Unsupervised Learning

Fundamentals of Machine Learning

Xufeng Zhang

Inria
xufeng.zhang@inria.fr

December 11, 2025

Outline

1. Machine Learning Landscape
2. Clustering Overview
3. Hierarchical Clustering
4. K-means Clustering
5. Gaussian Mixture Models
6. Clustering Evaluation
7. Dimensionality Reduction and PCA
8. Applications in Biology and Medicine
9. Summary

Machine Learning Methods

- **Supervised learning**
  - Labeled data: input–output pairs $(\mathbf{x}_i, y_i)$
  - Learn a mapping $f : \mathcal{X} \to \mathcal{Y}$
  - Tasks: classification, regression
- **Unsupervised learning**
  - Only inputs $\mathbf{x}_i$ are observed
  - Discover hidden structure in the data
  - Tasks: clustering, dimensionality reduction, density estimation

Supervised vs. Unsupervised Learning

Supervised

- Ground-truth labels available
- Objective: minimize prediction error
- Evaluation: accuracy, MSE, ROC AUC, ...

Unsupervised

- No labels
- Objective: reveal structure and patterns
- Evaluation: internal and external clustering metrics

Typical Unsupervised Learning Tasks

- **Clustering**
  - Group similar observations together
  - Examples: customer segmentation, document clustering
- **Dimensionality reduction**
  - Find low-dimensional representation of data
  - Examples: visualization, noise reduction, feature extraction
- **Density estimation**
  - Estimate probability distribution generating the data
  - Examples: anomaly detection, generative modeling

Notation

- Dataset of n observations in d dimensions:

$$X = \begin{bmatrix} - & \mathbf{x}_1^\top & - \\ - & \mathbf{x}_2^\top & - \\ & \vdots & \\ - & \mathbf{x}_n^\top & - \end{bmatrix} \in \mathbb{R}^{n\times d}.$$

- Each observation:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^\top \in \mathbb{R}^d.$$

- Clustering assigns a *cluster label* $z_i \in \{1, \ldots, K\}$ to each $\mathbf{x}_i$.
- Dimensionality reduction maps $\mathbf{x}_i \mapsto \mathbf{y}_i \in \mathbb{R}^m$ with $m \ll d$.

Unsupervised Learning Pipeline

1. Data preprocessing
   - Handle missing values
   - Standardize or normalize features
2. Choice of distance / similarity measure
3. Selection of algorithm
   - K-means, hierarchical clustering, GMM, PCA, ...
4. Hyperparameter selection
   - Number of clusters K
   - Choice of linkage, covariance type, etc.
5. Model evaluation and interpretation

Distance and Similarity Measures

- Euclidean distance:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$$

- Manhattan distance:

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

- Cosine similarity:

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

- Choice of distance has strong impact on clustering results.

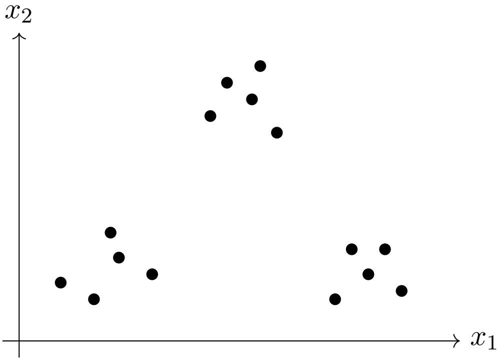
What is Clustering?

- Partition data into groups of similar objects.
- Objective: within a cluster, points should be close; between clusters, far apart.
- No class labels, so notions of similarity and cluster quality are *model assumptions*.
- Common approaches:
    - Hierarchical clustering
    - K-means
    - Gaussian Mixture Models (GMM)

Families of Clustering Algorithms

- **Partition-based**
  - Directly partition data into K clusters
  - Example: K-means
- **Hierarchical**
  - Build nested sequence of partitions
  - Example: agglomerative clustering
- **Model-based**
  - Assume generative probabilistic model
  - Example: Gaussian mixture models

Toy Example of Clustering (2D)



- Three well-separated clusters.
- Many algorithms can recover this partition easily.

Challenges in Clustering

- Unknown number of clusters.
- Scale and units of features.
- High-dimensional spaces (curse of dimensionality).
- Non-spherical or non-convex cluster shapes.
- Presence of noise and outliers.

