STA 35C: Statistical Data Science III

Lecture 25: Hierarchical Clustering

Dogyoon Song

Spring 2026, UC Davis

Announcement

Final exam: Fri, June 5 (1:00 pm–3:00 pm) in Wellman Hall 6 (=classroom)

- **Be on time**: The exam starts at 1:00 pm and ends at 3:00 pm sharp
- **Three hand-written cheat sheets allowed:** Letter-size (8.5"×11"), double-sided, brief formulas/notes
- **Calculator**: A simple (non-graphing) scientific calculator is allowed
- **No other materials** beyond the cheat sheets (no textbooks, etc.)
- **SDC accommodations:** Confirm scheduling with AES online as soon as possible

Homework 7 is due tomorrow (Tue, June 2, 11:59 pm)

Course evaluation: Please share your feedback comments by Thu, June 4

Agenda

Last time: K-means clustering

- Clustering problem: group observations using only features X
- K-means objective: minimize within-cluster variation around centroids
- Practical issues: choosing K, scaling, initialization, local optima

Today: Hierarchical clustering

- Dendrograms and nested clusters
- Agglomerative hierarchical clustering algorithm
- Linkage choices: single, complete, and average
- Worked example and interpretation
- Comparison with K-means

Quick review: Clustering problem

Setup:

- Data: $\mathcal{X} = \{X_1, \ldots, X_n\} \subset \mathbb{R}^p$
- **Goal:** Partition observations into clusters so that
  - observations in the same cluster are similar,
  - observations in different clusters are dissimilar.

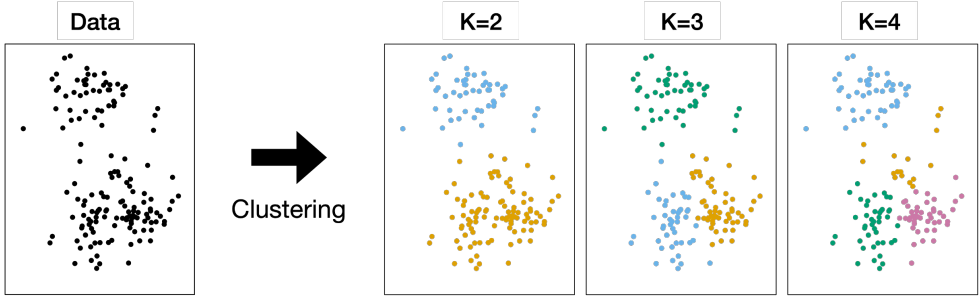

Figure: Given observations (**left**), clustering partitions them into groups (**right**).

Quick review: K-means clustering

K-means: choose K in advance and partition observations into K clusters.

For clusters $C_1, \ldots, C_K$, define the centroid

$$\bar{\mathbf{x}}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \mathbf{x}_i.$$

The K-means objective is

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|^2.$$

Algorithm:

1. Initialize cluster labels or centroids.
2. Update centroids.
3. Reassign each observation to the nearest centroid.
4. Repeat until assignments stop changing.

Limitation: K-means must pre-specify K, and the result can depend on initialization.

K-means clustering: Illustration of the algorithm iterations

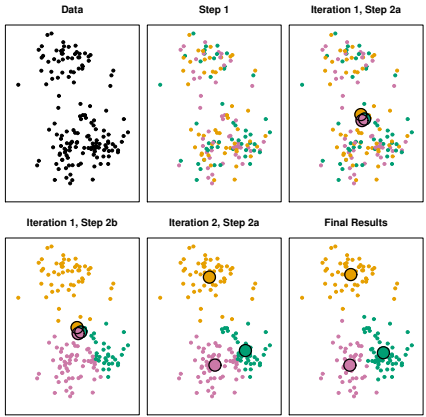

Figure: An example of the K-means with $K = 3$ over two iterations. Each iteration updates centroids (colored disks) and reassigns points [JWHT21, Figure 12.8].

Steps:

1. **Initialize** random cluster labels
2. **Iterate**:
   (a) Update cluster centroids
   (b) Reassign points to nearest centroid

Remarks:

- Each iteration reduces the objective
- Final solution depends on initialization

K-means clustering: Local optima and multiple runs

Figure: K-means with $K = 3$ repeated six times on the same data, each with a different random initial assignment. Above each plot is the final objective. Multiple local optima are found; the best has objective=235.8 [JWHT21, Figure 12.9].

