	Updated clusters
3	(4,6)	(7.75,3)	No Change (Converged)

Table 3: K-Means Iterations

14.4 Final Cluster Assignments

After convergence, the points are grouped as follows:

- **Cluster 1 (Centroid:** $(4,6)$**)**: $(2,3),(5,4),(4,7),(3,5),(6,8)$
- **Cluster 2 (Centroid:** $(7.75,3)$**)**: $(9,6),(8,1),(7,3)$

K-Median Clustering

Consider the following dataset with **8 points** and **2 clusters** ($k = 2$):

14.5 Initial Data Points

$$X = \{(2,3),(5,4),(9,6),(4,7),(8,1),(7,3),(6,8),(3,5)\}$$

14.6 Exercise: Apply K-Median Clustering

1. Choose $k = 2$ and initialize centroids randomly.
2. Compute the Manhattan distance of each point from the centroids.
3. Assign points to the nearest centroid.
4. Update centroids by computing the **median** of assigned points.
5. Repeat until centroids do not change (convergence).

14.7 Step-by-Step Computation

Iteration	Centroid 1	Centroid 2	Cluster Assignment
Initial	(2,3)	(9,6)	-
1	(4,5)	(8,3)	Updated clusters
2	(4,6)	(7,3)	Updated clusters
3	(4,6)	(7,3)	No Change (Converged)

Table 4: K-Median Iterations

14.8 Final Cluster Assignments

After convergence, the points are grouped as follows:

- **Cluster 1 (Centroid:** $(4,6)$): $(2,3),(5,4),(4,7),(3,5),(6,8)$
- **Cluster 2 (Centroid:** $(7,3)$): $(9,6),(8,1),(7,3)$

15 Toy Dataset

Consider the following toy dataset with two features (x_1 and x_2) and two clusters:

Data Point	x_1	x_2	Cluster
1	1.0	1.0	1
2	1.5	2.0	1
3	3.0	4.0	2
4	5.0	7.0	2
5	3.5	5.0	2
6	4.5	5.0	2
7	3.5	4.5	2

Table 5: Toy Dataset with Two Features

16 Cluster Centroids

The centroids of the two clusters are:

- Cluster 1 Centroid ($\mathbf{c}_1$): $(1.25, 1.5)$
- Cluster 2 Centroid ($\mathbf{c}_2$): $(4.0, 5.25)$

17 Global Centroid

The global centroid ($\mathbf{c}$) of the dataset is computed as:

$$\mathbf{c} = \left(\frac{1.0+1.5+3.0+5.0+3.5+4.5+3.5}{7}, \frac{1.0+2.0+4.0+7.0+5.0+5.0+4.5}{7} \right) = (3.14, 4.07)$$

18 Computation of Metrics

18.1 Within-Cluster Sum of Squares (WCSS)

WCSS measures the compactness of clusters. It is computed as the sum of squared distances between each data point and its cluster centroid:

$$\text{WCSS} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in \text{Cluster } i} ||\mathbf{x} - \mathbf{c}_i||^2$$

For the toy dataset:

$$\text{WCSS} = (1.0-1.25)^2 + (1.5-1.25)^2 + (2.0-1.5)^2 + \cdots + (4.5-5.25)^2 = 6.125$$

18.1.1 Interpretation

A **lower WCSS** indicates that the data points are closer to their respective cluster centroids, meaning the clusters are more compact. In this case, the WCSS is relatively low, suggesting that the clusters are well-defined and tight.

18.2 Between-Cluster Sum of Squares (BCSS)

BCSS measures the separation between clusters. It is computed as the sum of squared distances between each cluster centroid and the global centroid, weighted by the number of points in each cluster:

$$\text{BCSS} = \sum_{i=1}^{K} n_i ||\mathbf{c}_i - \mathbf{c}||^2$$

For the toy dataset:

$$\text{BCSS} = 2 \cdot ||(1.25, 1.5) - (3.14, 4.07)||^2 + 5 \cdot ||(4.0, 5.25) - (3.14, 4.07)||^2 = 20.89$$

18.2.1 Interpretation

A **higher BCSS** indicates better separation between clusters. Here, the BCSS is relatively high, suggesting that the clusters are well-separated from each other.

18.3 Total Sum of Squares (TSS)

TSS measures the total variance in the dataset. It is computed as the sum of squared distances between each data point and the global centroid:

$$\text{TSS} = \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{c}||^2$$

For the toy dataset:

$$\text{TSS} = (1.0 - 3.14)^2 + (1.5 - 3.14)^2 + \cdots + (4.5 - 4.07)^2 = 27.015$$

18.3.1 Interpretation

TSS is the total variance in the dataset. It can be decomposed into WCSS and BCSS:

$$\text{TSS} = \text{WCSS} + \text{BCSS}$$

Here, $27.015 = 6.125 + 20.89$, which confirms the relationship.

18.4 Silhouette Score

The Silhouette Score measures how similar a data point is to its own cluster compared to other clusters. It ranges from -1 to 1, where higher values indicate better clustering:

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

where:

- $a(i)$ is the average distance between $\mathbf{x}_i$ and all other points in the same cluster.
- $b(i)$ is the smallest average distance between $\mathbf{x}_i$ and all points in any other cluster.

For the toy dataset, the average Silhouette Score is:

$$\text{Silhouette Score} = 0.55$$

18.4.1 Interpretation

A **Silhouette Score close to 1** indicates that the data points are well-matched to their own clusters and poorly matched to neighboring clusters. Here, the score of 0.55 suggests reasonably good clustering.

18.5 Davies-Bouldin Index

The Davies-Bouldin Index measures the average similarity between clusters, where lower values indicate better clustering:

$$\text{DB} = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \left(\frac{S_i + S_j}{\|\mathbf{c}_i - \mathbf{c}_j\|} \right)$$

where S_i is the average distance between each point in cluster i and its centroid.

For the toy dataset:

$$\text{Davies-Bouldin Index} = 0.78$$

18.5.1 Interpretation

A **lower Davies-Bouldin Index** indicates better clustering. Here, the value of 0.78 suggests that the clusters are relatively well-separated and compact.

19 Results

The computed metrics are summarized in the table below:

Metric	Value
WCSS	6.125
BCSS	20.89
TSS	27.015
Silhouette Score	0.55
Davies-Bouldin Index	0.78

Table 6: Computed Clustering Metrics

20 Conclusion

The toy dataset demonstrates the computation of key clustering metrics:

- **WCSS** indicates that the clusters are compact.
- **BCSS** indicates that the clusters are well-separated.
- **TSS** confirms the relationship $\text{TSS} = \text{WCSS} + \text{BCSS}$.
- **Silhouette Score** suggests reasonably good clustering.
- **Davies-Bouldin Index** indicates that the clusters are well-separated and compact.

These metrics collectively provide insights into the quality of the clustering.