02-620 Week 5

Machine Learning for Scientists

Aidan Jan

February 13, 2026

Random Forests

Decision trees are high-variance classifiers, but small changes in the training data can lead to very different trees. **Solution: Random Forests**.

- Build a collection of decision trees trained on different subsets of the data
- Classify a test sample by majority vote across all trees in the forest

Procedure:

For $b = 1, \ldots, B$:

- Draw a bootstrap sample of size N from the training data
- Train a decision tree T_b on the bootstrap sample by recursively repeating, at each node:
  - Randomly select m attributes from the D available attributes
  - Choose the best split among the selected attributes
  - Split the node into two child nodes
  - Continue until the minimum node size n_{min} is reached
- Output the ensemble of trees $\{T_b\}_{b=1}^{B}$ and classify new samples by **majority vote**

Random Forest on Similated Data

- Random forest with 500 trees.
  - RF-1: depth 1 trees
  - RF-3: depth 3 trees
- Smaller trees perform well

Clustering

Supervised vs. Unsupervised Learning

Supervised learning
Learning to predict label y_i from features x_i using labeled data (X, y)

- **Regression:** $y_i \in \mathbb{R}$. E.g., linear regression, etc.
- **Classification:** $y_i \in \{0, 1\}$. E.g., Naive bayes classifier, logistic regression, decision trees, etc.

- **Models:** deterministic $y_i = f(x_i; \theta)$ or probabilistic $P(Y_i|X_i = x_i; \theta)$

Unsupervised learning
Learning structure or patterns from unlabeled data X, e.g., clustering, dimensionality reduction, etc.

- Models: deterministic $f(x_i; \theta)$ or probabilistic $P(X_i; \theta)$

Clustering is an unsupervised learning method where we try to find clusters of data points, or group similar data points together.

- We want high intra-cluster similarity
- low inter-cluster similarity
- Finding natural groupings among objects

Clustering Methods

- Non-probabilistic methods
  - Hierarchical clustering
  - K-means algorithm
- Probabilistic method
  - (Gaussian) Mixture Model

We will also discuss **dimensionality reduction**, another unsupervised learning method later in the course.

What is similarity?

From Webster's Dictionary: "The quality or state of being similar; likeness; resemblance; as, a similarity of features"

- It turns out that similarity is hard to define, but we know it when we see it.
- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

Distance Measures

- Suppose two data points $x, y \in \mathbb{R}^D$ (D features)

$$y = (y_1, \ldots, y_D)$$
$$x = (x_1, \ldots, x_D)$$

- Euclidean distance (L2 norm) (dissimilarity measure)

$$D(x, y) = \sqrt{\sum_{j=1}^{D} (x_j - y_j)^2}$$

- Correlation coefficient (similarity measure)

$$s(x, y) = \frac{1}{\sigma_x \sigma_y} \sum_{j=1}^{D} (x_j - \mu_x)(y_j - \mu_y)$$

where μ_x, μ_y, σ_x, σ_y are means and standard deviations across features.

General Distance Measures

Definitions: Let x and y be objects from a universe. The distance (dissimilarity) is a real-valued function $D(x, y)$, **typically** satisfying

- Distance from a point to itself is zero: $D(x, x) = 0$
- Positivity: if $x \neq y$, then $D(x, y) > 0$
- Symmetry: $D(x, y) = D(y, x)$
- Triangle inequality: $D(x, z) \leq D(x, y) + D(y, z)$

Overview

- K-means clustering: Construct partitions into K clusters and evaluate them by a centroid-based objective function
- Hierarchical clustering: Create a hierarchical decomposition of the set of objects using a linkage criterion

K-means Clustering

Data:

- N data points with D features

$$x_1, \cdots, x_N \in \mathbb{R}^D \quad \text{matrix form } X \in \mathbb{R}^{N \times D}$$

