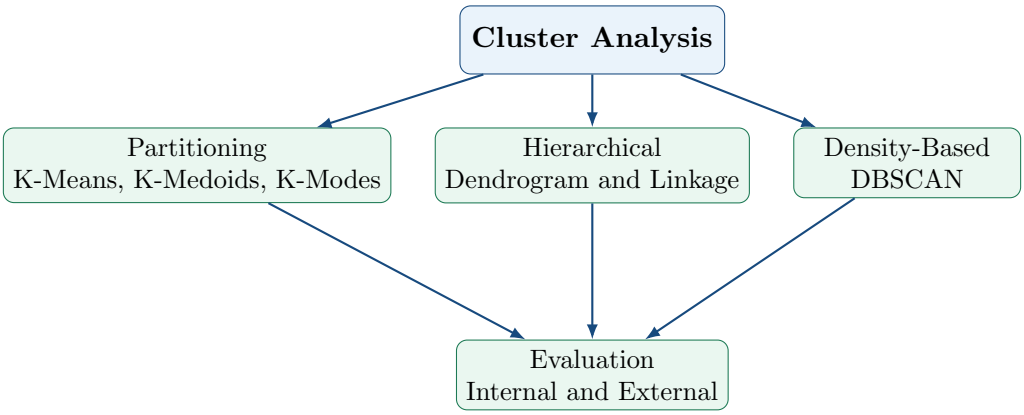
Cluster Analysis

Topics: Basic Concepts, K-Means, K-Medoids, K-Modes, Hierarchical Clustering, DBSCAN, and Evaluation

Central Idea

Clustering is an **unsupervised learning** task that divides objects into groups so that objects in the same group are highly similar, while objects in different groups are dissimilar. The meaning of "similar" depends on the data type, distance measure, and clustering method.



1 Cluster Analysis: Basic Concepts

Definition

Let a dataset contain n objects,

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \qquad \mathbf{x}_i \in \mathbb{R}^d.$$

A hard clustering with k clusters is a partition

$$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$$

such that

$$C_i \cap C_j = \varnothing \ (i \neq j), \qquad \bigcup_{j=1}^{k} C_j = X.$$

Good clusters have high **intra-cluster similarity** and low **inter-cluster similarity**.

1.1 Main types of clustering

Type	Main idea	Typical method
Partitioning	Directly divide the data into k non-overlapping clusters	K-Means, K-Medoids, K-Modes
Hierarchical	Build nested clusters by repeated merging or splitting	Agglomerative or divisive clustering
Density-based	Dense regions form clusters; sparse points may be noise	DBSCAN
Hard clustering	Every object belongs to exactly one cluster	Standard K-Means
Soft/fuzzy clustering	An object has a degree of membership in several clusters	Fuzzy C-Means

Table 1: Important clustering categories.

1.2 Similarity and distance measures

Key Formula

For two numerical vectors $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{y} = (y_1, \ldots, y_d)$:

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2} \qquad \text{(Euclidean)},$$

$$d_M(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} |x_j - y_j| \qquad \text{(Manhattan)},$$

$$d_{\text{Minkowski}}(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{d} |x_j - y_j|^p \right)^{1/p}.$$

For categorical data with d attributes, simple matching dissimilarity is

$$d_{\text{SM}}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} \delta(x_j, y_j), \qquad \delta(x_j, y_j) = \begin{cases} 0, & x_j = y_j, \\ 1, & x_j \neq y_j. \end{cases}$$

Worked Example 1: Full Data Table and Pairwise Distances

Consider the following six two-dimensional objects.

Object	x	y
P_1	1	1
P_2	2	1
P_3	4	5
P_4	5	4
P_5	8	8
P_6	9	8

Step 1: Calculate one Euclidean distance.

$$d_E(P_1, P_3) = \sqrt{(1-4)^2 + (1-5)^2} = \sqrt{9 + 16} = 5.$$

Step 2: Calculate the corresponding Manhattan distance.

$$d_M(P_1, P_3) = |1-4| + |1-5| = 3 + 4 = 7.$$

Step 3: Construct the complete Euclidean distance table.

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0	1.000	5.000	5.000	9.899	10.630
P_2	1.000	0	4.472	4.243	9.220	9.899
P_3	5.000	4.472	0	1.414	5.000	5.099
P_4	5.000	4.243	1.414	0	5.000	5.657
P_5	9.899	9.220	5.000	5.000	0	1.000
P_6	10.630	9.899	5.099	5.657	1.000	0

The matrix is symmetric because $d(P_i, P_j) = d(P_j, P_i)$, and every diagonal entry is zero.

