

# **STA 35C: Statistical Data Science III**

## **Lecture 25: K-means Clustering & Hierarchical Clustering**

Dogyoon Song

Spring 2025, UC Davis

# Announcement

---

**Final exam** on Fri, June 6 (1:00 pm–3:00 pm) in Wellman Hall 26 (=classroom)

- **Instructions:**

- **Arrive on time:** The exam starts at 1:00 pm and ends at 3:00 pm sharp
- **Up to three hand-written cheat sheets:** Letter-size (8.5"×11"), double-sided
- **Calculator:** A simple (non-graphing) scientific calculator is allowed
- **No other materials:** No textbooks, notes, etc., beyond the cheat sheets
- **SDC accommodations:** Confirm your schedule with AES *ASAP*

- **Preparation:**

- *Cumulative* coverage: Lectures 1–25
- A [practice final](#) and [brief answer key](#) are available on the course webpage
- Office hours this week:
  - *Instructor:* Wed, June 4 (4:00–6:00pm, extended); no OH on Thu, June 5
  - *TA:* Mon, June 2 & Thu, June 5, 1–2pm

**Homework 6:** Check out adjustment and correction (Prob 1-(c))

**Course evaluation:** Please share your feedback comments by Thu, June 5

# Today's topics

---

## Topics:

- Clustering (review)
  - Problem setup
  - K-means clustering

### Hierarchical clustering

- Algorithm
- Illustration
- Assessment & comparison to K-means

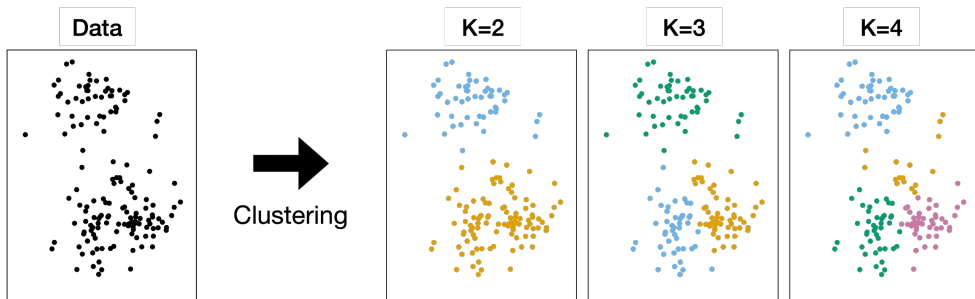
## Learning objectives:

- *Algorithm*: How each clustering method operates
- *Assessment & comparison*: Strengths and limitations of each method

# Quick review: Clustering problem

## Setup:

- Data:  $\mathcal{X} = \{X_1, \dots, X_n\} \subset \mathbb{R}^p$
- **Goal:** Partition the observations into clusters so that points within each cluster are “similar,” while points in different clusters are “different”



**Figure:** Illustration of clustering. Given a dataset of  $X$  (**Left**), we want to partition the observations into  $K$  distinct clusters (**Right**).

# K-means clustering: Objective and algorithm

**Objective:** Given  $K \in \mathbb{N}$ , partition  $\{X_1, \dots, X_n\} \subset \mathbb{R}^p$  into non-overlapping clusters  $C_1, \dots, C_K$  that solve

$$\text{minimize}_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K W(C_k) \right\}$$

- $W(C_k)$  measures within-cluster variation, e.g.,  $W(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \|X_i - X_{i'}\|^2$

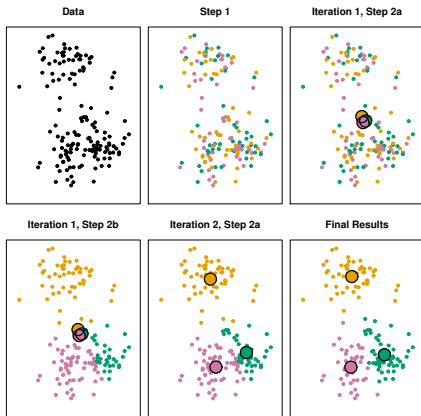
## K-means clustering algorithm (heuristic)

- 1 **Initialize:** Randomly assign each of the  $n$  observations to one of  $K$  clusters
- 2 **Iterate until assignments stop changing:**
  - (a) **Update the centroids.** For each cluster  $C_k$ , compute the centroid

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

- (b) **Reassign.** For each observation  $i$ , reassign it to the cluster whose centroid is closest in squared Euclidean distance

# K-means clustering: Illustration of the algorithm iterations



## 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

**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].

# K-means clustering: Local optima and multiple runs



## 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)

**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].

# K-means clustering: Strengths and limitations

---

## Strengths:

- Simple and computationally fast
- Produces non-overlapping clusters with easy-to-interpret centroids

## Limitations:

- Must pre-specify the number of clusters  $K$ , which a user may not know a priori
- Sensitive to initialization (may get stuck in local optimum)
  - May need to re-run clustering algs multiple times and choose the best
  - Still no guarantee of finding global optimum



# Hierarchical clustering: 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

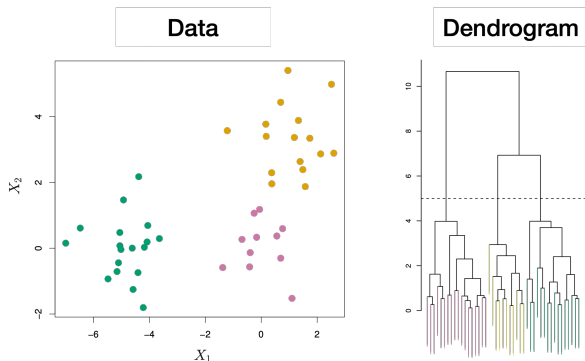
**Agglomerative (bottom-up)** approach:

