CS724: Topics in Algorithms
Hierarchical Clustering Algorithms

Prof. Dan A. Simovici

CS724: Topics in Algorithms
Hierarchical Clustering Algorithms

Prof. Dan A. Simovici

Table of Contents

1. Ultrametrics, Hieararchies, and Partition Chains
2. The Minimum Spanning Tree of a Graph
3. Ultrametrics and Minimum Spanning Trees
4. The Single-Linkage Algorithm
5. Other Hierarchical Clustering Algorithms
6. Experiments with hierarchical clustering
7. Divisive Hierarchical Clustering

Ultrametrics defined on a finite set S and chains of equivalence relations on S (or chains of partitions on S) have a close relationship that we present next.

Theorem

Let S be a finite set and let $d : S \times S \longrightarrow \mathbb{R}_{\geqslant 0}$ be a function whose range is $Ran(d) = \{r_1, \ldots, r_m\}$, where $r_1 = 0$ such that $d(x, y) = 0$ if and only if $x = y$. Define the relations $\eta_{r_i} = \{(x, y) \in S \times S \mid d(x, y) \leqslant r_i\}$ for $1 \leqslant i \leqslant m$.

The function d is an ultrametric on S if and only if the sequence of relations $\eta_{r_1}, \ldots, \eta_{r_m}$ is an increasing chain of equivalences on S such that $\eta_{r_1} = \iota_S$ and $\eta_{r_m} = \theta_S = S \times S$.

Proof

Suppose that d is an ultrametric on S. We have $(x,x) \in \eta_{r_i}$ because $d(x,x) = 0$, so all relations η_{r_i} are reflexive. Also, it is clear that the symmetry of d implies $(x,y) \in \eta_{r_i}$ if and only if $(y,x) \in \eta_{r_i}$, so these relations are symmetric.

The ultrametric inequality is essential for proving the transitivity of the relations η_{r_i}. If $(x,y),(y,z) \in \eta_{r_i}$, then $d(x,y) \leqslant r_i$ and $d(y,z) \leqslant r_i$, which implies $d(x,z) \leqslant \max\{d(x,y), d(y,z)\} \leqslant r_i$. Thus, $(x,z) \in \eta_{r_i}$, which shows that every relation η_{r_i} is transitive and therefore an equivalence.

Proof (cont'd)

It is straightforward to see that $\eta_{r_1} \leqslant \eta_{r_2} \leqslant \cdots \leqslant \eta_{r_m}$; that is, this sequence of relations is indeed a chain of equivalences.

Conversely, suppose that $\eta_{r_1}, \ldots, \eta_{r_m}$ is an increasing sequence of equivalences on S such that $\eta_{r_1} = \iota_S$ and $\eta_{r_m} = \theta_S$, where $\eta_{r_i} = \{(x,y) \in S \times S \mid d(x,y) \leqslant r_i\}$ for $1 \leq i \leqslant m$ and $r_1 = 0$.

Note that $d(x,y) = 0$ is equivalent to $(x,y) \in \eta_{r_1} = \iota_S$, that is, to $x = y$.

Proof (cont'd)

We claim that for every $x, y \in S$ we have:

$$d(x, y) = \min\{r \mid (x, y) \in \eta_r\}.$$

Indeed, since $\eta_{r_m} = \theta_S$, it is clear that there is an equivalence η_{r_i} such that $(x, y) \in \eta_{r_i}$. If $(x, y) \in \eta_{r_i}$, the definition of η_{r_i} implies $d(x, y) \leqslant r_i$, so $d(x, y) \leqslant \min\{r \mid (x, y) \in \eta_r\}$. This inequality can be easily seen to become an equality since $(x, y) \in \eta_{d(x,y)}$. This implies immediately that d is symmetric.

