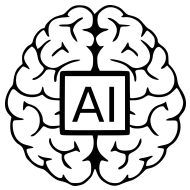
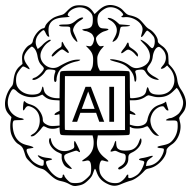


Clustering Techniques in Machine Learning

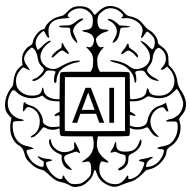
Author Name

Institute/Organization

December 17, 2024



Outline

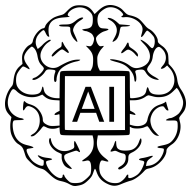


Definition

Clustering: Grouping unlabeled data points into subsets (clusters) where points in the same cluster are more similar to each other than to those in other clusters.

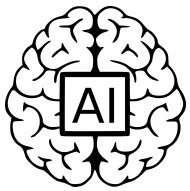
Mathematical Setup: Given $X = \{x_1, x_2, \ldots, x_n\}$ with each $x_i \in \mathbb{R}^d$, clustering aims to partition X into k disjoint subsets (clusters) $C = \{C_1, C_2, \ldots, C_k\}$ such that:

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



Applications

- Customer segmentation
- Image compression and object grouping
- Document/topic clustering in NLP
- Bioinformatics (gene expression data analysis)



Common Metrics

Euclidean Distance (L2):

$$d(x, y) = \sqrt{\sum_{m=1}^{d} (x_m - y_m)^2}$$

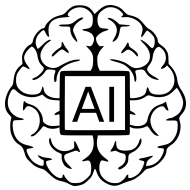
Manhattan Distance (L1):

$$d(x, y) = \sum_{m=1}^{d} |x_m - y_m|$$

Cosine Similarity:

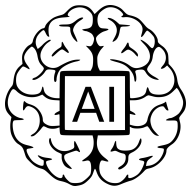
$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

Note: Choice of metric influences clustering results, especially in high dimensions.



Overview of Clustering Methods

Method Type	**Examples**
Centroid-Based	K-means, K-medoids
Hierarchical	Agglomerative, Divisive
Density-Based	DBSCAN, OPTICS
Distribution-Based	Gaussian Mixtures (GMM)
Graph-Based	Spectral Clustering
Deep/Embedding-Based	DEC, VAE-based methods



K-Means

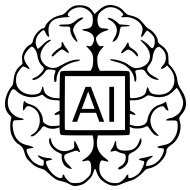
Objective: Minimize the Within-Cluster Sum of Squares (WCSS):

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2,$$

where μ_j is the centroid of cluster C_j.

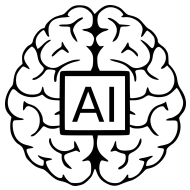
Algorithm:

1. Initialize k centroids μ_j.
2. Assign each point x_i to the closest centroid:
$$C_j^{(t)} = \{x_i : \|x_i - \mu_j^{(t)}\| \leq \|x_i - \mu_l^{(t)}\|, \forall l\}.$$
3. Update centroids:
$$\mu_j^{(t+1)} = \frac{1}{|C_j^{(t)}|} \sum_{x_i \in C_j^{(t)}} x_i.$$



K-Means Pros/Cons

- **Pros:** Simple, fast, widely used.
- **Cons:** Sensitive to initialization, primarily finds spherical clusters.



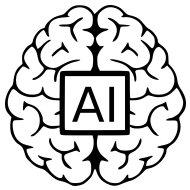
K-Medoids

Similar to K-means, but uses actual data points as centers (medoids).

Update Step:

$$\tilde{\mu}_j = \arg\min_{x \in C_j} \sum_{x_i \in C_j} d(x_i, x)$$

Pros: More robust to outliers than K-means.



Hierarchical Clustering

Concept: Build a hierarchy of clusters without specifying k upfront.

Agglomerative Clustering:

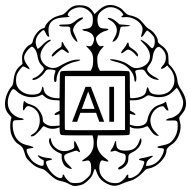
- Start with each point as its own cluster.
- Iteratively merge the two closest clusters until one cluster remains.