Agglomerative Hierarchical Clustering

- **Bottom-up** approach:
  1. Start with n clusters, each point in its own cluster.
  2. Iteratively merge the two closest clusters.
  3. Continue until a single cluster containing all points is obtained.
- Output can be visualized as a **dendrogram**.
- No need to pre-specify the number of clusters.

Linkage Criteria

Consider clusters A and B.

- **Single linkage**

$$d_{\mathsf{single}}(A, B) = \min_{\mathbf{x}\in A, \mathbf{y}\in B} d(\mathbf{x}, \mathbf{y})$$

- **Complete linkage**

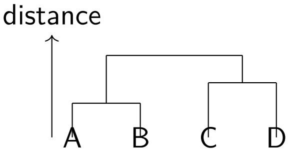
$$d_{\mathsf{complete}}(A, B) = \max_{\mathbf{x}\in A, \mathbf{y}\in B} d(\mathbf{x}, \mathbf{y})$$

- **Average linkage**

$$d_{\mathsf{average}}(A, B) = \frac{1}{|A||B|} \sum_{\mathbf{x}\in A} \sum_{\mathbf{y}\in B} d(\mathbf{x}, \mathbf{y})$$

Choice of linkage heavily affects resulting hierarchy.

Dendrogram Illustration



- Vertical axis encodes distance at which clusters are merged.
- Cutting the dendrogram at a chosen height yields a clustering.

City Distance Example

- Example with six US cities:
  - Boston, New York, Chicago, Denver, San Francisco, Seattle.
- Distances in miles form a 6×6 matrix.
- Hierarchical clustering progressively merges closest cities (e.g., Boston with New York, San Francisco with Seattle).
- Dendrogram reveals geographic structure of the US map.

Advantages and Drawbacks

Advantages

- Deterministic (given distance and linkage).
- Flexible choice of linkage criterion.
- Dendrogram provides multi-scale view of data.

Drawbacks

- Naïve implementations have $O(n^3)$ time complexity.
- Once clusters are merged, they cannot be split again.
- Sensitive to noisy points and outliers.

K-means Algorithm: Intuition

- Partition data into K clusters.
- Each cluster represented by its **centroid** (mean of points).
- Objective: minimize sum of squared distances between each point and its cluster centroid.
- Works well when clusters are roughly spherical and of similar size.

K-means Objective Function

- Cluster centroids: $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$.
- Assignment variables:

$$z_{ik} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is assigned to cluster } k, \\ 0 & \text{otherwise.} \end{cases}$$

- Objective:

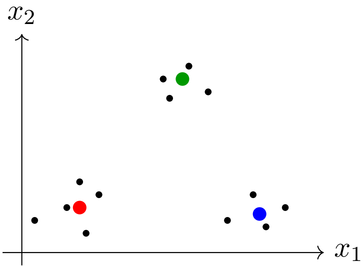
$$\min_{\{z_{ik}\},\{\boldsymbol{\mu}_k\}} \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2.$$

- Alternating minimization leads to the standard K-means algorithm.

K-means Algorithm

1. Choose the number of clusters K.
2. Initialize K centroids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ (often randomly).
3. **Repeat** until convergence:
    1. **Assignment step:** assign each $\mathbf{x}_i$ to the closest centroid.
    2. **Update step:** recompute each centroid as the mean of points currently assigned to it.
4. Converges when assignments stop changing or objective reduction is negligible.index=4

K-means: Visual Example



- Colors indicate different clusters.
- Stars show current centroids.

Initialization Matters

- Random initialization may lead to poor local minima.
- Different runs can give different results.
- Popular solution: **K-means++** initialization
  - Choose first centroid uniformly at random.
  - Choose subsequent centroids with probability proportional to squared distance from closest existing centroid.
  - Encourages spread-out initial centroids.
- Run K-means multiple times with different seeds and keep the best solution.

Choosing the Number of Clusters K

- No universal rule; several heuristics:
- **Elbow method**
  - Plot within-cluster sum of squares versus K.
  - Look for “elbow” where marginal gain drops.
- **Silhouette score**
  - Measures how similar an object is to its own cluster vs. other clusters.
  - Average silhouette over all points for different K.
- Domain knowledge often crucial.

Advantages and Limitations of K-means

Advantages

- Simple and easy to implement.
- Fast: each iteration is linear in $n \times d$.
- Scales well to large datasets.