To prove that d satisfies the ultrametric inequality, let x, y, z be three members of the set S. Let $p = \max\{d(x, z), d(z, y)\}$. Since $(x, z) \in \eta_{d(x,z)} \subseteq \eta_p$ and $(z, y) \in \eta_{d(z,y)} \subseteq \eta_p$, it follows that $(x, y) \in \eta_p$, due to the transitivity of the equivalence η_p. Thus, $d(x, y) \leqslant p = \max\{d(x, z), d(z, y)\}$, which proves the ultrametric inequality for d.

Previous theorem can be formulated in terms of partitions.

Theorem

Let S be a finite set and let $d : S \times S \longrightarrow \mathbb{R}_{\geqslant 0}$ be a function whose range is $Ran(f) = \{r_1, \ldots, r_m\}$, where $r_1 = 0$ such that $d(x, y) = 0$ if and only if $x = y$. For $u \in S$ and $r \in \mathbb{R}_{\geqslant 0}$, define the set
$D_{u,r} = \{x \in S \mid d(u, x) \leqslant r\}$.
Define the collection of sets

$$\pi_{r_i} = \{D(u, r_i) \mid u \in S\}$$

for $1 \leqslant i \leqslant m$. The function d is an ultrametric on S if and only if the sequence of collections $\pi_{r_1}, \ldots, \pi_{r_m}$ is an increasing sequence of partitions on S such that $\pi_{r_1} = \alpha_S$ and $\pi_{r_m} = \omega_S$.

Ultrametrics whose range is the set $\{0, 1\}$ are said to be *binary*.

Corollary

Let S be a finite set and let $d : S \times S \longrightarrow \mathbb{R}_{\geqslant 0}$ be a function whose range is $Ran(d) = \{0, 1\}$. Then, d is a binary ultrametric on S if and only if the relation

$$\rho = \{(x, y) \in S \times S \mid d(x, y) = 0\}$$

is an equivalence on S. Conversely, every equivalence ρ on a set S defines a binary ultrametric $u(\rho)$ given by

$$u(\rho)(x, y) = \begin{cases} 0 & \textit{if } (x, y) \in \rho, \\ 1 & \textit{otherwise.} \end{cases}$$

Definition

Let S be a set. A *hierarchy on the set S* is a pair $(S, \mathcal{H})$, where $\mathcal{H}$ is collection of sets $\mathcal{H} \subseteq \mathcal{P}(S)$ that satisfies the following conditions:

- the members of $\mathcal{H}$ are nonempty sets;
- $S \in \mathcal{H}$;
- for every $x \in S$, we have $\{x\} \in \mathcal{H}$;
- if $H, H' \in \mathcal{H}$, $H \neq H'$, and $H \cap H' \neq \emptyset$, then we have either $H \subset H'$ or $H' \subset H$.

Note that the last condition is equivalent to $H \cap H' \in \{H, H', \emptyset\}$ for every $H, H' \in \mathcal{H}$.

A standard technique for constructing a hierarchy on a set S starting with a rooted tree is given next.

Theorem

Let S be a set and let $T = (V, E, v_0)$ be a rooted tree. Define the mapping $\mu : V \longrightarrow \mathcal{P}(S)$ as follows:

- *the restriction of μ to L_T, the leaves of T is a bijection between L_T and the collection $\{\{s\} \mid s \in S\}$.*
- *if v is a vertex of T that has the immediate descendants $v_1, \ldots, v_m$, then $\mu(v) = \bigcup\{\mu(v_i) \mid 1 \leqslant i \leqslant m\}$.*

Then, $\mathcal{H} = \{\mu(v) \mid v \in V\}$ defines a hierarchy $(S, \mathcal{H})$ on the set S.

Proof

The set of labels $\mathcal{H}_T$ of the rooted tree $T = (V, E, v_0)$ defines a hierarchy $(S, \mathcal{H})$. Indeed, note that each singleton $\{x\}$ is a label of a leaf. An easy argument by induction on the height of the tree shows that every vertex is labelled by the set of labels of the leaves that descend from that vertex. Therefore, the root v_0 of the tree is labelled by S.