1.3 Why scaling matters

Key Formula

When features have very different units, standardize feature j by

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j},$$

where $\bar{x}_j$ is the feature mean and s_j is its standard deviation. This prevents a large-scale feature from dominating the distance.

Note

Centroid, medoid, and mode are different. The centroid is an arithmetic mean and may not be an observed object. A medoid is an actual object minimizing total dissimilarity. A mode is the most frequent categorical value in each attribute.

2 Partitioning Method I: K-Means

Definition

K-Means divides numerical data into k clusters. Each cluster is represented by its centroid. It minimizes the within-cluster sum of squared errors (SSE):

$$J = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2, \qquad \boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i.$$

2.1 Algorithm

1. Choose k initial centroids.
2. Assign every object to its nearest centroid.
3. Recalculate each centroid as the mean of its assigned objects.
4. Repeat Steps 2–3 until assignments no longer change.

Worked Example 2: K-Means with Six Data Points

Use $k = 2$ for the following complete dataset.

Point	x	y
A	1	1
B	1	2
C	2	1
D	8	8
E	8	9
F	9	8

Choose deliberately imperfect initial centroids

$$\boldsymbol{\mu}_1^{(0)} = A = (1, 1), \qquad \boldsymbol{\mu}_2^{(0)} = C = (2, 1).$$

Iteration 1: assignment step. Use squared Euclidean distance; the square root is unnecessary because it does not change which centroid is nearer.

Point	$\|x - \mu_1^{(0)}\|^2$	$\|x - \mu_2^{(0)}\|^2$	Assignment
A	0	1	C_1
B	1	2	C_1
C	1	0	C_2
D	98	85	C_2
E	113	100	C_2
F	113	100	C_2

Thus,

$$C_1^{(1)} = \{A, B\}, \qquad C_2^{(1)} = \{C, D, E, F\}.$$

Iteration 1: update step.

$$\boldsymbol{\mu}_1^{(1)} = \left(\frac{1+1}{2}, \frac{1+2}{2}\right) = (1, 1.5),$$
$$\boldsymbol{\mu}_2^{(1)} = \left(\frac{2+8+8+9}{4}, \frac{1+8+9+8}{4}\right) = (6.75, 6.50).$$

Iteration 2: assignment step.

Point	$\|x-\mu_1^{(1)}\|^2$	$\|x-\mu_2^{(1)}\|^2$	Assignment
A	0.25	63.3125	C_1
B	0.25	53.3125	C_1
C	1.25	52.8125	C_1
D	91.25	3.8125	C_2
E	105.25	7.8125	C_2
F	106.25	7.3125	C_2

The point C moves from C_2 to C_1:

$$C_1^{(2)} = \{A, B, C\}, \qquad C_2^{(2)} = \{D, E, F\}.$$

Iteration 2: update step.

$$\boldsymbol{\mu}_1^{(2)} = \left(\frac{1+1+2}{3}, \frac{1+2+1}{3}\right) = \left(\frac{4}{3}, \frac{4}{3}\right),$$
$$\boldsymbol{\mu}_2^{(2)} = \left(\frac{8+8+9}{3}, \frac{8+9+8}{3}\right) = \left(\frac{25}{3}, \frac{25}{3}\right).$$

Iteration 3: convergence check. Reassignment to these centroids gives the same two clusters, so the algorithm stops.
Final SSE.

$$\mathrm{SSE}(C_1) = \frac{2}{9} + \frac{5}{9} + \frac{5}{9} = \frac{4}{3},$$
$$\mathrm{SSE}(C_2) = \frac{5}{9} + \frac{5}{9} + \frac{2}{9} = \frac{4}{3},$$
$$\mathrm{SSE}_{\text{total}} = \frac{4}{3} + \frac{4}{3} = \boxed{\frac{8}{3} \approx 2.667}.$$

K-Means Result

$$C_1 = \{A, B, C\}, \quad \mu_1 = (1.333, 1.333), \qquad C_2 = \{D, E, F\}, \quad \mu_2 = (8.333, 8.333).$$

K-Means is fast and effective for compact, roughly spherical numerical clusters, but it is sensitive to initialization, feature scale, and outliers.

3 Partitioning Method II: K-Medoids

Definition

K-Medoids represents every cluster by an actual data object called a **medoid**. For medoids $m_1, \ldots, m_k$, the objective is

$$J = \sum_{i=1}^{n} \min_{1 \le j \le k} d(\mathbf{x}_i, m_j).$$

Because it uses dissimilarities rather than squared distances to a mean, it is usually more robust to outliers than K-Means.