Linkage Methods:

$$\text{Single: } d(C_a, C_b) = \min_{x \in C_a, y \in C_b} d(x, y)$$

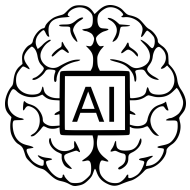
$$\text{Complete: } d(C_a, C_b) = \max_{x \in C_a, y \in C_b} d(x, y)$$

$$\text{Average: } d(C_a, C_b) = \frac{1}{|C_a||C_b|} \sum_{x \in C_a} \sum_{y \in C_b} d(x, y)$$



Hierarchical Clustering Pros/Cons

- **Pros:** No need to pre-specify k, interpretable dendrogram.
- **Cons:** High complexity, no backtracking once merged.



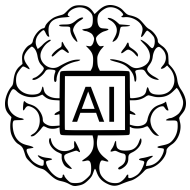
DBSCAN Concept

Idea: Identifies "core" points in dense regions.

Parameters:

- ϵ: Neighborhood radius
- MinPts: Minimum points within ϵ for a core point

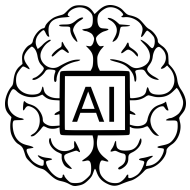
Core Point: If at least MinPts are within ϵ of it. **Reachability:** Points reachable via a chain of core points are in the same cluster. Non-reachable points are noise.



DBSCAN Algorithm Steps

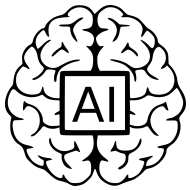
1. Identify core points.
2. Form clusters by connecting core points within ϵ.
3. Assign non-core points to clusters if within ϵ of a core point.
4. Unreachable points are noise.

- **Pros:** Detects arbitrarily shaped clusters, finds outliers.
- **Cons:** Sensitive to ϵ and MinPts, struggles with varying density.



Density-Based Variants

OPTICS: Handles varying densities, outputs an ordering.
HDBSCAN: Hierarchical density-based clustering, no fixed ϵ needed.



GMM Concept

Assume data from a mixture of k Gaussians:

$$p(x) = \sum_{j=1}^{k} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j).$$

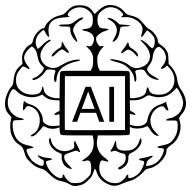
EM Algorithm:

E-step:

$$\gamma_{ij} = \frac{\pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}{\sum_{l=1}^{k} \pi_l \mathcal{N}(x_i|\mu_l, \Sigma_l)}$$

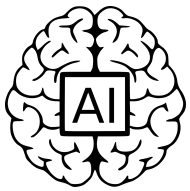
M-step:

$$\pi_j := \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij}, \quad \mu_j := \frac{\sum_{i=1}^{n} \gamma_{ij} x_i}{\sum_{i=1}^{n} \gamma_{ij}}, \quad \Sigma_j := \frac{\sum_{i=1}^{n} \gamma_{ij}(x_i - \mu_j)(x$$



GMM Pros/Cons

- **Pros:** Can model complex cluster shapes, probabilistic interpretation.
- **Cons:** May converge to local maxima, assumes Gaussianity.



Spectral Clustering Idea

Idea: Use eigenvectors of a similarity graph's Laplacian matrix to cluster points.

Steps:

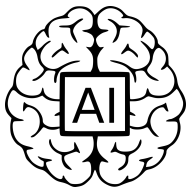
1. Construct similarity graph W:

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right).$$

2. Compute Laplacian $L = D - W$, where $D_{ii} = \sum_j w_{ij}$.
3. Compute eigenvectors of L.
4. Use top k eigenvectors as features and cluster (e.g., with K-means).

Pros: Finds non-linearly separable clusters.

Cons: Eigen-decomposition can be costly, parameter sensitive.



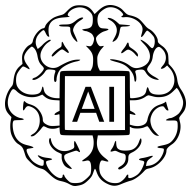
Deep Embedding Clustering (DEC)

Idea: Modern methods integrate deep learning to produce suitable embeddings for clustering.
Jointly optimize a reconstruction loss (via autoencoder) and a clustering loss (e.g., Kullback–Leibler divergence).