Key points:

- K-means can converge to a suboptimal (local) solution
- Different initial cluster assignments can yield different final partitions
- Usually, re-run with multiple random starts and pick the best (lowest objective)

Hierarchical clustering: Main concept

Motivation: Avoid choosing the number of clusters K in advance

- Instead, build a *dendrogram* that captures how data points "merge" (or "split") at all levels of (dis)similarity to obtain a hierarchy of nested clusters

Agglomerative (bottom-up) approach:

1. Start with n clusters, each containing one observation
2. Repeatedly *merge* the two most similar clusters until only one cluster remains
3. Record the (dis)similarity at each merge to build a **dendrogram**

Clustering after building a dendrogram:

- "Cut" the dendrogram at a chosen height to produce clusters
- Advantage: A single dendrogram can yield clustering into many different K clusters, depending on where we cut

Dendrograms & cutting for clusters

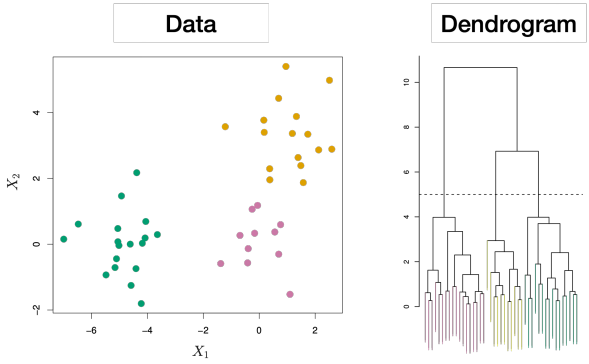

Figure: **Left:** A synthetic dataset (45 points) in 2D. **Right:** Its dendrogram, cut at height 5 (dashed line) yielding three clusters (colored). Colors are for display only, not used in clustering [JWHT21, adapted from Figs. 12.10 & 12.11]

Reading a dendrogram:

- Vertical axis = (dis)similarity at which merges occur
- Lower "merge height" = more similar clusters
- Horizontal spacing is not meaningful for distance

Obtaining clusters:

- "Cut" at a chosen height
- The branches below that cut form the clusters
- The method is *"hierarchical"* as lower cuts nest within higher cuts

Pop-up quiz: Reading a dendrogram

Suppose a dendrogram has the following merges:

$$A \text{ and } B \text{ merge at height } 1, \qquad C \text{ and } D \text{ merge at height } 2,$$

and then $\{A, B\}$ merges with $\{C, D\}$ at height 7.

Question: If we cut the dendrogram at height 3, what clusters do we obtain?

A) $\{A\}, \{B\}, \{C\}, \{D\}$

B) $\{A, B\}, \{C\}, \{D\}$

C) $\{A, B\}, \{C, D\}$

D) Cannot be determined because the horizontal order is missing

Answer: C. A cut at height 3 is above the merges at heights 1 and 2, but below the merge at height 7. Thus we get two clusters: $\{A, B\}$ and $\{C, D\}$.

Interpreting a dendrogram requires care!

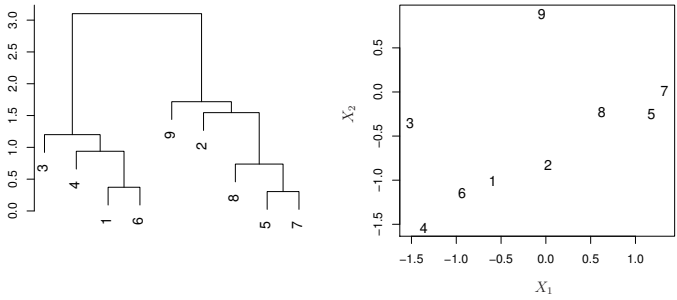

Figure: An illustration of how to interpret a dendrogram with nine observations in 2D. Though points 9 and 2 appear horizontally close, they actually fuse at a higher height than 9 with {8,5,7}, so 9 is no more similar to 2 than it is to {8,5,7} [JWHT21, Figure 12.12].

Note:

- Do *not* interpret horizontal spacing between leaves as distance
- Only merge height indicates dissimilarity
- Dendrogram leaves can often be rotated without changing the clustering

Agglomerative hierarchical clustering algorithm

Agglomerative hierarchical clustering algorithm

1. **Initialize:** Begin with each observation in its own cluster. Compute **pairwise cluster dissimilarities** (e.g., Euclidean distance).
2. **For** $i = n, n-1, \ldots, 2$**:**
   (a) Examine all **pairwise inter-cluster dissimilarities (linkage)** among the i clusters and merge the two closest clusters. Record the dissimilarity of that merge as the "height" in the dendrogram.
   (b) Recompute pairwise distances between the new cluster and all others. Repeat until one cluster remains.