3.1 PAM-style procedure

1. Select k data objects as initial medoids.

2. Assign each non-medoid object to its nearest medoid.
3. Try swapping a medoid with a non-medoid.
4. Accept a swap if it reduces total cost; stop when no swap improves the cost.

Worked Example 3: K-Medoids with Six 2D Data Points

Use $k = 2$ and Manhattan distance

$$d_M(P_i, P_j) = |x_i - x_j| + |y_i - y_j|.$$

The complete two-dimensional dataset is:

Point	x	y
A	1	1
B	2	1
C	2	2
D	8	8
E	9	8
F	12	12

Step 1: Calculate the pairwise dissimilarities. For example,

$$d_M(A, D) = |1 - 8| + |1 - 8| = 7 + 7 = 14,$$

$$d_M(E, F) = |9 - 12| + |8 - 12| = 3 + 4 = 7.$$

The complete distance matrix is:

	A	B	C	D	E	F
A	0	1	2	14	15	22
B	1	0	1	13	14	21
C	2	1	0	12	13	20
D	14	13	12	0	1	8
E	15	14	13	1	0	7
F	22	21	20	8	7	0

Step 2: Choose initial medoids $m_1 = B = (2, 1)$ and $m_2 = D = (8, 8)$.

Point	$d(P, B)$	$d(P, D)$	Nearest medoid	Cost
A	1	14	B	1
B	0	13	B	0
C	1	12	B	1
D	13	0	D	0
E	14	1	D	1
F	21	8	D	8
			Total cost	11

Thus the initial clusters are

$$C_1 = \{A, B, C\}, \qquad C_2 = \{D, E, F\}.$$

Step 3: Try the swap $D \to E$.

Point	$d(P,B)$	$d(P,E)$	Nearest medoid	Cost
A	1	15	B	1
B	0	14	B	0
C	1	13	B	1
D	13	1	E	1
E	14	0	E	0
F	21	7	E	7
			Total cost	10

Since $10 < 11$, accept the swap. The current medoids are B and E.

Step 4: Check every remaining one-swap alternative.

Candidate medoids	Cost	Candidate medoids	Cost
$\{A, E\}$	11	$\{B, A\}$	49
$\{C, E\}$	11	$\{B, C\}$	46
$\{D, E\}$	46	$\{B, D\}$	11
$\{F, E\}$	43	$\{B, F\}$	17

Every alternative costs more than 10. Therefore, no further swap is accepted.

K-Medoids Result

$$m_1 = B = (2,1), \quad C_1 = \{A, B, C\},$$
$$m_2 = E = (9,8), \quad C_2 = \{D, E, F\}, \qquad \boxed{J = 10}.$$

Both representatives are actual two-dimensional observations. The distant point $F = (12, 12)$ does not create a non-observed representative.

4 Partitioning Method III: K-Modes

Definition

K-Modes extends the partitioning idea to categorical data. It replaces:

$$\text{mean} \longrightarrow \text{mode}, \qquad \text{Euclidean distance} \longrightarrow \text{simple matching dissimilarity}.$$

The objective is

$$J = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} d_{\text{SM}}(\mathbf{x}_i, \mathbf{m}_j),$$

where $\mathbf{m}_j$ contains the most frequent category in each attribute of cluster C_j.

Worked Example 4: K-Modes with Eight 2D Categorical Points

K-Modes does not use numerical coordinates. Here, "2D" means that each object has exactly two categorical dimensions: *Color* and *Shape*. Let $k = 2$.

Object	Dimension 1: Color	Dimension 2: Shape
A	Red	Circle
B	Red	Circle
C	Blue	Square
D	Blue	Square
E	Red	Circle
F	Blue	Square
G	Red	Square
H	Blue	Circle

Choose the initial modes

$$m_1^{(0)} = (\text{Red, Circle}), \qquad m_2^{(0)} = (\text{Blue, Circle}).$$

Iteration 1: count the mismatching dimensions. For example,

$$d_{\text{SM}}(C, m_1^{(0)}) = 2, \qquad d_{\text{SM}}(C, m_2^{(0)}) = 1.$$

Object	$d(x, m_1^{(0)})$	$d(x, m_2^{(0)})$	Assignment
A	0	1	C_1
B	0	1	C_1
C	2	1	C_2
D	2	1	C_2
E	0	1	C_1
F	2	1	C_2
G	1	2	C_1
H	1	0	C_2

Therefore,

$$C_1 = \{A, B, E, G\}, \qquad C_2 = \{C, D, F, H\}.$$

Iteration 1: update each mode using category frequencies.

