# Recap: Hierarchical Clustering

DS701 Session 5 — Mon Sep 21, 2026

# Today’s plan

# Knowledge-Check Review

# KC 1: The procedure and the picture

*Describe the agglomerative hierarchical clustering procedure in three steps, and say what the height at which two branches join in a dendrogram represents.*

# KC 2: min versus max

*Single linkage defines the distance between two clusters as the distance between their* ***closest*** *pair of points; complete linkage uses their* ***farthest*** *pair. Explain, from these definitions alone, why single linkage tends to produce long chains that can connect two well-separated groups through a few noise points, and why complete linkage tends toward compact clusters of similar diameter.*

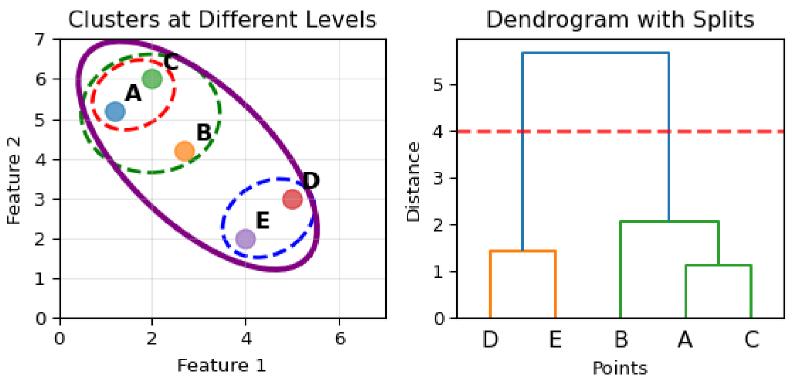
# KC 3: When the client will not tell you $k$

*A client asks you to segment their customers but cannot tell you how many segments they want. Explain why hierarchical clustering is a natural fit here compared to $k$-means, and how you would still get a concrete number of segments out of it.*

# Highlights

# Not one partition — a whole tree

Last session: a **strict partition** — every point in exactly one of $k$ clusters, $k$ chosen in advance.

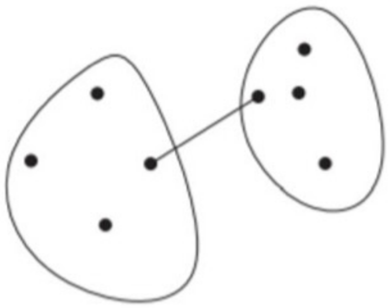


# The agglomerative procedure

1. **Initialise**: each point is its own cluster; compute the pairwise distance matrix if not given.
2. **Merge** the two closest clusters (per the linkage and the metric).
3. **Update** the distances from the new cluster to all remaining clusters.
4. **Repeat** 2–3 until one cluster remains.

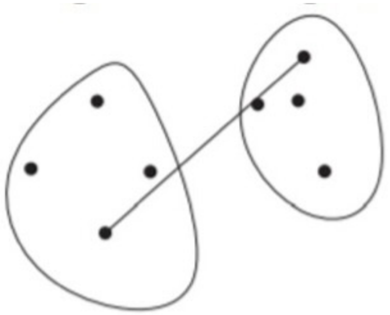
# Everything hinges on the linkage

Distance between two *clusters* $C_i, C_j$ — three natural choices, plus one borrowed from $k$-means:



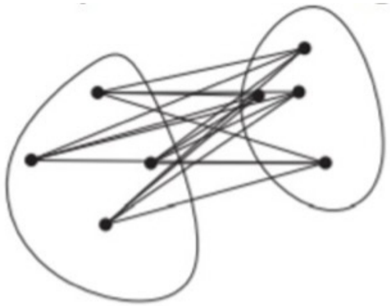
**Single**

$$\min_{x \in C_i,\, y \in C_j} d(x, y)$$



**Complete**

$$\max_{x \in C_i,\, y \in C_j} d(x, y)$$



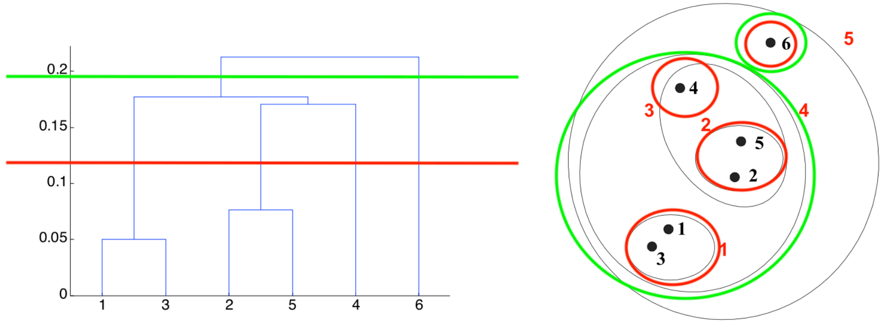
**Average**

$$\frac{1}{|C_i||C_j|} \sum_{x \in C_i,\, y \in C_j} d(x, y)$$

# Same data, four personalities

| linkage | good at | fails when | shape bias |
|---|---|---|---|
| **single** | connected, odd shapes (moons, rings); unequal sizes | a few noise points bridge two groups → **chaining** | none |
| **complete** | balanced, similar-diameter clusters; ignores bridges | one far **outlier** dominates the max | compact / spherical |
| **average** | a compromise; less noise- and outlier-sensitive | — | elliptical |
| **ward** | like $k$-means: robust, tidy blobs | non-convex shapes; needs Euclidean features | elliptical |

# Cutting the tree




Two ways to flatten a hierarchy:

# Hierarchical vs. partitional

| | $k$-**means (partitional)** | **agglomerative (hierarchical)** |
|---|---|---|
| needs $k$ up front | **yes** | no — cut afterwards |
| output | one partition | a tree of nested partitions |
| input | points in $\mathbb{R}^d$ (needs means) | feature matrix **or** any distance / similarity matrix |
| cluster shape | spherical Voronoi cells | depends on linkage |
| cost | cheap, iterative | $O(n^2)$ memory, $O(n^2 \log n)$ time — fine for thousands, not millions |
| repeatability | random init | deterministic |

# In-Class Activity

# Activity: dendrograms on real data, then make it messy

Goal: build and *read* dendrograms for four linkages on a real dataset, cut them into flat clusters, then inject a unit change, two outliers and a chaining bridge and watch which linkages survive — and whether $k$-means would have been fine all along.

- Work in groups of 2–3. Open **your section's** notebook.
- Parts marked **(autograded)** are submitted to Gradescope; the rest is participation.
- Staff will circulate — be ready to explain any part of your work.
- Dependencies: `numpy`, `pandas`, `matplotlib`, `scipy`, `scikit-learn`.

### Section A1 Open in Colab Open on GitHub

### Section B1 Open in Colab Open on GitHub

> ⓘ The activity notebook goes live on the day of the lecture. Colab is optional — you can also open it on GitHub and run it locally.

# Going deeper

Full lecture notes: Hierarchical Clustering

- How Many Clusters? — the same data at 3, 4, 5 and 6 clusters
- Example: Dendrogram — five points, built up merge by merge
- Hierarchical Clustering Algorithms — agglomerative vs divisive; the linkage matrix
- Defining Cluster Proximity — single, complete, average, and their strengths and failures on moons and blobs
- Ward's Distance
- Selecting the Number of Clusters — silhouette on the newsgroups tree
- Comparison of Linkages — scikit-learn's side-by-side figure
- Cluster evaluation (ARI, silhouette) and real-data practice: Clustering in Practice