Central question: How do we measure distance between two clusters?

- For individual points, Euclidean distance is common.
- For clusters with multiple points, we need a **linkage rule**.

Linkage: Measuring distance between clusters

Let A and B be two clusters.

Single linkage:

$$d(A, B) = \min_{x \in A,\, y \in B} \|x - y\|.$$

- Distance between closest pair of points.

Complete linkage:

$$d(A, B) = \max_{x \in A,\, y \in B} \|x - y\|.$$

- Distance between farthest pair of points.

Average linkage:

$$d(A, B) = \frac{1}{|A||B|} \sum_{x \in A} \sum_{y \in B} \|x - y\|.$$

- Average distance across all cross-cluster pairs.

In R: `hclust(..., method = "single"/"complete"/"average")`

Hierarchical clustering: Visual illustration of linkage

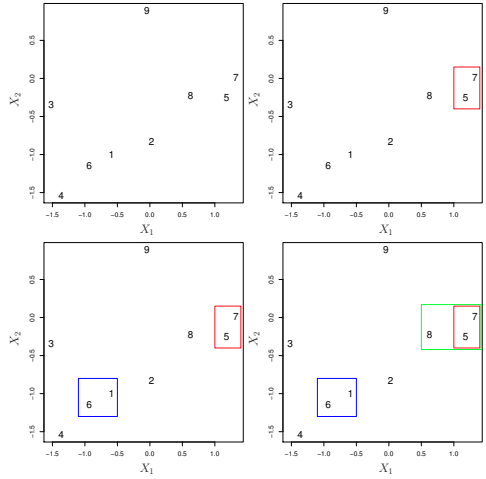

Figure: An illustration of first few steps of hierarchical clustering. **Top Left:** each observation is its own cluster. **Top Right:** clusters $\{5\}$ and $\{7\}$ merge first. **Bottom Left:** next, $\{6\}$ and $\{1\}$ merge. **Bottom Right:** now $\{8\}$ merges with the cluster $\{5, 7\}$. Merges occur at heights = pairwise distances [JWHT21, Figure 12.13].

Pop-up quiz: Linkage

Suppose $A = \{a_1, a_2\}$ and $B = \{b_1, b_2\}$, with cross-cluster distances

	b_1	b_2
a_1	2	8
a_2	3	7

Question: What are the single, complete, and average linkage distances between A and B?

A) Single $= 2$, complete $= 8$, average $= 5$

B) Single $= 8$, complete $= 2$, average $= 5$

C) Single $= 2$, complete $= 7$, average $= 4$

D) All three are 5

Answer: A. Single linkage uses the minimum distance 2; complete linkage uses the maximum distance 8; average linkage uses $(2 + 8 + 3 + 7)/4 = 5$.

Example: Comparing complete vs. single linkage (1/2)

Example

Data set: Let

$$X = \{\, A = (-3, 2),\ B = (-1, 3),\ C = (1, 0),\ D = (4, -3)\} \subset \mathbb{R}^2.$$

We label these four points $\{A, B, C, D\}$, and first compute all $\binom{4}{2} = 6$ pairwise distances:

$$\|A - B\| = \sqrt{5}, \quad \|A - C\| = \sqrt{20}, \quad \|A - D\| = \sqrt{74},$$
$$\|B - C\| = \sqrt{13}, \quad \|B - D\| = \sqrt{61}, \quad \|C - D\| = \sqrt{18}.$$

Numerically, $\sqrt{5} \approx 2.236$, $\sqrt{20} \approx 4.472$, $\sqrt{74} \approx 8.602$, $\sqrt{13} \approx 3.606$, $\sqrt{61} \approx 7.810$, $\sqrt{18} \approx 4.243$.

Step 1: First merge. The smallest pairwise distance is

$$\|A - B\| = \sqrt{5} \approx 2.236.$$

Hence, both *complete* and *single* linkage begin by merging $\{A\}$ with $\{B\}$, forming a new cluster

$$U = \{A, B\}, \quad \text{so we now have clusters } U, \{C\}, \{D\}.$$

Example: Comparing complete vs. single linkage (2/2)

Example

Step 2: Second Merge. Now we have three branches: $U = \{A, B\}, \{C\}, \{D\}$. We compute their pairwise distances using complete and single linkage, respectively.