Cluster	Color frequencies	Shape frequencies
C_1	Red: 4	Circle: 3, Square: 1
C_2	Blue: 4	Square: 3, Circle: 1

Thus,

$$m_1^{(1)} = (\text{Red, Circle}), \qquad m_2^{(1)} = (\text{Blue, Square}).$$

Iteration 2: calculate distances to the updated modes.

Object	$d(x, m_1^{(1)})$	$d(x, m_2^{(1)})$	Assignment
A	0	2	C_1
B	0	2	C_1
C	2	0	C_2
D	2	0	C_2
E	0	2	C_1
F	2	0	C_2
G	1	1	C_1 (retain cluster)
H	1	1	C_2 (retain cluster)

For a tie, retain the object's previous cluster. The assignments are unchanged. The final objective is

$$J = (0+0+0+1) + (0+0+0+1) = \boxed{2}.$$

K-Modes Result

$$C_1 = \{A, B, E, G\}, \quad m_1 = (\text{Red, Circle}),$$
$$C_2 = \{C, D, F, H\}, \quad m_2 = (\text{Blue, Square}), \qquad \boxed{J = 2}.$$

The objects have two categorical dimensions, so the representative is an attribute-wise mode rather than a numerical mean.

5 Hierarchical Clustering

Definition

Hierarchical clustering creates a nested sequence of clusters represented by a dendrogram.

- **Agglomerative:** start with singleton clusters and repeatedly merge the closest pair.
- **Divisive:** start with one cluster and repeatedly split it.

The method does not require a final k in advance; k can be selected by cutting the dendrogram at a chosen height.

5.1 Linkage rules

For clusters A and B:

$$d_{\text{single}}(A, B) = \min_{x \in A, y \in B} d(x, y),$$
$$d_{\text{complete}}(A, B) = \max_{x \in A, y \in B} d(x, y),$$
$$d_{\text{average}}(A, B) = \frac{1}{|A||B|} \sum_{x \in A} \sum_{y \in B} d(x, y).$$

Terminology Used in This Note

The standard three linkage rules are **single**, **average**, and **complete** linkage. Because "double linkage" is not a standard linkage formula in most data-mining texts, the requested double-linkage case is presented here as **average (group-average) linkage**.

Worked Example 5A: Common 2D Dataset and Initial Matrix

The following same dataset will be used for single, average, and complete linkage. Use Euclidean distance.

Point	x	y
A	1	1
B	2	1
C	4	4
D	5	4
E	8	8

For example,

$$d(A,C) = \sqrt{(1-4)^2 + (1-4)^2} = \sqrt{18} \approx 4.243,$$
$$d(D,E) = \sqrt{(5-8)^2 + (4-8)^2} = \sqrt{25} = 5.$$

The complete initial Euclidean distance matrix is

	A	B	C	D	E
A	0	1.000	4.243	5.000	9.899
B	1.000	0	3.606	4.243	9.220
C	4.243	3.606	0	1.000	5.657
D	5.000	4.243	1.000	0	5.000
E	9.899	9.220	5.657	5.000	0

The smallest nonzero value is 1. There is a tie between (A, B) and (C, D). We merge A, B first and then C, D. This tie choice does not change the final calculations shown below.

Worked Example 5B: Single Linkage Using the 2D Dataset

Single linkage uses the minimum pairwise distance:

$$d_{\text{single}}(U,V) = \min_{u \in U, v \in V} d(u,v).$$

Step 1: Merge A and B at distance 1. Calculate the new distances:

$$d(AB,C) = \min\{d(A,C), d(B,C)\} = \min\{4.243, 3.606\} = 3.606,$$
$$d(AB,D) = \min\{5.000, 4.243\} = 4.243,$$
$$d(AB,E) = \min\{9.899, 9.220\} = 9.220.$$

Updated matrix after merging AB:

	AB	C	D	E
AB	0	3.606	4.243	9.220
C	3.606	0	1.000	5.657
D	4.243	1.000	0	5.000
E	9.220	5.657	5.000	0

Step 2: Merge C and D at distance 1.

$$\begin{aligned} d(AB, CD) &= \min\{d(A,C), d(A,D), d(B,C), d(B,D)\} \\ &= \min\{4.243, 5.000, 3.606, 4.243\} = 3.606, \\ d(CD, E) &= \min\{d(C,E), d(D,E)\} \\ &= \min\{5.657, 5.000\} = 5.000. \end{aligned}$$

