HIERARCHICAL CLUSTERING ALGORITHMS

- They produce a hierarchy of (hard) clusterings instead of a single clustering.

- Applications in:
  - Social sciences
  - Biological taxonomy
  - Modern biology
  - Medicine
  - Archaeology
  - Computer science and engineering
Let \( X=\{x_1,\ldots,x_N\} \), \( x_i=[x_{i1},\ldots,x_{il}]^T \). Recall that:

- In hard clustering each vector belongs **exclusively** to a single cluster.
- An \( m \)-hard clustering of \( X \), \( \mathcal{R} \), is a partition of \( X \) into \( m \) sets (clusters) \( C_1,\ldots,C_m \), so that:
  
  \[ C_i \neq \emptyset, \, i=1,2,\ldots,m \]

  \[ \bigcup_{i=1}^{m} C_i = X \]

  \[ C_i \cap C = \emptyset, \, i \neq j, \, i, \, j=1,2,\ldots,m \]

By the definition: \( \mathcal{R} = \{C_j, \, j=1,\ldots,m\} \)

**Definition:** A clustering \( \mathcal{R}_1 \) containing \( k \) clusters is said to be **nested** in the clustering \( \mathcal{R}_2 \) containing \( r \) (\(<k\)) clusters, if each cluster in \( \mathcal{R}_1 \) is a subset of a cluster in \( \mathcal{R}_2 \).

We write \( \mathcal{R}_1 \preceq \mathcal{R}_2 \)
Example: Let $\mathcal{R}_1 = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\}$, $\mathcal{R}_2 = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$,

$$\mathcal{R}_3 = \{\{x_1, x_4\}, \{x_3\}, \{x_2, x_5\}\}, \mathcal{R}_4 = \{\{x_1, x_2, x_4\}, \{x_3, x_5\}\}.$$ 

It is $\mathcal{R}_1 \not\leq \mathcal{R}_2$, but not $\mathcal{R}_1 \not\leq \mathcal{R}_3$, $\mathcal{R}_1 \not\leq \mathcal{R}_4$, $\mathcal{R}_1 \not\leq \mathcal{R}_1$.

Remarks:
- Hierarchical clustering algorithms produce a hierarchy of nested clusterings.
- They involve $N$ steps at the most.
- At each step $t$, the clustering $\mathcal{R}_t$ is produced by $\mathcal{R}_{t-1}$.

Main categories:
- **Agglomerative** clustering algorithms: Here $\mathcal{R}_0 = \{\{x_1\}, \ldots, \{x_N\}\}$, $\mathcal{R}_{N-1} = \{\{x_1, \ldots, x_N\}\}$ and $\mathcal{R}_0 \not\leq \ldots \not\leq \mathcal{R}_{N-1}$.
- **Divisive** clustering algorithms: Here $\mathcal{R}_0 = \{\{x_1, \ldots, x_N\}\}$, $\mathcal{R}_{N-1} = \{\{x_1\}, \ldots, \{x_N\}\}$ and $\mathcal{R}_{N-1} \not\leq \ldots \not\leq \mathcal{R}_0$. 
AGGLOMERATIVE ALGORITHMS

- Let $g(C_i, C_j)$ a proximity function between two clusters of $X$.

- **Generalized Agglomerative Scheme (GAS)**
  
  **Initialization**
  - Choose $\mathcal{R}_0=\{\{x_1\}, \ldots, \{x_N\}\}$
  - $t=0$

  **Repeat**
  - $t=t+1$
  - Choose $(C_i, C_j)$ in $\mathcal{R}_{t-1}$ such that
  
  $$g(C_i, C_j) = \begin{cases} 
  \min_{r,s} g(C_r, C_s), & \text{if } g \text{ is a disim. function} \\
  \max_{r,s} g(C_r, C_s), & \text{if } g \text{ is a sim. function}
  \end{cases}$$

  - Define $C_q=C_i \cup C_j$ and produce $\mathcal{R}_t=(\mathcal{R}_{t-1}-\{C_i, C_j\}) \cup \{C_q\}$

  **Until all vectors form a single cluster.**
Remarks:

• If two vectors come together into a single cluster at level \( t \) of the hierarchy, they will remain in the same cluster for all subsequent clusterings.