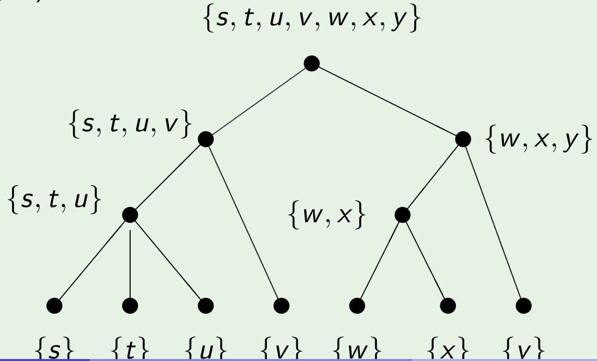
Suppose that H, H' are labels of the nodes u, v of T, respectively. If $H \cap H' \neq \emptyset$, then the vertices u, v have a common descendant. In a tree, this can take place only if u is a descendant of v or v is a descendant of u; that is, only if $H \subset H'$, or $H' \subset H$, respectively. This gives the desired conclusion.

Example

Let $S = \{s, t, u, v, w, x, y\}$ and let T be a tree whose vertices are labelled as shown next. It is easy to verify that the family of subsets $\mathcal{H}$ of S that label the nodes of T,

$$\begin{aligned}\mathcal{H} &= \{\{s\}, \{t\}, \{u\}, \{v\}, \{w\}, \{x\}, \{y\},\\ &\quad \{s, t, u\}, \{w, x\}, \{s, t, u, v\}, \{w, x, y\}, \{s, t, u, v, w, x, y\}\}\end{aligned}$$

is a hierarchy $(S, \mathcal{H})$.



Theorem

Let $S_1, \ldots, S_k$ be k pairwise disjoint sets such that $(S_1, \mathcal{H}_1), \ldots, (S_k, \mathcal{H}_k)$ are k hierarchies. If $S = \bigcup_{i=1}^{k} S_i$, then the pair $(S, \mathcal{H})$, where

$$\mathcal{H} = \{S\} \cup \bigcup_{i=1}^{k} \mathcal{H}_i$$

is a hierarchy on the set S.

Proof

It is immediate that $(S, \mathcal{H})$ satisfies the first three conditions of the definition.

Let now $H, H' \in \mathcal{H}$ such that $H \cap H' \neq \emptyset$. Since the sets $S_1, \ldots, S_k$ are pairwise disjoint, one of the following cases may occur:

- both H and H' belong to one of the families $\mathcal{H}_i$, or
- $H = S$ and there exists a collection $\mathcal{H}_i$ such that $H' \in \mathcal{H}_i$.

In the first case we have either $H \subseteq H'$ or $H' \subseteq H$ because $\mathcal{H}_i$ is a hierarchy. In the second case we have $H' \subseteq H$.

The hierarchy introduced in the Theorem is the *sum of the hierarchies* $(S_i, \mathcal{H}_i)$ for $1 \leqslant i \leqslant k$. We denote this hierarchy by $\sum_{i=1}^{k}(S_i, \mathcal{H}_i)$.

Example

For the hierarchy $\mathcal{H}$ defined in previously on the set $S = \{s, t, u, v, w, x, y\}$, the function $h : \mathcal{H} \longrightarrow \mathbb{R}$ given by

$$h(\{s\}) = h(\{t\}) = h(\{u\}) = h(\{v\}) = h(\{w\}) = h(\{x\}) = h(\{y\}) = 0,$$
$$h(\{s, t, u\}) = 3, h(\{w, x\}) = 4, h(\{s, t, u, v\}) = 5, h(\{w, x, y\}) = 6,$$
$$h(\{s, t, u, v, w, x, y\}) = 7,$$

is a grading function and $\mathrm{H} = (S, \mathcal{H}, h)$ is a graded hierarchy on S.