- 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 from a dendrogram:**

- Once the dendrogram is built, “cut” it at a chosen height to produce a specific number of clusters
- Advantage: A single dendrogram can yield clustering into many different  $K$  clusters

# 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]

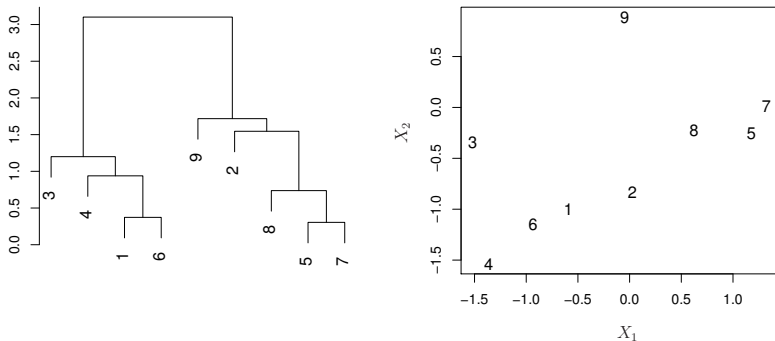
## Reading a dendrogram:

- Vertical axis = (dis)similarity at which merges occur
- Lower “merge height” = more similar
- 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

# 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:** Proximity along the horizontal axis does *not* represent similarity

- Only the height at which merges happen indicates (dis)similarity

# Constructing the dendrogram & linkage choices

## Hierarchical clustering algorithm

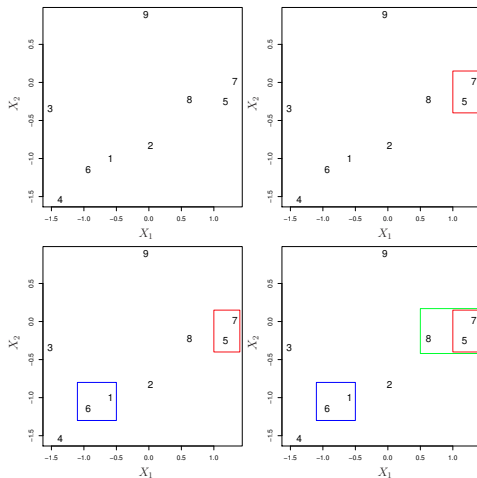
- 1 **Initialize:** Begin with each observation in its own cluster. Compute **pairwise cluster dissimilarities** (e.g., Euclidean distance).
- 2 **For**  $i = n, n - 1, \dots, 2$ :
  - (a) Examine all **pairwise inter-cluster dissimilarities (linkage)** among the  $i$  clusters and merge the two closest 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.

**Linkage options:** how to measure distance between two clusters  $A$  and  $B$

- **Complete** linkage:  $\text{dist}(A, B) = \max\{\|x - y\| : x \in A, y \in B\}$
- **Single** linkage:  $\text{dist}(A, B) = \min\{\|x - y\| \dots\}$
- **Average** linkage:  $\text{dist}(A, B) = \frac{1}{|A| \cdot |B|} \sum_{x \in A} \sum_{y \in B} \|x - y\|$

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

# 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].

## Example: Comparing complete vs. single linkage (Part 1)

### 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:

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

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 (Part 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

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

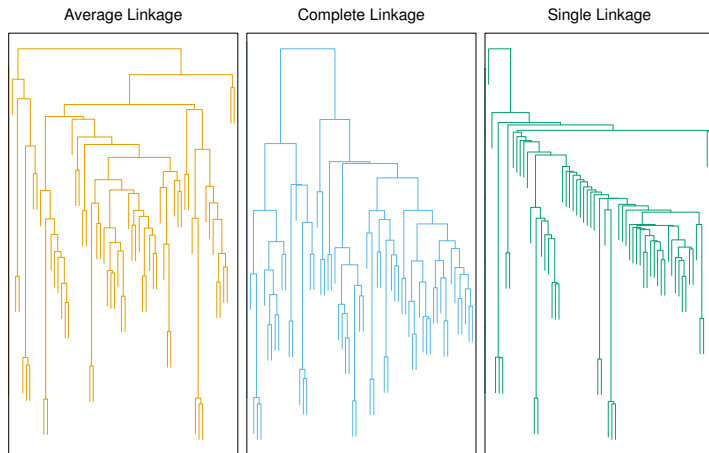
Single linkage merges  $\{A, B\}$  with  $\{C\}$  second because

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

This example illustrates how different linkages can yield different merges.

# Hierarchical clustering: Visual illustration of linkage

---



**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].



# Hierarchical clustering: Strengths and limitations

---

## Strengths:

- No need to specify the number of clusters in advance
- Produces a dendrogram that can be cut at different levels to obtain various clusterings

## Limitations:

- Greedy merges: once two clusters are merged, cannot “unmerge”
- Sensitive to the choice of linkage and distance metric
- 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	Hierarchical
<ul style="list-style-type: none"><li>- Partition data into <math>K</math> clusters</li><li>- Minimizes within-cluster variation</li></ul>	<ul style="list-style-type: none"><li>- Builds a <i>dendrogram</i> from bottom-up</li><li>- Cut at a certain height to obtain clusters</li></ul>
<ul style="list-style-type: none"><li>- Simple, computationally fast</li><li>- Easy-to-interpret “centroids” for each cluster</li></ul>	<ul style="list-style-type: none"><li>- No need to specify <math>K</math> in advance</li><li>- One dendrogram can yield many clusterings</li></ul>
<ul style="list-style-type: none"><li>- Must pre-specify <math>K</math></li><li>- Local search can yield suboptimal solutions</li></ul>	<ul style="list-style-type: none"><li>- Greedy merges rely on linkage choice</li><li>- Nested clusters may be less optimal</li></ul>

# 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.