HIERARCHICAL CLUSTERING

As its name suggests, hierarchical clustering produces not just one partition of a dataset into clusters, but a hierarchy of clusterings.

Let the data be \( x_1, \ldots, x_n \).

A hierarchical clustering has \( n \) levels. Each level corresponds to a different partition or cluster map. These levels are hierarchical in the sense that:

- Level \( n \) \( \rightarrow \) \{ \( x_1 \), 3, 3, 2, 3, \ldots, 3 \( x_n \) \}
- Level 1 \( \rightarrow \) \{ \( x_1 \), \ldots, \( x_n \) \}
- Level \( k \), \( 1 \leq k < n \) \( \rightarrow \) Formed by merging two clusters at level \( k+1 \)
The primary reason why people like hierarchical clustering is because of a graphical representation called a ___________. Any horizontal line across this graph corresponds to a particular partition in the hierarchy.

- **Horizontal axis**: no physical meaning, just shows organization of clusters (not unique)
- **Vertical axis**: dissimilarity of ________ clusters.
Compared to non-hierarchical clustering algorithms such as K-means, hierarchical clustering has the following advantages:

- Clusters may exist at multiple scales, i.e., clusters may have ________
- Does not require specifying the number of clusters in advance.

There are two classes of algorithms for HC:

- bottom-up or ________
- top-down or ________
Agglomerative Hierarchical Clustering

Both agglomerative and divisive HC require input in the form of a matrix

\[ D = [d_{ij}]_{ij=1}^n, \quad d_{ij} = d(x_i, x_j) \]

The dissimilarity matrix is then used to define the dissimilarity between two... there are several ways to do this.

Example

Suppose $A, B$ are clusters. We may define the dissimilarity between $A$ and $B$ to be

\[ d_{avg}(A, B) := \]

Note In these notes we speak of clusters as subsets of $\{x_1, \ldots, x_n\}$ as opposed to subsets of $\mathbb{R}^d$. 
Agglomerative HC implements the following algorithm, where

\[ H_k \] is the set of clusters at level \( k \)

\[
\begin{align*}
\text{Initialize } & H_0 = \left\{ \{x_1\}, \{x_2\}, \ldots, \{x_n\} \right\} \\
\text{For } k = n-1 \text{ down to } 1 & \\
& \quad \text{Select cluster } A, B \in H_{k+1} \text{ for which } d(A, B) \text{ is } \underline{\quad} \\
& \quad \text{Set } H_k \text{ to be } H_{k+1} \text{ with } A \text{ and } B \text{ deleted and } A \cup B \text{ added} \\
\text{End}
\end{align*}
\]

In other words, we iteratively the two least dissimilar clusters until we have one cluster.
Linkage

The formula that relates point dissimilarities to cluster dissimilarities is called the

 Examples:

- average \( d_{avg}(A,B) = \)
- single \( d_{min}(A,B) = \)
- complete \( d_{max}(A,B) = \)
- centroid \( d_{cent}(A,B) = \)
- Ward's \( d_{Ward}(A,B) = \sqrt{\frac{n_A n_B}{n_A + n_B}} \| \overline{x}_A - \overline{x}_B \| \)
Remarks

- The centroid and Ward's linkage are not built out of an underlying point dissimilarity.

- The average, single, and complete linkages can be applied to cluster data as long as point dissimilarities can be defined.

- The choice of linkage function has a major effect on the HC. Furthermore, there is often no clear choice which one to use.
More Remarks

- single linkage

  \[ \rightarrow \] generates a _____ _____ _____

  \[ \rightarrow \] sensitive to outliers: tends to merge them at the very end

  \[ \rightarrow \] chaining: tends to produce elongated clusters

- complete linkage

  \[ \rightarrow \] discourages elongated clusters,
  favors clusters with small _____

- average

  \[ \rightarrow \] compromise between single and complete

  \[ \rightarrow \] affected by monotone scaling of \( d_{ij} \)

- centroid

  \[ \rightarrow \] easy to compute

  \[ \rightarrow \] dendrogram can be non-monotone

- Ward's

  \[ \rightarrow \] corrects centroid's monotonicity problem
Monotonicity

Certain linkages have a monotonicity property that allows us to assign a quantitative value to the height of nodes in the

In particular, suppose a node was formed by merging two clusters A and B. Then the height of $A \cup B$ is defined to be

**Definition 1** A linkage $d$ is monotone if, for any cluster $\{A \cup B \cup C$ produced by HC, we have

Denotes order of merging
Consider a simple example:

Denote \( h = \text{height} \)

\[ h(M) = \]

\[ h(O) = \]

\[ h(P) = \]
Example 1 The single linkage has the monotone property.

To see this, suppose HC produces the cluster \( \{A \cup B \} \cup C \).