Updated matrix after merging CD:

	AB	CD	E
AB	0	3.606	9.220
CD	3.606	0	5.000
E	9.220	5.000	0

Step 3: Merge AB and CD at distance 3.606. The remaining distance is

$$d(ABCD, E) = \min\{9.899, 9.220, 5.657, 5.000\} = 5.000.$$

Final two-cluster matrix:

	$ABCD$	E
$ABCD$	0	5.000
E	5.000	0

Single-link merge sequence:

$$(A,B)_{1.000}, \quad (C,D)_{1.000}, \quad (AB,CD)_{3.606}, \quad (ABCD,E)_{5.000}.$$

Worked Example 5C: Average Linkage (Requested Double Case)

Average linkage uses the mean of all cross-cluster pairwise distances:

$$d_{\text{average}}(U,V) = \frac{1}{|U||V|} \sum_{u \in U} \sum_{v \in V} d(u,v).$$

Step 1: Merge A and B at distance 1.

$$\begin{aligned} d(AB, C) &= \frac{4.243 + 3.606}{2} = 3.925, \\ d(AB, D) &= \frac{5.000 + 4.243}{2} = 4.622, \\ d(AB, E) &= \frac{9.899 + 9.220}{2} = 9.560. \end{aligned}$$

Updated matrix after merging AB:

	AB	C	D	E
AB	0	3.925	4.622	9.560
C	3.925	0	1.000	5.657
D	4.622	1.000	0	5.000
E	9.560	5.657	5.000	0

Step 2: Merge C and D at distance 1.

$$\begin{aligned} d(AB, CD) &= \frac{4.243 + 5.000 + 3.606 + 4.243}{4} \\ &= \frac{17.092}{4} = 4.273, \\ d(CD, E) &= \frac{5.657 + 5.000}{2} = 5.329. \end{aligned}$$

Updated matrix after merging CD:

	AB	CD	E
AB	0	4.273	9.560
CD	4.273	0	5.329
E	9.560	5.329	0

Step 3: Merge AB and CD at distance 4.273. The average distance from $ABCD$ to E is

$$\begin{aligned} d(ABCD, E) &= \frac{d(A, E) + d(B, E) + d(C, E) + d(D, E)}{4} \\ &= \frac{9.899 + 9.220 + 5.657 + 5.000}{4} \\ &= \frac{29.776}{4} = 7.444. \end{aligned}$$

Final two-cluster matrix:

	$ABCD$	E
$ABCD$	0	7.444
E	7.444	0

Average-link merge sequence:

$$(A, B)_{1.000}, \quad (C, D)_{1.000}, \quad (AB, CD)_{4.273}, \quad (ABCD, E)_{7.444}.$$

Worked Example 5D: Complete Linkage Using the 2D Dataset

Complete linkage uses the maximum pairwise distance:

$$d_{\text{complete}}(U, V) = \max_{u \in U, v \in V} d(u, v).$$

Step 1: Merge A and B at distance 1.

$$\begin{aligned} d(AB, C) &= \max\{4.243, 3.606\} = 4.243, \\ d(AB, D) &= \max\{5.000, 4.243\} = 5.000, \\ d(AB, E) &= \max\{9.899, 9.220\} = 9.899. \end{aligned}$$

Updated matrix after merging AB:

	AB	C	D	E
AB	0	4.243	5.000	9.899
C	4.243	0	1.000	5.657
D	5.000	1.000	0	5.000
E	9.899	5.657	5.000	0

Step 2: Merge C and D at distance 1.

$$d(AB, CD) = \max\{4.243, 5.000, 3.606, 4.243\} = 5.000,$$
$$d(CD, E) = \max\{5.657, 5.000\} = 5.657.$$

Updated matrix after merging CD:

	AB	CD	E
AB	0	5.000	9.899
CD	5.000	0	5.657
E	9.899	5.657	0

Step 3: Merge AB and CD at distance 5.000. The complete-link distance from $ABCD$ to E is

$$d(ABCD, E) = \max\{9.899, 9.220, 5.657, 5.000\} = 9.899.$$

Final two-cluster matrix:

	$ABCD$	E
$ABCD$	0	9.899
E	9.899	0

Complete-link merge sequence:

$$(A, B)_{1.000}, \quad (C, D)_{1.000}, \quad (AB, CD)_{5.000}, \quad (ABCD, E)_{9.899}.$$

Comparison of the Three Linkage Results

All three rules produce the same merge order for this dataset, but the merge heights differ.