• As a consequence, there is no way to recover a “poor” clustering that may have occurred in an earlier level of hierarchy.

• Number of operations: \( O(N^3) \)
Definitions of some useful quantities:

Let $X=\{x_1, x_2, \ldots, x_N\}$, with $x_i=[x_{i1}, x_{i2}, \ldots, x_{il}]^T$.

- **Pattern matrix** ($D(X)$): An $N \times l$ matrix whose $i$-th row is $x_i$ (transposed).

- **Proximity (similarity or dissimilarity) matrix** ($P(X)$): An $N \times N$ matrix whose $(i,j)$ element equals the proximity $\varrho(x_i, x_j)$ (similarity $s(x_i, x_j)$, dissimilarity $d(x_i, x_j)$).

**Example 1**: Let $X=\{x_1, x_2, x_3, x_4, x_5\}$, with $x_1=[1, 1]^T$, $x_2=[2, 1]^T$, $x_3=[5, 4]^T$, $x_4=[6, 5]^T$, $x_5=[6.5, 6]^T$.

$$D(X) = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{bmatrix} \quad P(X) = \begin{bmatrix} 0 & 1 & 5 & 6.4 & 7.4 \\ 1 & 0 & 4.2 & 5.7 & 6.7 \\ 5 & 4.2 & 0 & 1.4 & 2.5 \\ 6.4 & 5.7 & 1.4 & 0 & 1.1 \\ 7.4 & 6.7 & 2.5 & 1.1 & 0 \end{bmatrix} \quad P'(X) = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1 & 0.44 & 0.35 & 0.20 \\ 0.26 & 0.44 & 1 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1 & 0.98 \\ 0.18 & 0.20 & 0.90 & 0.98 & 1 \end{bmatrix}$$
Threshold dendrogram (or dendrogram): It is an effective way of representing the sequence of clusterings which are produced by an agglomerative algorithm.

In the previous example, if $d_{	ext{min}}^{ss}(C_i, C_j)$ is employed as the distance measure between two sets and the Euclidean one as the distance measure between two vectors, the following series of clusterings are produced:

\[
\begin{align*}
\{ \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \} \\
\{ \{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \} \\
\{ \{x_1, x_2\}, \{x_3\}, \{x_4, x_5\} \} \\
\{ \{x_1, x_2\}, \{x_3, x_4, x_5\} \} \\
\{ \{x_1, x_2, x_3, x_4, x_5\} \}
\end{align*}
\]
Proximity (dissimilarity or dissimilarity) dendrogram: A dendrogram that records the level of proximity (dissimilarity or similarity) where two clusters are merged.

Example 2: In terms of the previous example, the proximity dendrograms that correspond to $P'(X)$ and $P(X)$ are

Remark: One can readily observe the level in which a cluster is formed and the level in which it is absorbed in a larger cluster (indication of the natural clustering).
Agglomerative algorithms are divided into:

- Algorithms based on matrix theory.
- Algorithms based on graph theory.

In the sequel we focus only on dissimilarity measures.

Algorithms based on matrix theory (?).

- They take as input the $N \times N$ dissimilarity matrix $P_0 = P(X)$.
- At each level $t$ where two clusters $C_i$ and $C_j$ are merged to $C_q$, the dissimilarity matrix $P_t$ is extracted from $P_{t-1}$ by:
  - Deleting the two rows and columns of $P_t$ that correspond to $C_i$ and $C_j$.
  - Adding a new row and a new column that contain the distances of newly formed $C_q = C_i \cup C_j$ from the remaining clusters $C_s$, via a relation of the form
    \[
    d(C_q, C_s) = f(d(C_i, C_s), d(C_j, C_s), d(C_i, C_j))
    \]
A number of distance functions comply with the following update equation

\[ d(C_q, C_s) = a_i d(C_i, C_s) + a_j (d(C_j, C_s) + bd(C_i, C_j) + c|d(C_i, C_s) - d(C_j, C_s)|) \]