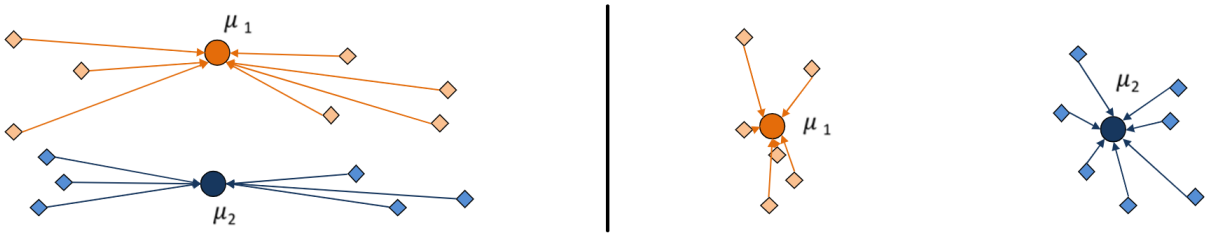
Learning objective: Partition the data points into K sets $S_1, \ldots, S_k$ to minimize the within-cluster squared Euclidean distance

$$\arg\min_{S_1,\ldots,S_k} \sum_{k=1}^{K} \sum_{i \in S_k} \|x_i - \mu_k\|^2$$

where $\mu_k = \frac{1}{|S_k|} \sum_{i \in S_k} x_i$ is the centroid of cluster k

K-means Clustering: Optimization

Not all clusters are good. In the below example, the right clustering is better than the left clustering.



How do we solve this optimization problem?

- Exact solution: exhaustive search over partitions (NP-hard)
- Practical solution: K-means (Lloyd's algorithm), a fast heuristic converging to a local minimum

Lloyd's Algorithm

- **Input:** data $x_1, \cdots, x_N \in \mathbb{R}^D$; hyperparameter K
- **Initialize:** cluster center $\mu_1, \ldots, \mu_K$, randomly
- **Iterate until convergence** (no chnages in assignments a_i)
  - **Assign** each data point to its closest cluster center

$$a_i \leftarrow \underset{k \in \{1,\ldots,K\}}{\arg\min} \|x_i - \mu_k\|_2^2, \quad \text{for } i = 1, \ldots, N$$

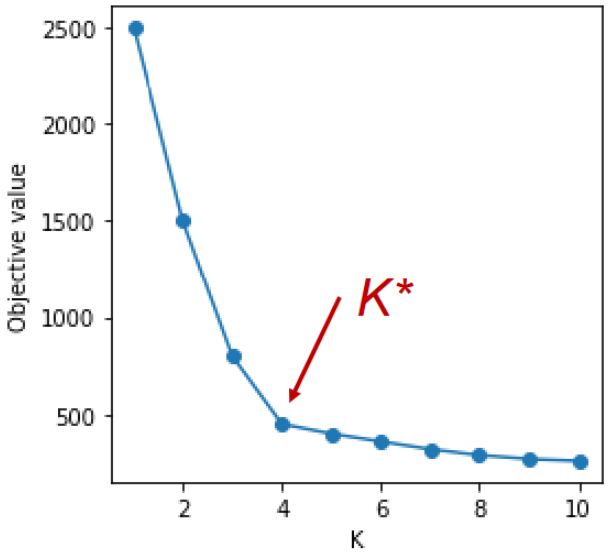
  - **Update** cluster centers based on the current assignments

$$\mu_k \leftarrow \frac{\sum_{i=1}^N \mathbb{1}(a_i = k) x_i}{\sum_{i=1}^N \mathbb{1}(a_i = k)}, \quad \text{for } k = 1, \ldots, K$$

Choosing K

How do we choose the number of clusters K?

- In general, this is an unsolved problem. However, many approximate methods exist.
- **Heuristic (elbow method):** Run cluster with $K = 1, 2, 3, \ldots$, and choose the value of K at which improvements in the objective begin to slow down



Hierarchical Clustering

Create a hierarchical decomposition of the set of objects using a linkage criterion

- **Bottom-up (agglomerative):**
  - Start with each point as its own cluster
  - Repeatedly **merge** the closest clusters
  - Continue until one cluster remains
- **Top-down (divisive):**

- Start with all points in one cluster
- Repeatedly **split** clusters
- Continue until each point is its own cluster (or stop early)

Cluster assignments are obtained by "cutting" the tree at a chosen level.

Bottom-up hierarchical clustering

We begin with a distance matrix containing the pairwise distances between all objects in the dataset.

	A	B	C	D	E
A	0	8	8	7	7
B		0	2	4	4
C			0	3	3
D				0	1
E					0

- Start with each point as its own cluster
- Repeatedly merge the closest clusters
- Continue until one cluster remains

In this example, we will first merge D and E into one cluster first, since they are the closest at the start. After that, we choose B and C, followed by (BC) and (DE), finally A and (BCDE).