Then

\[
d(A \cup B, C) = \min_{x \in A \cup B, z \in C} d(x, z) d(z, A) \\
= \min \left\{ \min_{x \in A} d(x, z), \min_{y \in B} d(y, z) \right\}
\]

≥
• complete is monotone: same proof as for single
  \[ \text{with } \min \to \max \]

• average is monotone:
\[
d_{\text{avg}}(A \cup B, C) = \frac{1}{n_C} \cdot \frac{1}{n_A+n_B} \sum_{z \in C} \sum_{x \in A \cup B} d(z, x)
\]
\[
= \frac{1}{n_C} \sum_{z \in C} \left( \frac{1}{n_A+n_B} \sum_{x \in A} d(z, x) + \frac{1}{n_A+n_B} \sum_{y \in B} d(z, y) \right)
\]
\[
= \frac{n_A}{n_A+n_B} d(A, C) + \frac{n_B}{n_A+n_B} d(B, C)
\]
\[
\geq \frac{n_A}{n_A+n_B} d(A, B) + \frac{n_B}{n_A+n_B} d(A, B)
\]
\[
\text{[otherwise } C \text{ would have merged with } A \text{ or } B]\]
\[
= d(A, B)
\]

• centroid is not monotone

\[
\text{counterexample: equilateral triangle } \rightarrow \text{ centroid }
\]

• Ward's is monotone: proof based on connection
to within-class scatter
Global criterion?

HC defines a cluster to be the output of a certain algorithm.

Can we view HC as an algorithm for (approximately) optimizing a global objective function?

Let $J_k$ be an objective function that assess the quality of a clustering into $K$ clusters.

\[
\text{Initialize } \mathcal{H}_n = \{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}
\]

For $k = n \downarrow 1$

1. Find $A, B \in \mathcal{H}_{k+1}$ such that merging $A$ and $B$ to form $\mathcal{H}_k$ yields the smallest $J_k$

End
Does this algorithm ever coincide with HIC? Sometimes.

Examples:

- \( d = d_{\text{max}} = \text{complete linkage} \)

\[
\gamma_k(H) = 
\]

- \( d = d_{\text{ward}} = \text{Ward's linkage} \)

\[
\gamma_k(H) = 
\]

\( \) requires a little algebra to verify this.
Divisive Hierarchical Clustering

Initialize \( \mathcal{H}_1 = \{ x_1, \ldots, x_n \} \)

For \( k = 2 : n \)
  - Select a cluster \( C \in \mathcal{H}_{k-1} \)
  - Split \( C \) into clusters \( A \) and \( B \)
  - Set \( \mathcal{H}_k \) to be \( \mathcal{H}_{k-1} \) with \( C \) replaced by \( A \) and \( B \)

End

Comments:

- Less common than agglomerative methods
- Splits must be chosen carefully to ensure a monotone dendrogram
- Can be faster than agglomeration if only a small number of clusters is desirable.
Other uses

- Initialization for other clustering methods such as K-means

Choosing K

Although HC produces a nice tree, we may want to choose a specific level. We can

- use the same method used for K-means
- look for a large jump in the dendrogram
Instability

Like ________, HC is sensitive to perturbations of the data

Interpretation

Dendrogram = summary of ________

≠ summary of ________

To what extent does dendrogram represent the actual structure of the data?

Model-Based Interpretation

HC may be viewed as a greedy method for maximum likelihood estimation of cluster parameters, where different generative models correspond to different linkages. See Kamvar, Klein, and Manning, "Interpreting and Extending Classical Agglomerative Clustering Algorithms Using a Model-Based Approach."
A. n  B. dendrogram  C. child
D. subclusters; agglomerative/divisive
E. dissimilarity, clusters, $d_{av}(A,B) = \frac{1}{|A| \cdot |B|} \sum_{x \in A} \sum_{y \in B} d(x,y)$
F. minimal, merge  G. linkage

\[ d_{\text{min}}(A,B) = \min_{x \in A \atop y \in B} d(x,y) \]
\[ d_{\text{max}}(A,B) = \max_{x \in A \atop y \in B} d(x,y) \]
\[ d_{\text{cent}}(A,B) = \| \bar{x}_A - \bar{x}_B \| \]

H. non-Euclidean  I. minimal spanning tree, diameter
J. dendrogram; $d(A,B)$; $d(A \cup B, C) \geq d(A,B)$
K. $h(M) = d(L,F) = d(D \cup E, F)$
   \[ \geq d(D, E) = h(L) \]
   $h(C) = d(M, K) = d(L \cup F, K)$
   \[ \geq d(L, F) = h(M) \]
   $h(P) = d(C, N)$ ...
   $\geq h(C)$ and $h(N)$
L. $\geq \min_{x \in A \atop y \in B} d(x,y) = d(A,B)$, otherwise $A,B$ would have merged with $C$
M. greedy, 
\[ J_k(H) = \max_{A \in H} \left( \max_{x, y \in A} d(x, y) \right) \]
\[ \text{max cluster "diameter"} \]
\[ J_k(H) = \text{within-cluster scatter (as in K-means)} \]

N. Decision trees; algorithm, data