Lecture Note: Hierarchical Clustering

Luu Minh Sao Khue

1 What is Hierarchical Clustering and Why Do We Use It?

1.1 Motivation: clustering without fixing K at the start

Many clustering methods (for example K-Means) require us to decide the number of clusters K *before* running the algorithm. In practice, this can be difficult:

- We may not know how many groups exist.
- The dataset may have meaningful structure at multiple resolutions (e.g., large groups that split into subgroups).
- We may want an interpretable picture of how clusters merge or split.

Hierarchical clustering addresses these issues by constructing a **hierarchy** of clusters rather than a single partition. The output is a **tree structure** called a **dendrogram**. By cutting the tree at different heights, we obtain different clusterings.

1.2 Key idea: clusters at multiple levels

Instead of producing one "final" set of clusters, hierarchical clustering produces a **sequence of nested clusterings**:

each point → small clusters → larger clusters → one cluster containing all points.

This is especially useful in exploratory data analysis, because it shows how the dataset is organized at different levels of granularity.

2 Notation and Setup

Assume we have:

- n: number of data points (samples).
- d: number of features.
- $x_i \in \mathbb{R}^d$: the i-th data point, for $i = 1, \ldots, n$.

To decide which points/clusters are close, hierarchical clustering needs:

- a **distance (or dissimilarity)** function between points,
- and a rule to define **distance between clusters**, called **linkage**.

3 Distances Between Points

3.1 Euclidean distance (common choice for continuous features)

For two points $x, y \in \mathbb{R}^d$, Euclidean distance is:

$$d(x,y) = \|x-y\|_2 = \sqrt{\sum_{j=1}^{d}(x_j - y_j)^2}.$$

Here:

- x_j is the j-th feature of x,
- y_j is the j-th feature of y,
- the sum measures squared differences across dimensions.

3.2 Why distance choice matters (rationale)

Hierarchical clustering is fundamentally driven by distance comparisons. If the distance does not reflect "similarity" in your domain, then the resulting dendrogram will not be meaningful. This is why preprocessing and feature scaling (standardization) are often essential when features have different units.

4 The Distance Matrix

Hierarchical clustering (in its classical form) uses pairwise distances. We define the **distance matrix** $D \in \mathbb{R}^{n \times n}$ by:

$$D_{ij} = d(x_i, x_j).$$

Properties:

- $D_{ij} \geq 0$ for all i, j (distances are nonnegative).
- $D_{ij} = D_{ji}$ (symmetric for common distances).
- $D_{ii} = 0$ (a point has zero distance to itself).

Why this matters. Storing D requires n^2 entries, so memory can become a bottleneck for large n. This is one major reason hierarchical clustering does not scale as well as K-Means.

5 Two Types of Hierarchical Clustering

5.1 Agglomerative (bottom-up) clustering

This is the most common version in practice.

- Start with n clusters, each cluster contains exactly one point.
- Repeatedly merge the two closest clusters.
- Continue until only one cluster remains.

5.2 Divisive (top-down) clustering

- Start with one cluster containing all points.
- Repeatedly split clusters into smaller ones.

Divisive methods are conceptually useful but often more expensive. In many courses and practical workflows, we focus on agglomerative clustering because it is widely implemented and easier to teach.

6 Cluster Distance: Linkage Criteria

When we have clusters (sets of points), we need a definition of distance between clusters. Let A and B be two clusters, where:

$$A = \{x_i : i \in I_A\}, \quad B = \{x_j : j \in I_B\}.$$

Here I_A and I_B are the index sets of points in the clusters.

6.1 Single linkage (minimum distance)

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

Meaning: define cluster distance as the smallest distance between any pair of points across the two clusters.

Rationale and behavior. Single linkage can connect clusters through chains of points. This often creates long, thin clusters even if the true groups are separate (**chaining effect**). It can work well when clusters are connected by dense paths, but it is sensitive to noise bridges.

6.2 Complete linkage (maximum distance)

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

Meaning: define cluster distance as the largest distance between any pair of points across clusters.

Rationale and behavior. Complete linkage tends to produce compact clusters because it refuses to merge clusters if *any* pair of points would be very far apart. It is more robust to chaining than single linkage but can split large clusters if they have wide spread.

6.3 Average linkage (mean distance)

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

Here $|A|$ is the number of points in cluster A and $|B|$ is the number of points in cluster B.

Rationale and behavior. Average linkage is a compromise between single and complete linkage. It reduces sensitivity to extreme pairs (a single very close or very far pair), and often gives balanced cluster shapes.

6.4 Ward linkage (variance increase; very important)

Ward's method does not define cluster distance purely by pairwise point distances. Instead, it merges clusters that cause the smallest increase in the total within-cluster sum of squares (similar spirit to K-Means).