- Whenever we merge two clusters, we have to recalculate the distance of that cluster to all other clusters.
- For this example, we picked the minimum distance between two clusters to merge.
- However, there are many other metrics to calculate distance between clusters.

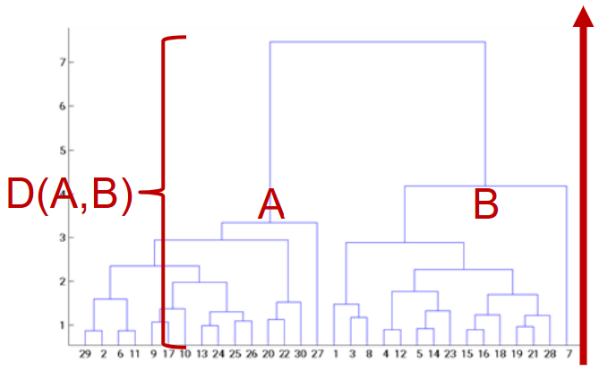
Compute distance between clusters

- **Recipe 1: Single Linkage**:
  - Cluster distance is the distance between two closest members (one from each cluster).
  - Drawback: may produce long, "chain-like" (skinny) clusters.
- **Recipe 2: Complete Linkage**:
  - Cluster distance is the distance between the two farthest members.
  - Drawback: sensitive to outliers and may favor compact, spherical clusters.
- **Recipe 3: Average Linkage**:
  - Cluster distance = average distance over all pairs of points between two clusters.
  - Most widely used; more robust to noise than single or complete linkage.

Algorithmically, this merging process runs in $O(n^2)$ time. There are n merges to create a tree, and each merge requires O(n) updates.

Height in Dendrogram Represents Cluster Distance

- Merge distances are monotonically non-decreasing (rely on properties of the linkage rule, such as: $D(A \cup B, C) \geq \min(D(A, C), D(B, C))$)
- The height in the dendrogram represents the distance at which clusters are merged.

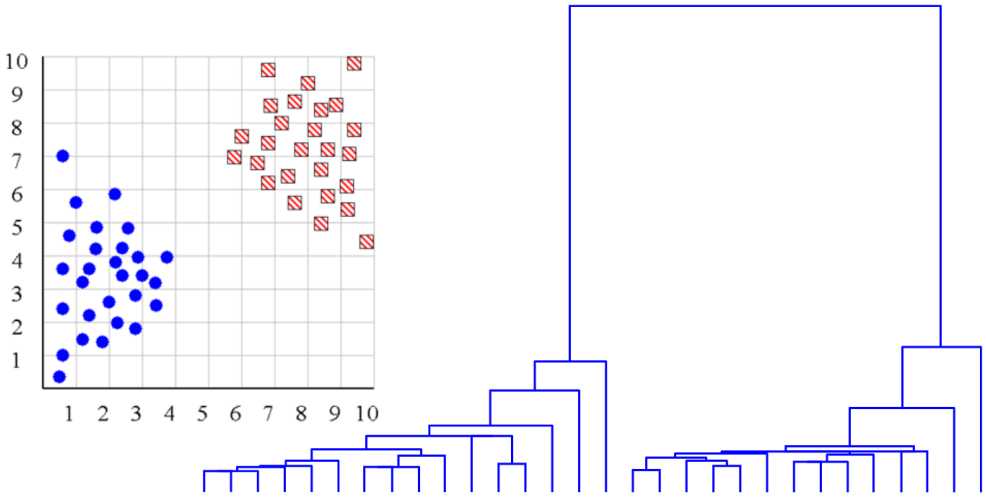


Summary of Hierarchical Clustering Methods

- No need to specify the number of clusters in advance.
- Hierarchical structure maps nicely onto human intuition
- They do not scale well: time complexity of at least $O(N^2)$, where N is the number of total objects.
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results is (very) subjective.

But what are the clusters?

In some cases, we can determine the "correct" number of clusters. However, things are rarely this clear cut, unfortunately.



One potential use of a dendrogram is to detect outliers. An outlier would produce a single, isolated branch.

Takeaway

- Clustering is an unsupervised learning method, and how it works
- What are the different types of clustering algorithms?
- What are the assumptions we are making for each, and what can we get from them?
- Unsolved issues: number of clusters, initiailization, etc.