Algorithms that follow the above equation are:

- **Single link (SL) algorithm** \((a_i=1/2, a_j=1/2, b=0, c=-1/2)\). In this case
  
  \[ d(C_q, C_s) = \min\{d(C_i, C_s), d(C_j, C_s)\} \]

- **Complete link (CL) algorithm** \((a_i=1/2, a_j=1/2, b=0, c=1/2)\). In this case
  
  \[ d(C_q, C_s) = \max\{d(C_i, C_s), d(C_j, C_s)\} \]

Remarks:

- Single link forms clusters at low dissimilarities while complete link forms clusters at high dissimilarities.
- Single link tends to form elongated clusters (chaining effect) while complete link tends to form compact clusters.
- The rest algorithms are compromises between these two extremes.
Example:

(a) The data set $X$.
(b) The single link algorithm dissimilarity dendrogram.
(c) The complete link algorithm dissimilarity dendrogram.
Weighted Pair Group Method Average (WPGMA) \((a_i=1/2, a_j=1/2, b=0, c=0)\). In this case:

\[
d(C_q, C_s) = \frac{(d(C_i, C_s) + d(C_j, C_s))}{2}
\]

Unweighted Pair Group Method Average (UPGMA) \((a_i=n_i/(n_i+n_j), a_j=n_j/(n_i+n_j), b=0, c=0, \) where \(n_i\) is the cardinality of \(C_i\). In this case:

\[
d(C_q, C_s) = \frac{(n_i d(C_i, C_s) + n_j d(C_j, C_s))}{(n_i+n_j)}
\]

Unweighted Pair Group Method Centroid (UPGMC) \((a_i=n_i/(n_i+n_j), a_j=n_j/(n_i+n_j), b=-n_i n_j/(n_i+n_j)^2, c=0)\). In this case:

\[
d_{qs} = \frac{n_i}{n_i+n_j} d_{is} + \frac{n_j}{n_i+n_j} d_{js} - \frac{n_i n_j}{(n_i+n_j)^2} d_{ij}
\]

For the UPGMC, it is true that \(d_{qs} = \|m_q - m_s\|^2\), where \(m_q\) is the mean of \(C_q\).
Weighted Pair Group Method Centroid (WPGMC) \((a_i=1/2, \ a_j=1/2, \ b=-1/4, \ c=0)\). In this case
\[ d_{qs} = \frac{(d_{is} + d_{js})}{2} - \frac{d_{ij}}{4} \]

For WPGMC there are cases where \(d_{qs} \leq \max\{d_{is}, \ d_{js}\}\) (crossover)

Ward or minimum variance algorithm. Here the distance \(d_{ij}^{'}\) between \(C_i\) and \(C_j\) is defined as
\[ d_{ij}^{'} = \frac{n_i n_j}{(n_i+n_j)} \|m_i - m_j\|^2 \]

\(d_{qs}^{'}\) can also be written as
\[ d_{qs}^{'} = \left( \left( n_i + n_j \right) d_{is}^{'} + \left( n_i + n_j \right) d_{js}^{'} - n_s d_{ij}^{'} \right) / (n_i+n_j+n_s) \]

Remark: Ward’s algorithm forms \(R_{t+1}\) by merging the two clusters that lead to the smallest possible increase of the total variance, i.e.,
\[ E_t = \sum_{r=1}^{N-t} \sum_{x \in C_r} \| x - m_r \|^2 \]
Example 3: Consider the following dissimilarity matrix (Euclidean distance)

\[
P_0 = \begin{bmatrix}
0 & 1 & 2 & 26 & 37 \\
1 & 0 & 3 & 25 & 36 \\
2 & 3 & 0 & 16 & 25 \\
26 & 25 & 16 & 0 & 1.5 \\
37 & 36 & 25 & 1.5 & 0 \\
\end{bmatrix}
\]

\[
\mathcal{R}_0 = \{ \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \},
\]

\[
\mathcal{R}_1 = \{ \{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \},
\]

\[
\mathcal{R}_2 = \{ \{x_1, x_2\}, \{x_3\}, \{x_4, x_5\} \},
\]

\[
\mathcal{R}_3 = \{ \{x_1, x_2, x_3\}, \{x_4, x_5\} \},
\]

\[
\mathcal{R}_4 = \{ \{x_1, x_2, x_3, x_4, x_5\} \}
\]