Loss:

$$L = L_r + \lambda L_c$$

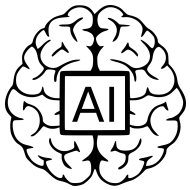
- L_r: Reconstruction loss, $\|X - \hat{X}\|_F^2$
- L_c: KL divergence between soft assignments Q and a target distribution P



Internal Validation

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

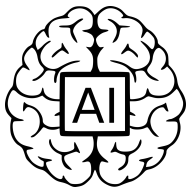
- $a(i)$: mean intra-cluster distance
- $b(i)$: min mean distance to any other cluster



Internal Validation

Davies-Bouldin Index: $DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{\sigma_i + \sigma_j}{\|\mu_i - \mu_j\|}$, where:

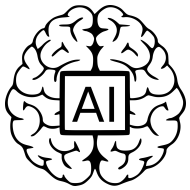
- k is the number of clusters,
- σ_i is the average distance of all points in cluster i to the centroid of cluster i,
- μ_i is the centroid of cluster i,
- $\|\mu_i - \mu_j\|$ is the distance between the centroids of clusters i and j.



Validation Summary

Silhouette Score:

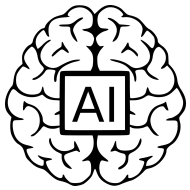
- **Value Range:** [-1, 1]
- **Interpretation:**
  - ≈ 1.0: Point is well-clustered
  - ≈ 0.0: Point is on cluster boundary
  - ≈ -1.0: Point might be in wrong cluster
- **Best Use Case:** Evaluating individual point placement
- **Complexity:** $O(n^2)$



Validation Summary

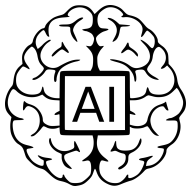
Davies-Bouldin Index:

- **Value Range:** $[0, \infty)$
- **Interpretation:**
  - Close to 0: Better clustering
  - Larger values: Worse clustering
- **Best Use Case:** Comparing different clustering results
- **Complexity:** $O(k^2)$ where k = number of clusters



Validation Summary

Aspect	Silhouette Score	Davies-Bouldin Index
Range	[-1, 1]	[0, ∞)
Optimal Value	1	0
Measures	Point-level cohesion and separation	Cluster-level separation
Complexity	$O(n^2)$	$O(k^2)$
Best Use Case	Evaluating individual point placement	Comparing different clustering results



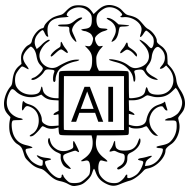
External Validation

With ground truth available:

Rand Index (RI):

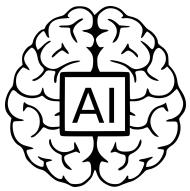
$$RI = \frac{TP + TN}{TP + TN + FP + FN}.$$

Adjusted Rand Index (ARI): Adjusts RI for chance.



Complexity Overview

Algorithm	Complexity	Notes
K-means	$O(nkd)$ per iteration	Fast, simple
Hierarchical	$O(n^3)$ (naive)	Often for smaller n
DBSCAN	$O(n \log n)$ to $O(n^2)$	Depends on indexin
GMM (EM)	$O(nkd^2)$ per iteration	Covariance matters
Spectral Clustering	$O(n^3)$	Eigen-decompositio

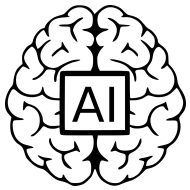


Advanced Techniques

- **Dimensionality Reduction (PCA, t-SNE, UMAP):** Handle high-dimensional data.
- **Kernel Methods:** Clustering in nonlinear feature spaces.
- **Fuzzy Clustering (Fuzzy C-means):** Soft membership:

$$J_m = \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij}^m \|x_i - \mu_j\|^2.$$

- **Semi-Supervised Clustering:** Must-link/cannot-link constraints guide clustering.



Conclusion

- Wide range of clustering algorithms available.
- Choice depends on data shape, scale, and domain knowledge.
- Use validation metrics and possibly dimensionality reduction.
- Experimentation is key to discovering meaningful structure.