Limitations

- Finds only local optima.
- Sensitive to initialization and the choice of K.
- Assumes spherical, equally sized clusters.
- Sensitive to outliers.

Motivation for Gaussian Mixture Models

- K-means provides hard assignments: each point belongs to exactly one cluster.
- Many applications require **soft assignments** (probabilities).
- Gaussian Mixture Models (GMM) generalize K-means:
  - Each cluster is a Gaussian distribution.
  - Data are generated from a mixture of these Gaussians.
- Clusters can be ellipsoidal, not necessarily spherical.

Gaussian Mixture Model

- Mixture of K Gaussian components:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k),$$

  where
  - $\pi_k \geq 0$, $\sum_k \pi_k = 1$ (mixture weights),
  - $\boldsymbol{\mu}_k$ mean vector,
  - Σ_k covariance matrix.
- Latent variable z_i indicates component membership.
- Posterior $p(z_i = k \mid \mathbf{x}_i)$ gives responsibility of each component.

Expectation-Maximization for GMM

- Parameters: $\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}_{k=1}^{K}$.
- Direct maximization of likelihood is difficult.
- Use **Expectation–Maximization (EM)**:
  1. Initialize parameters.
  2. **E-step:** compute responsibilities

$$\gamma_{ik} = p(z_i = k \mid \mathbf{x}_i) = \frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j)}.$$

  3. **M-step:** update parameters using γ_{ik}.
  4. Repeat until convergence.

GMM: M-step Updates

- Effective number of points in cluster k:

$$N_k = \sum_{i=1}^{n} \gamma_{ik}.$$

- Updated mixture weights:

$$\pi_k^{\text{new}} = \frac{N_k}{n}.$$

- Updated means:

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{ik} \mathbf{x}_i.$$

- Updated covariances:

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_i - \boldsymbol{\mu}_k^{\text{new}})^\top.$$

GMM Examples

- GMM can fit complex shapes by combining several Gaussian components.
- For non-linearly separable datasets (e.g., interleaving moons), GMM often outperforms K-means.
- Soft cluster assignments useful for:
  - Anomaly detection (low likelihood points).
  - Mixed-membership models.

GMM: Pros and Cons

Advantages

- Generalization of K-means with soft assignments.
- Flexible covariance structure (spherical, diagonal, full).
- Probabilistic interpretation; can compute likelihood.

Drawbacks

- Only local optima (like K-means).
- Not ideal for very complex or non-Gaussian cluster shapes.
- Choosing number of components remains challenging.

Why is Unsupervised Evaluation Hard?

- No ground-truth labels available in general.
- Many reasonable clusterings may exist.
- Evaluation often uses:
  - **Internal** metrics: use only data and cluster labels.
  - **External** metrics: require true labels (when available).
- Qualitative assessment by domain experts is very important.

Internal Metric: Silhouette Coefficient

For each point i:

- $a(i) =$ average distance to points in the same cluster.
- $b(i) =$ minimum average distance to points in other clusters.
- Silhouette:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \in [-1, 1].$$

- $s(i) \approx 1$: well-clustered.
- $s(i) \approx 0$: on the border between clusters.
- $s(i) < 0$: possibly misclassified.

External Metrics

Assume true labels y_i are known.

- **Adjusted Rand Index (ARI)**
  - Measures similarity between two partitions.
  - Corrected for chance; ARI $=1$ indicates perfect match.
- **Adjusted Mutual Information (AMI)**
  - Based on information theory.
  - Also adjusted for chance.
- **Homogeneity, Completeness, V-measure**
  - Homogeneity: each cluster contains only members of a single class.
  - Completeness: all members of a given class are assigned to the same cluster.
  - V-measure: harmonic mean of homogeneity and completeness.

Hyperparameter Tuning via Curves

- Example: K-means with different K.
- Plot *within-cluster sum of squares* (inertia) vs. K.
- Choose K around the elbow, where improvement slows down.
- Similar curves can be drawn for silhouette score, ARI, etc.

Why Dimensionality Reduction?

- High-dimensional data are common (e.g., genomics, text, images).
- Challenges:
  - Distance measures become less informative.
  - Computation becomes expensive.
- Dimensionality reduction:
  - Compress data while retaining important information.
  - Useful for visualization (2D/3D).
  - Preprocessing step before clustering.