All the algorithms produce the above sequence of clusterings at different proximity levels:

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>CL</th>
<th>WPGMA</th>
<th>UPGMA</th>
<th>WPGMC</th>
<th>UPGMC</th>
<th>Ward</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mathcal{R}_0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(\mathcal{R}_1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>(\mathcal{R}_2)</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>0.75</td>
</tr>
<tr>
<td>(\mathcal{R}_3)</td>
<td>2</td>
<td>3</td>
<td>2.5</td>
<td>2.5</td>
<td>2.25</td>
<td>2.25</td>
<td>1.5</td>
</tr>
<tr>
<td>(\mathcal{R}_4)</td>
<td>16</td>
<td>37</td>
<td>25.75</td>
<td>27.5</td>
<td>24.69</td>
<td>26.46</td>
<td>31.75</td>
</tr>
</tbody>
</table>
Monotonicity and crossover:

For the following dissimilarity matrix

\[
P = \begin{bmatrix}
0 & 1.8 & 2.4 & 2.3 \\
1.8 & 0 & 2.5 & 2.7 \\
2.4 & 2.5 & 0 & 1.2 \\
2.3 & 2.7 & 1.2 & 0 \\
\end{bmatrix}
\]

the dissimilarity dendrograms produced by single link, complete link and UPGMC (the same result is produced if WPGMC is employed) are:

\{x_1, x_2, x_3, x_4\} is formed at lower dissimilarity level than \{x_1, x_2\} (crossover)
Monotonicity condition:

If clusters $C_i$ and $C_j$ are selected to be merged in cluster $C_q$, at the $t$th level of the hierarchy, the condition

$$d(C_q, C_k) \geq d(C_i, C_j)$$

must hold for all $C_k, k \neq i, j, q$.

In other words, the monotonicity condition implies that a cluster is formed at higher dissimilarity level than any of its components.

Remarks:

- Monotonicity is a property that is exclusively related to the clustering algorithm and not to the (initial) proximity matrix.
- An algorithm that does not satisfy the monotonicity condition, does not necessarily produce dendrograms with crossovers.
- Single link, complete link, UPGMA, WPGMA and the Ward’s algorithm satisfy the monotonicity condition, while UPGMC and WPGMC do not.
Complexity issues:
- GAS requires, in general, $O(N^3)$ operations.
- More efficient implementations require $O(N^2 \log N)$ computational time.
- For a class of widely used algorithms, implementations that require $O(N^2)$ computational time and $O(N^2)$ or $O(N)$ storage have also been proposed.
- Parallel implementations on SIMD machines have also been considered.
Some basic definitions from graph theory:

- A graph, $G$, is defined as an ordered pair $G=(V,E)$, where $V=\{v_i, i=1,\ldots,N\}$ is a set of vertices and $E$ is a set of edges connecting some pairs of vertices. An edge connecting $v_i$ and $v_j$ is denoted by $e_{ij}$ or $(v_i,v_j)$.
- A graph is called undirected graph if there is no direction assigned to any of its edges. Otherwise, we deal with directed graphs.
- A graph is called unweighted graph if there is no cost associated with any of its edges. Otherwise, we deal with weighted graphs.
- A path in $G$ between vertices $v_{i_1}$ and $v_{i_n}$ is a sequence of vertices and edges of the form $v_{i_1} e_{i_1i_2} v_{i_2} \ldots v_{i_{n-1}} e_{i_{n-1}i_n} v_{i_n}$.
- A loop in $G$ is a path where $v_{i_1}$ and $v_{i_n}$ coincide.
- A subgraph $G´=(V´,E´)$ of $G$ is a graph with $V´ \subseteq V$ and $E´ \subseteq E_f$, where $E_f$ is a subset of $E$ containing vertices that connect vertices of $V´$. Every graph is a subgraph to itself.
- A connected subgraph $G´=(V´,E´)$ is a subgraph where there exists at least one path connecting any pair of vertices in $V´$. 
• A complete subgraph $G'=(V',E)$ is a subgraph where for any pair of vertices in $V'$ there exists an edge in $E'$ connecting them.
• A maximally connected subgraph of $G$ is a connected subgraph $G'$ of $G$ that contains as many vertices of $G$ as possible.
• A maximally complete subgraph of $G$ is a complete subgraph $G'$ of $G$ that contains as many vertices of $G$ as possible.