Complete linkage merges $\{C\}$ with $\{D\}$ next because

$$\begin{aligned}
\operatorname{dist}\big(U, \{C\}\big) &= \max\{\|A - C\|,\ \|B - C\|\} = \max\{\sqrt{20},\ \sqrt{13}\} = \sqrt{20} \approx 4.472, \\
\operatorname{dist}\big(U, \{D\}\big) &= \max\{\|A - D\|,\ \|B - D\|\} = \max\{\sqrt{74},\ \sqrt{61}\} = \sqrt{74} \approx 8.602, \\
\operatorname{dist}\big(\{C\}, \{D\}\big) &= \sqrt{18} \approx 4.243 \quad \textbf{(smallest)}.
\end{aligned}$$

Single linkage merges $\{A, B\}$ with $\{C\}$ second because

$$\begin{aligned}
\operatorname{dist}\big(U, \{C\}\big) &= \min\{\|A - C\|,\ \|B - C\|\} = \min\{\sqrt{20},\ \sqrt{13}\} = \sqrt{13} \approx 3.606 \quad \textbf{(smallest)}, \\
\operatorname{dist}\big(U, \{D\}\big) &= \min\{\|A - D\|,\ \|B - D\|\} = \min\{\sqrt{74},\ \sqrt{61}\} = \sqrt{61} \approx 7.810, \\
\operatorname{dist}\big(\{C\}, \{D\}\big) &= \sqrt{18} \approx 4.243.
\end{aligned}$$

This example illustrates how different linkages can yield different merges.

Visual comparison of linkage choices

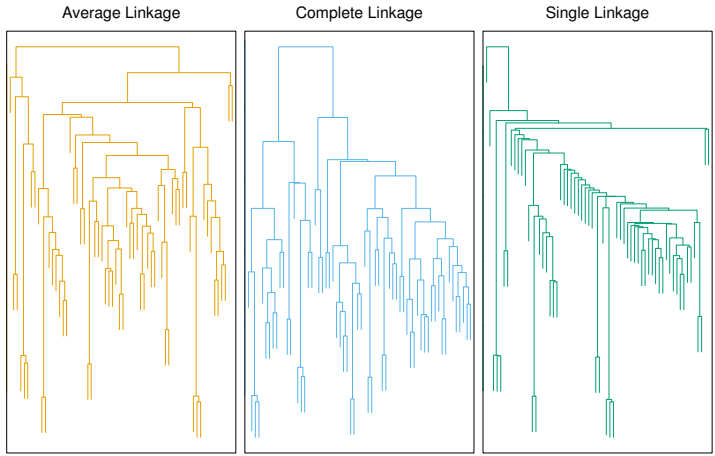

Figure: Comparison of single, average, and complete linkage on the same data. Note that single linkage can produce long "chains," while complete yields more balanced clusters [JWHT21, Figure 12.14].

Choosing clusters from a dendrogram

After building the dendrogram, we still choose a final clustering.

Common choices:

- **Cut by height:** choose a dissimilarity threshold.
- **Cut by number of clusters:** choose K, then cut to get K clusters.
- **Use domain knowledge:** choose clusters that are interpretable and meaningful.

Important: Hierarchical clustering avoids choosing K before fitting, but we still choose how to cut the tree afterward.

Practical advice:

- Try multiple linkage rules.
- Check sensitivity to scaling and distance choice.
- Avoid over-interpreting small visual differences in the dendrogram.

Hierarchical clustering: Strengths and limitations

Strengths:

- Does not require choosing K before fitting.
- Produces a dendrogram showing cluster structure at multiple resolutions.
- Can reveal nested structure.

Limitations:

- Greedy merges: once clusters are merged, they cannot be unmerged.
- Sensitive to distance metric, scaling, and linkage choice.
- Can be computationally expensive for large n.

Wrap-up: Clustering summary

Clustering:

- *Goal:* Partition a dataset (no response labels) into subgroups of "similar" observations
- *Unsupervised*: Typically used for exploratory analysis or hypothesis generation
- No single "correct" distance or method; different choices lead to different clusterings

K-means vs. hierarchical clustering:

K-means	**Hierarchical**
- Partition data into K clusters - Minimizes within-cluster variation	- Builds a *dendrogram* from bottom-up - Cut at a certain height to obtain clusters
- Simple, computationally fast - Easy-to-interpret "centroids" for each cluster	- No need to specify K in advance - One dendrogram can yield many clusterings
- Must pre-specify K - Local search can yield suboptimal solutions	- Greedy merges rely on linkage choice - Nested clusters may be less optimal

References


Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani.
An Introduction to Statistical Learning: with Applications in R, volume 112 of *Springer Texts in Statistics*.
Springer, New York, NY, 2nd edition, 2021.