Define the relation "$\prec_{\mathcal{H}}$" on a hierarchy $(S, \mathcal{H})$ by $H \prec_{\mathcal{H}} K$ if $H, K \in \mathcal{H}$, $H \subset K$, and there is no set $L \in \mathcal{H}$ such that $H \subset L \subset K$.

Lemma

Let $\mathcal{H}$ be a hierarchy on a finite set S and let $L \in \mathcal{H}$. The collection $\mathcal{P}_L = \{H \in \mathcal{H} \mid H \prec_{\mathcal{H}} L\}$ is a partition of the set L.

Proof

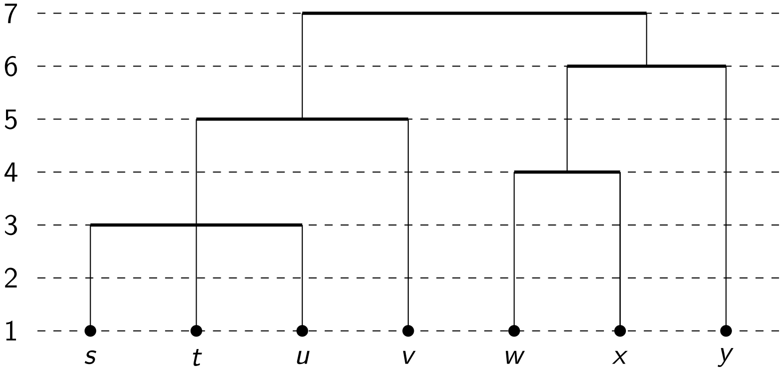
We claim that $L = \bigcup \mathcal{P}_L$. Indeed, it is clear that $\bigcup \mathcal{P}_L \subseteq L$. Conversely, suppose that $z \in L$ but $z \not\in \bigcup \mathcal{P}_L$. Since $\{z\} \in \mathcal{H}$ and there is no $K \in \mathcal{P}_L$ such that $z \in K$, it follows that $\{z\} \in \mathcal{P}_L$, which contradicts the assumption that $z \not\in \bigcup \mathcal{P}_L$. This means that $L = \bigcup \mathcal{P}_L$.
Let $K_0, K_1 \in \mathcal{P}_L$ be two distinct sets. These sets are disjoint since otherwise we would have either $K_0 \subset K_1$ or $K_1 \subset K_0$, and this would contradict the definition of $\mathcal{P}_L$.

Theorem

Let $(S, \mathcal{H})$ be a hierarchy on a set S. The graph of the relation $\prec_{\mathcal{H}}$ on $\mathcal{H}$ is a tree whose root is S; its leaves are the singletons $\{x\}$ for every $x \in S$.

We shall draw the tree of a graded hierarchy $(S, \mathcal{H}, h)$ using a special representation known as a *dendrogram*.

In a dendrogram, an interior vertex K of the tree is represented by a horizontal line drawn at the height $h(K)$.



A graded hierarchy defines an ultrametric, as shown next.

Theorem

Let $H = (S, \mathcal{H}, h)$ be a graded hierarchy on a finite set S. Define the function $d_H : S^2 \longrightarrow \mathbb{R}$ as $d_H(x, y) = \min\{h(U) \mid U \in \mathcal{H} \text{ and } \{x, y\} \subseteq U\}$ for $x, y \in S$. The mapping d_H is an ultrametric on S.

Proof

Observe that for every $x, y \in S$ there exists a set $H \in \mathcal{H}$ such that $\{x, y\} \subseteq H$ because $S \in \mathcal{H}$.
It is immediate that $d_{\mathrm{H}}(x, x) = 0$. Conversely, suppose that $d_{\mathrm{H}}(x, y) = 0$. Then, there exists $H \in \mathcal{H}$ such that $\{x, y\} \subseteq H$ and $h(H) = 0$. If $x \neq y$, then $\{x\} \subset H$, hence $0 = h(\{x\}) < h(H)$, which contradicts the fact that $h(H) = 0$. Thus, $x = y$.
The symmetry of d_{H} is immediate.