Merge	Single	Average	Complete
A with B	1.000	1.000	1.000
C with D	1.000	1.000	1.000
AB with CD	3.606	4.273	5.000
$ABCD$ with E	5.000	7.444	9.899

Thus, single linkage gives the smallest inter-cluster distances, complete linkage gives the largest, and average linkage lies between them. Cutting each dendrogram immediately before its final merge gives

$$\boxed{\{A, B, C, D\} \quad \text{and} \quad \{E\}}.$$

Note

Single linkage can discover chain-shaped clusters but may suffer from chaining. Complete linkage tends to produce compact clusters. Average linkage is a compromise. Hierarchical merges are normally irreversible.

6 Density-Based Clustering: DBSCAN

Definition

DBSCAN uses two parameters:

- ε: neighborhood radius;
- MinPts: minimum number of points in the neighborhood, counting the point itself.

The ε-neighborhood of p is

$$N_\varepsilon(p) = \{q \in X : d(p, q) \le \varepsilon\}.$$

A point is a **core point** if $|N_\varepsilon(p)| \ge \text{MinPts}$. A non-core point inside a core point's neighborhood is a **border point**. A point that is neither core nor border is **noise**.

Worked Example 6: DBSCAN with Eleven Data Points

Use Euclidean distance, $\varepsilon = 1.5$, and MinPts $= 4$.

Point	x	y	Point	x	y
A	1	1	G	9	8
B	1	2	H	9	9
C	2	1	I	5	5
D	2	2	J	1	3
E	8	8	K	3	2
F	8	9			

Step 1: Find every ε-neighborhood.

Point	$N_{1.5}(p)$	Size	Preliminary type
A	$\{A, B, C, D\}$	4	Core
B	$\{A, B, C, D, J\}$	5	Core
C	$\{A, B, C, D, K\}$	5	Core
D	$\{A, B, C, D, J, K\}$	6	Core
E	$\{E, F, G, H\}$	4	Core
F	$\{E, F, G, H\}$	4	Core
G	$\{E, F, G, H\}$	4	Core
H	$\{E, F, G, H\}$	4	Core
I	$\{I\}$	1	Non-core
J	$\{B, D, J\}$	3	Non-core
K	$\{C, D, K\}$	3	Non-core

Step 2: Expand the first cluster. Start from core point A. Its neighborhood contains A, B, C, D. Because B, C, D are also core points, add all points in their neighborhoods. This adds J and K:

$$C_1 = \{A, B, C, D, J, K\}.$$

J and K are not core points, but they are within ε of core points, so both are border points.

Step 3: Expand the second cluster. Start from unvisited core point E. The mutually connected core points are

$$C_2 = \{E, F, G, H\}.$$

Step 4: Identify noise. Point I has only itself in its neighborhood and is not within the neighborhood of any core point. Therefore,

$$\text{Noise} = \{I\}.$$

Final classification table.

Point(s)	DBSCAN type	Final group	Reason
A, B, C, D	Core	C_1	Neighborhood size ≥ 4
J, K	Border	C_1	Reachable from core points
E, F, G, H	Core	C_2	Neighborhood size $= 4$
I	Noise	None	Not density-reachable

DBSCAN Result

$$C_1 = \{A, B, C, D, J, K\}, \qquad C_2 = \{E, F, G, H\}, \qquad \text{Noise} = \{I\}.$$

DBSCAN can find non-spherical clusters and explicitly detect noise, but a single $(\varepsilon, \text{MinPts})$ pair may struggle when clusters have very different densities.

7 Evaluation of Clustering Results

Definition

Clustering evaluation asks whether a partition is compact, well separated, stable, and meaningful.

- **Internal evaluation:** uses only the data and cluster assignments (SSE, silhouette, Davies–Bouldin index).
- **External evaluation:** compares clusters with known reference labels (purity, Rand index, adjusted Rand index, normalized mutual information).
- **Relative evaluation:** compares results across choices such as different k values or parameters.

For the internal examples, reuse the six-point K-Means result:

Point	x	y	Assigned cluster	Centroid
A	1	1	C_1	$(4/3, 4/3)$
B	1	2	C_1	$(4/3, 4/3)$
C	2	1	C_1	$(4/3, 4/3)$
D	8	8	C_2	$(25/3, 25/3)$
E	8	9	C_2	$(25/3, 25/3)$
F	9	8	C_2	$(25/3, 25/3)$

7.1 SSE: compactness

Key Formula

$$\text{SSE} = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2.$$

Smaller SSE means more compact clusters, but SSE always tends to decrease as k increases.