Examples for the above, are shown in the following figure.
NOTE: In the framework of clustering, each vertex of a graph corresponds to a feature vector.

Useful tools for the algorithms based on graph theory are the threshold graph and the proximity graph.

- A threshold graph $G(a)$
  - is an undirected, unweighted graph with $N$ nodes, each one corresponding to a vector of $X$.
  - The set of edges of $G(a)$ contains those edges $(v_i, v_j)$ for which the distance $d(x_i, x_j)$ between the vectors corresponding to $v_i$ and $v_j$ is less than $a$.

- A proximity graph $G_p(a)$ is a threshold graph $G(a)$, all of whose edges $(v_i, v_j)$ are weighted with the proximity measure $d(x_i, x_j)$. 


(a) The threshold graph $G(3)$, (b) the proximity (dissimilarity) graph $G_p(3)$, (c) the threshold graph $G(5)$, (d) the dissimilarity graph $G_p(5)$, for the dissimilarity matrix $P(X)$ given in example 1.
More definitions:
In this framework, we consider graphs $G$, of $N$ nodes, where each node corresponds to a vector of $X$.
Valid clusters are connected components of $G$ that satisfy an additional graph property $h(k)$.

Typical graph properties for a connected subgraph $G'$ of $G$ are:

- **Node connectivity**: The largest integer $k$ such that all pairs of nodes of $G'$ are joined by at least $k$ paths having no nodes in common.
- **Edge connectivity**: The largest integer $k$ such that all pairs of nodes are joined by at least $k$ paths having no edges in common.
- **Node degree**: The largest integer $k$ such that each node has at least $k$ incident edges.
Node connectivity : 3

Edge connectivity : 3

Node degree : 3
• The proximity function $g_{h(k)}(C_r, C_s)$ between two clusters is defined in terms of
  – a proximity measure between vectors (nodes)
  – certain constraints imposed by property $h(k)$ on the subgraphs that are formed.
  – In other words,

$$g_{h(k)}(C_r, C_s) = \min_{x_u \in C_r, x_v \in C_s} \{d(x_u, x_v) \equiv a: \text{the } G(a) \text{ subgraph defined by } C_r \cup C_s \text{ is connected and either (I) has the the property } h(k)\text{or (II)is complete}\}$$

➢ **Graph theory-based algorithmic scheme (GTAS):** It is the GAS in the context of graph theory. In the context of GTAS, a pair of clusters $(C_i, C_j)$ is selected to be merged according to:

$$g_{h(k)}(C_i, C_j) = \min_{r, s} g_{h(k)}(C_r, C_s)$$
• **Single link (SL) algorithm.** Here

\[ g_{h(k)}(C_r, C_s) = \min_{x_u \in C_r, x_v \in C_s} \{d(x_u, x_v) \equiv a: \text{the } G(a) \text{ subgraph defined by } C_r \cup C_s \text{ is connected} \} \equiv \min_{x \in C_r, y \in C_s} d(x, y) \]

• **Remarks:**
  – No property \( h(k) \) or completeness is required.
  – The SL stemming from the graph theory is exactly the same with the SL stemming from the matrix theory.