Let μ_A and μ_B be the means of clusters A and B:

$$\mu_A = \frac{1}{|A|}\sum_{x\in A} x, \quad \mu_B = \frac{1}{|B|}\sum_{x\in B} x.$$

Ward's merge cost (one common form) is:

$$\Delta(A,B) = \frac{|A|\,|B|}{|A|+|B|}\|\mu_A - \mu_B\|_2^2.$$

Explanation of notation:

- $\Delta(A,B)$ is the increase in within-cluster variance if we merge A and B.
- $\|\mu_A - \mu_B\|_2^2$ is the squared Euclidean distance between the cluster means.
- The factor $\frac{|A|\,|B|}{|A|+|B|}$ weights the cost by cluster sizes.

Rationale and behavior. Ward linkage tends to produce compact, roughly spherical clusters and is often a strong default when Euclidean distance is appropriate. It also connects hierarchical clustering to the same geometric idea as K-Means: minimizing within-cluster squared distances.

7 Agglomerative Clustering Algorithm (Step by Step)

7.1 Algorithm overview

Input: points $\{x_i\}_{i=1}^n$, a distance $d(\cdot,\cdot)$, a linkage rule, and (optionally) a target number of clusters K.

Output: a dendrogram (hierarchy) and cluster labels after cutting the dendrogram.

7.2 Step 1: Initialize clusters

Start with each point as its own cluster:

$$\mathcal{C}^{(0)} = \{\{x_1\},\{x_2\},\ldots,\{x_n\}\}.$$

Here $\mathcal{C}^{(0)}$ is the set of clusters at iteration 0.

7.3 Step 2: Compute distances between clusters

Using the linkage rule, compute the distance between every pair of clusters in the current set $\mathcal{C}^{(t)}$.

7.4 Step 3: Merge the closest pair

Find the two clusters with minimum distance:

$$(A^*, B^*) = \arg \min_{A \neq B \in \mathcal{C}^{(t)}} d(A, B).$$

Explanation:

- $\arg\min$ returns the *pair* of clusters that achieves the smallest distance.
- $A \neq B$ ensures we consider two different clusters.

Then merge them into a new cluster:

$$C_{\text{new}} = A^* \cup B^*.$$

7.5 Step 4: Update the cluster set and distances

Replace A^* and B^* in $\mathcal{C}^{(t)}$ with C_{new} to form $\mathcal{C}^{(t+1)}$, and update distances from C_{new} to all remaining clusters (according to the linkage rule).

7.6 Step 5: Repeat

Repeat Steps 2–4 until:

- only one cluster remains, or
- we have exactly K clusters (if we stop early), or
- a distance threshold is reached (stop when merges become too costly).

8 Dendrogram: Interpretation and How We Choose Clusters

8.1 What a dendrogram represents

A dendrogram is a tree where:

- leaves are individual data points,
- internal nodes represent merges of clusters,
- the height of a merge corresponds to the distance (or merge cost) at which two clusters were combined.

8.2 Choosing the number of clusters by cutting the tree

If we draw a horizontal line across the dendrogram at height h, each connected component below that line corresponds to one cluster. In other words:

- A low cut (small h) produces many small clusters.
- A high cut (large h) produces fewer large clusters.

This provides a flexible way to obtain clusterings at different levels.

9 Complexity and Practical Considerations

9.1 Why hierarchical clustering can be expensive

Because hierarchical clustering relies on pairwise distances, it often requires:

- memory: storing $D \in \mathbb{R}^{n \times n}$ is $O(n^2)$,
- time: repeated merging and distance updates can be $O(n^2)$ or worse.

Therefore, hierarchical clustering is best suited for small-to-medium datasets, or situations where interpretability is more important than scalability.

10 Strengths and Limitations (With Rationale)

10.1 Strengths

- **No need to predefine K:** the dendrogram allows choosing the number of clusters after seeing the structure.
- **Interpretability:** the hierarchy gives insight into how data points group together at multiple levels.
- **Deterministic (often):** unlike K-Means, there is no random initialization in classical agglomerative clustering.

10.2 Limitations

- **Scalability:** $O(n^2)$ memory/time is a major limitation.
- **Sensitivity to noise and outliers:** a single outlier can create misleading early merges or long branches in the dendrogram.
- **Greedy merges:** once two clusters are merged, the decision cannot be undone, so early mistakes can affect the final structure.

11 Practical Workflow Recommendations

A reliable workflow when using hierarchical clustering:

1. Clean missing values and handle outliers if necessary.
2. Scale features (e.g., z-score standardization) so distances are meaningful.
3. Choose a distance metric that matches the data type and domain.
4. Choose linkage:
    - Ward for compact, Euclidean-style clusters,
    - average for balanced behavior,
    - complete for compact clusters with stricter merging,

- single only when chaining structure is expected.

5. Inspect the dendrogram and decide a cut (or choose K) based on interpretability and domain needs.

12 Summary

Hierarchical clustering constructs a tree of nested clusters, allowing us to explore structure at multiple resolutions. Its mathematics is driven by:

- a point-to-point distance function $d(x_i, x_j)$,
- a linkage rule $d(A, B)$ defining distance between clusters,
- a greedy bottom-up merging process that builds a dendrogram.

Compared to K-Means, hierarchical clustering is more interpretable and does not require choosing K upfront, but it is less scalable due to pairwise distance computations and memory requirements.