Worked Example 7A: Calculate SSE

The pointwise squared distances are:

Point	Cluster	Squared distance	Point	Cluster	Squared distance
A	C_1	$2/9$	D	C_2	$2/9$
B	C_1	$5/9$	E	C_2	$5/9$
C	C_1	$5/9$	F	C_2	$5/9$

Therefore,

$$\text{SSE} = \left(\frac{2}{9}+\frac{5}{9}+\frac{5}{9}\right)+\left(\frac{2}{9}+\frac{5}{9}+\frac{5}{9}\right) = \boxed{\frac{8}{3} \approx 2.667}.$$

7.2 Elbow method: selecting k

Worked Example 7B: Elbow Selection

Suppose repeated K-Means runs give the following complete comparison table.

k	SSE	Reduction from previous k
1	300	—
2	80	220
3	35	45
4	30	5
5	28	2

The improvement is large up to $k = 3$ and becomes small after $k = 3$. Thus the bend, or elbow, is at

$$\boxed{k = 3}.$$

The elbow method is visual and should be combined with other evidence when the bend is unclear.

7.3 Silhouette coefficient: cohesion and separation

Key Formula

For point i:

- $a(i)$ is the average distance from i to other points in its own cluster;
- $b(i)$ is the minimum average distance from i to any other cluster.

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i), b(i)\}}, \qquad -1 \le s(i) \le 1.$$

Values near 1 are good, values near 0 indicate overlap, and negative values suggest possible misassignment.

Worked Example 7C: Full Silhouette Calculation

For point A, its distances to B, C are both 1, so

$$a(A) = \frac{1+1}{2} = 1.$$

Its average distance to the other cluster is

$$b(A) = \frac{\sqrt{98} + \sqrt{113} + \sqrt{113}}{3} \approx 10.3866.$$

Therefore,

$$s(A) = \frac{10.3866 - 1}{10.3866} \approx 0.9037.$$

Repeating the same calculation for all six points gives:

Point	Cluster	$a(i)$	$b(i)$	$s(i)$
A	C_1	1.0000	10.3866	0.9037
B	C_1	1.2071	9.7063	0.8756
C	C_1	1.2071	9.7063	0.8756
D	C_2	1.0000	9.4462	0.8941
E	C_2	1.2071	10.1765	0.8814
F	C_2	1.2071	10.1765	0.8814

The mean silhouette is

$$\bar{s} = \frac{1}{6}\sum_{i=1}^{6} s(i) = \boxed{0.8853}.$$

This high positive value indicates compact and well-separated clusters.

7.4 Davies–Bouldin index

Key Formula

Let

$$S_i = \frac{1}{|C_i|}\sum_{x \in C_i} d(x, \mu_i), \qquad M_{ij} = d(\mu_i, \mu_j).$$

Then

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}, \qquad \text{DBI} = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} R_{ij}.$$

Lower DBI is better.

Worked Example 7D: Davies–Bouldin Index

For C_1, the Euclidean distances to $\mu_1 = (4/3, 4/3)$ are

$$\frac{\sqrt{2}}{3}, \quad \frac{\sqrt{5}}{3}, \quad \frac{\sqrt{5}}{3}.$$

Thus

$$S_1 = \frac{\sqrt{2} + 2\sqrt{5}}{9} \approx 0.6540.$$

By symmetry, $S_2 \approx 0.6540$. The centroid separation is

$$M_{12} = \sqrt{\left(\frac{25}{3} - \frac{4}{3}\right)^2 + \left(\frac{25}{3} - \frac{4}{3}\right)^2} = \sqrt{7^2 + 7^2} = \sqrt{98} \approx 9.8995.$$

For two clusters, both maximum ratios are the same:

$$\text{DBI} = \frac{S_1 + S_2}{M_{12}} = \frac{1.3080}{9.8995} = \boxed{0.1321}.$$

The low value agrees with the high silhouette score.

7.5 Purity: external evaluation

Key Formula

If reference class labels are available, purity is

$$\text{Purity}(\Omega, \mathcal{L}) = \frac{1}{n} \sum_{k} \max_{j} |C_k \cap L_j|.$$

It is the fraction of objects obtained by assigning each cluster to its majority class. Higher is better, but purity can be artificially increased by using too many clusters.