• **Complete link (CL) algorithm.** Here

\[ g_{h(k)}(C_r, C_s) = \min_{x_u \in C_r, x_v \in C_s} \{d(x_u, x_v) \equiv a: \text{the } G(a) \text{ subgraph defined by } C_r \cup C_s \text{ is complete} \} \equiv \max_{x \in C_r, y \in C_s} d(x, y) \]

• **Remarks:**
  – No property \( h(k) \) is required.
  – The CL stemming from graph theory is exactly the same with the CL stemming from matrix theory.
Example 5: For the dissimilarity matrix,

\[
P = \begin{bmatrix}
0 & 1.2 & 3 & 3.7 & 4.2 \\
1.2 & 0 & 2.5 & 3.2 & 3.9 \\
3 & 2.5 & 0 & 1.8 & 2.0 \\
3.7 & 3.2 & 1.8 & 0 & 1.5 \\
4.2 & 3.9 & 2.0 & 1.5 & 0
\end{bmatrix}
\]

SL and CL produce the same hierarchy of clusterings at the levels given in the table.

<table>
<thead>
<tr>
<th>(R_0)</th>
<th>(R_1)</th>
<th>(R_2)</th>
<th>(R_3)</th>
<th>(R_4)</th>
<th>SL</th>
<th>CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>({x_1}, {x_2}, {x_3}, {x_4}, {x_5})</td>
<td>({x_1, x_2}, {x_3}, {x_4}, {x_5})</td>
<td>({x_1, x_2}, {x_3}, {x_4, x_5})</td>
<td>({x_1, x_2, x_3}, {x_4, x_5})</td>
<td>({x_1, x_2, x_3, x_4, x_5})</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.2</td>
<td>1.5</td>
<td>1.8</td>
<td>2.5</td>
<td>1.2</td>
<td>1.5</td>
<td>2.0</td>
</tr>
</tbody>
</table>
Remarks:

- SL poses the weakest possible graph condition (connectivity) for the formation of a cluster, while CL poses the strongest possible graph condition (completeness) for the formation of a cluster.
- A variety of graph theory-based algorithms, that lie between these two extremes result for various choices of $h(k)$.
  - For $k=1$ all these algorithms collapse to the single link algorithm.
  - As $k$ increases, the resulting subgraphs approach completeness.

Clustering algorithms based on the Minimum Spanning Tree (MST)

Definitions:

- Spanning Tree: It is a connected graph (containing all the vertices of the graph), with no loops (only one path connects any two vertices).
- Weight of a Spanning Tree: The sum of the weights of its edges (provided that they have been assigned with a weight).
- Minimum Spanning Tree (MST): A spanning tree with the smallest weight among the spanning trees connecting all the vertices of the graph.
Remarks:

- The MST has $N-1$ edges.
- When all the weights are different from each other, the MST is unique. Otherwise, it may not be unique.

Employing the GTAS and substituting $g_{h(k)}(C_r, C_s)$ with

$$g(C_r, C_s) = \min_{ij} \{ w_{ij} : x_i \in C_r, x_j \in C_s \}$$

where $w_{ij} = d(x_i, x_j)$, we can determine the MST.

Alternatively, a hierarchy of clusterings may be obtained by the MST as follows:

The clustering $\mathcal{R}_t$ at the $t$th level is the set of connected components of the MST, when only its $t$ smallest weights are considered.

Remark:
- The hierarchy produced by MST is the same with that produced by the single link algorithm.
Ties in the proximity matrix
- SL produces the same hierarchy of clusterings, independent of the order of consideration of edges with equal weights.
- CL may produce different hierarchies, depending on the order of consideration of edges with equal weights.
- The other graph theory-based algorithms behave as the CL.

Example 6: Let
\[
P = \begin{bmatrix}
0 & 4 & 9 & 6 & 5 \\
4 & 0 & 3 & 8 & 7 \\
9 & 3 & 0 & 3 & 2 \\
6 & 8 & 3 & 0 & 1 \\
5 & 7 & 2 & 1 & 0
\end{bmatrix}
\]
Note that \(P(2,3)=P(3,4)\).