PCA: High-Level Idea

- Principal Component Analysis (PCA):
  - Find orthogonal directions of maximum variance.
  - Project data onto first few principal components.
- Each principal component is a linear combination of original features.
- PCA handles multicollinearity and noisy measurements.

PCA: Geometric View

- Consider centered data in 3D.
- PCA finds a line (PC1) that best fits data in least-squares sense.
- Second component (PC2) is orthogonal to PC1 and captures remaining variance.
- Together, PC1 and PC2 define a plane approximating the data.

Mean Centering

1. Compute empirical mean
$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i.$$
2. Subtract mean from each data point:
$$\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}.$$
3. Centered data matrix $\tilde{X}$ has mean zero in each dimension.

Mean centering is essential before computing principal components.

Covariance Matrix

- Covariance matrix of centered data:

$$\Sigma = \frac{1}{n-1} \tilde{X}^\top \tilde{X} \in \mathbb{R}^{d \times d}.$$

- Diagonal entries: variances of each feature.
- Off-diagonal entries: pairwise covariances.
- PCA directions are eigenvectors of Σ corresponding to largest eigenvalues.

Eigenvalues and Eigenvectors

- Solve
$$\Sigma \mathbf{v}_j = \lambda_j \mathbf{v}_j,$$
where λ_j is eigenvalue and $\mathbf{v}_j$ eigenvector.
- Eigenvector with largest eigenvalue gives first principal component.
- Second principal component is eigenvector with second largest eigenvalue, etc.
- Eigenvalues indicate amount of variance explained by each component.

Explained Variance

- Total variance: $\sum_{j=1}^{d} \lambda_j$.
- Proportion of variance explained by first m components:

$$\mathsf{PVE}(m) = \frac{\sum_{j=1}^{m} \lambda_j}{\sum_{j=1}^{d} \lambda_j}.$$

- Plotting PVE vs. m helps decide number of components (scree plot).
- Often keep components explaining, e.g., 90%–95% of variance.

PCA Algorithm Summary

1. Standardize features to zero mean and unit variance (optional but common).
2. Compute covariance matrix of standardized data.
3. Perform eigen-decomposition (or SVD) of covariance matrix.
4. Sort eigenvectors by decreasing eigenvalues.
5. Select top m eigenvectors to form feature matrix W.
6. Project data: $\mathbf{y}_i = W^\top \tilde{\mathbf{x}}_i$.

PCA in Practice

- Commonly used as preprocessing for:
  - Clustering (K-means, GMM).
  - Visualization of high-dimensional data.
  - Regression or classification with many correlated features.
- In biology, PCA is used to summarize gene expression profiles and to reduce noise in high-throughput experiments.

Examples of ML in Biology and Medicine

- **Decision Trees / Random Forests**
  - Risk prediction (e.g., cardiovascular disease).
  - Can include principal components as features.
- **K-Nearest Neighbors**
  - Classify cells from new samples based on known cell types.
- **Linear Models**
  - Generalized Linear Models (GLMs) for risk prediction and survival analysis.
  - Linear regression for gene expression analysis.

Unsupervised Methods in Bio/Med

- **K-means**
  - Clustering of single-cell RNA-seq data after dimensionality reduction (e.g., PCA, UMAP).
  - Personalized medicine: identify patient subgroups.
- **Hierarchical clustering**
  - Construction of phylogenetic trees in evolutionary biology.
- **Deep learning**
  - Protein folding prediction.
- **SVM and imaging**
  - Tumor detection from medical images.

Example Pipeline: scRNA-seq Analysis

1. Preprocess counts (normalization, log transform).
2. Apply PCA to reduce dimensionality.
3. Optionally apply nonlinear reduction (UMAP or t-SNE).
4. Cluster cells using K-means or graph-based methods.
5. Interpret clusters using marker genes and biological knowledge.

Key Takeaways

- Unsupervised learning discovers structure in unlabeled data.
- Hierarchical clustering builds a dendrogram of nested clusters.
- K-means is simple and fast but assumes spherical clusters and needs K.
- Gaussian Mixture Models extend K-means with soft probabilistic assignments.
- Evaluation of clustering uses internal and external metrics.
- PCA is a powerful tool for dimensionality reduction and visualization.

Sources

- Lecture slides: *Fundamentals of Machine Learning* (Unsupervised Learning).
- Additional standard textbooks and online resources on machine learning and statistics.

Questions?