Worked Example 7E: Purity with Six Labeled Objects

Suppose the clustering and reference labels are:

Object	Cluster	True label	Object	Cluster	True label
A	C_1	X	D	C_2	Y
B	C_1	X	E	C_2	Y
C	C_1	X	F	C_2	X

The contingency table is:

	Label X	Label Y	Cluster size
C_1	3	0	3
C_2	1	2	3
Total	4	2	6

The majority count is 3 in C_1 and 2 in C_2. Therefore,

$$\text{Purity} = \frac{3+2}{6} = \boxed{\frac{5}{6} \approx 0.8333}.$$

7.6 Evaluation summary

Measure	Needs labels?	What it measures	Preferred value
SSE	No	Within-cluster squared compactness	Lower
Silhouette	No	Cohesion and separation together	Near 1
Davies–Bouldin	No	Within scatter relative to between separation	Lower
Purity	Yes	Agreement with majority reference class	Near 1
ARI/NMI	Yes	Label agreement corrected or normalized	Higher
Stability	No	Reproducibility under resampling/initialization	Higher

Table 2: Common clustering evaluation measures.

8 Comparison of the Clustering Methods

Method	Best data type	Representative	Main strength	Main limitation	Important parameters
K-Means	Numerical	Centroid	Fast; easy to interpret	Sensitive to scale, outliers, initialization; favors spherical clusters	k, initialization
K-Medoids	Numerical or dissimilarity data	Actual medoid	More robust to outliers; arbitrary distance matrix possible	More computationally expensive	k, distance measure
K-Modes	Categorical	Attribute-wise mode	Directly handles categorical variables	Tie handling and initialization can affect result	k, initialization
Hierarchical	Numerical or dissimilarity data	Dendrogram	Reveals nested structure; k not needed initially	Greedy merges/splits; linkage sensitive	Distance, linkage, cut height
DBSCAN	Usually numerical/spatial	Dense connected region	Finds arbitrary shapes and noise; no k required	Difficult with varying density or high dimension	ε, MinPts

Table 3: Method-selection guide.

Practical Selection Rules

- Use **K-Means** for large numerical datasets with compact clusters.
- Use **K-Medoids** when outliers are important or representatives must be actual observations.
- Use **K-Modes** for categorical attributes.
- Use **hierarchical clustering** when nested structure and a dendrogram are valuable.
- Use **DBSCAN** when clusters may have irregular shapes and noise should be identified.
- Always scale numerical features when their units differ substantially, and evaluate more than one parameter choice.

9 Short Practice Problems

1. For points $A(0,0)$, $B(0,1)$, $C(1,0)$, $D(6,6)$, $E(6,7)$, and $F(7,6)$, run one K-Means assignment step using centroids $(0,0)$ and $(6,6)$.

2. For 2D points $A(1,1)$, $B(2,1)$, $C(2,2)$, $D(8,8)$, $E(9,8)$, and $F(12,12)$, compute the Manhattan K-Medoids assignment cost using medoids B and E.
3. With simple matching dissimilarity, find the distance between the two categorical objects

   (Red, Small, Circle) and (Blue, Small, Square).

4. For 2D points $P(0,0)$, $Q(0,1)$, $R(4,4)$, $S(5,4)$, and $T(9,9)$, list the first two single-link agglomerative merges.
5. In DBSCAN, let $\varepsilon = 2$ and MinPts $= 4$. A point has three other points plus itself in its neighborhood. Is it core, border, or noise?
6. A point has $a(i) = 2$ and $b(i) = 8$. Calculate its silhouette coefficient.

Answer Check

1. $\{A, B, C\}$ go to $(0,0)$ and $\{D, E, F\}$ go to $(6,6)$.
2. Costs: $1, 0, 1, 1, 0, 7$; total 10.
3. Two mismatches, so $d_{\text{SM}} = 2$.
4. Merge P with Q at distance 1, and R with S at distance 1 (either tied merge may be listed first).
5. Core, because its neighborhood size is 4 when the point itself is counted.
6. $s(i) = (8-2)/8 = 0.75$.

10 Key Takeaways

1. Clustering quality depends on data representation, scaling, and the definition of similarity.
2. K-Means uses centroids and squared Euclidean error; K-Medoids uses observed representatives; K-Modes uses categorical modes.
3. Hierarchical clustering produces a dendrogram, and the linkage rule controls the meaning of "closest clusters."
4. DBSCAN distinguishes core, border, and noise points through density connectivity.
5. No single evaluation score is sufficient. Combine compactness, separation, stability, domain usefulness, and external labels when available.

Prepared By
Md. Atikuzzaman
Lecturer, Department of Computer Science and Engineering
Green University of Bangladesh
Email: atik@cse.green.edu.bd