Proof (cont'd)

To prove the ultrametric inequality, let $x, y, z \in S$, and suppose that $d_{\mathrm{H}}(x, y) = p$, $d_{\mathrm{H}}(x, z) = q$, and $d_{\mathrm{H}}(z, y) = r$. There exist $H, K, L \in \mathcal{H}$ such that $\{x, y\} \subseteq H$, $h(H) = p$, $\{x, z\} \subseteq K$, $h(K) = q$, and $\{z, y\} \subseteq L$, $h(L) = r$. Since $K \cap L \neq \emptyset$ (because both sets contain z), we have either $K \subseteq L$ or $L \subseteq K$, so $K \cup L$ equals either K or L and, in either case, $K \cup L \in \mathcal{H}$. Since $\{x, y\} \subseteq K \cup L$, it follows that

$$d_{\mathrm{H}}(x, y) \leqslant h(K \cup L) = \max\{h(K), H(L)\} = \max\{d_{\mathrm{H}}(x, z), d_{\mathrm{H}}(z, y)\},$$

which is the ultrametric inequality.

We refer to the ultrametric d_{H} as the *ultrametric generated by the graded hierarchy $H = (S, \mathcal{H}, h)$*.

Example

The values of the ultrametric generated by the graded hierarchy $(S, \mathcal{H}, h)$ on the set S introduced before is

d	s	t	u	v	w	x	y
s	0	3	3	5	7	7	7
t	3	0	3	5	7	7	7
u	3	3	0	5	7	7	7
v	5	5	5	0	7	7	7
w	7	7	7	7	0	4	6
x	7	7	7	7	4	0	6
y	7	7	7	7	6	6	0

Example

The dendrogram of the graded hierarchy shows that S can be regarded as several unions of spheres in the ultrametric space:

$$\begin{aligned} S &= B[s,3] \cup B[v,1] \cup B[w,3] \cup B[y,6] \\ &= B[v,5] \cup B[y,6] = B[y,7] \end{aligned}$$

Theorem

Let (S, d) be a finite ultrametric space. There exists a graded hierarchy $H = (S, \mathcal{H}, h)$ on S such that d_H is the ultrametric associated to $H = (S, \mathcal{H}, h)$.

Proof

Let $\mathcal{H}$ be the collection of equivalence classes of the equivalences $\eta_r = \{(x, y) \in S^2 \mid d_{\mathsf{H}}(x, y) \leqslant r\}$ defined by the ultrametric d_{H} on the finite set S, where the index r takes its values in the range R_d of the ultrametric d. Define $h(E) = \min\{r \in R_d \mid E \in S/\eta_r\}$ for every equivalence class E.

It is clear that $h(\{x\}) = 0$ because $\{x\}$ is an η_0-equivalence class for every $x \in S$.

Proof (cont'd)

Let $[x]_t$ be the equivalence class of x relative to the equivalence η_t. Suppose that E and E' belong to the hierarchy and $E \subset E'$. We have $E = [x]_r$ and $E' = [x]_s$ for some $x \in X$. Since E is strictly included in E', there exists $z \in E' - E$ such that $d(x,z) \leqslant s$ and $d(x,z) > r$. This implies $r < s$. Therefore,

$$h(E) = \min\{r \in R_d \mid E \in S/\eta_r\} \leq \min\{s \in R_d \mid E' \in S/\eta_s\} = h(E'),$$

which proves that $(S, \mathcal{H}, h)$ is a graded hierarchy.

The ultrametric e generated by the graded hierarchy $\mathsf{H} = (S, \mathcal{H}, h)$ is given by

$$\begin{aligned} e(x,y) &= \min\{h(B) \mid B \in \mathcal{H} \text{ and } \{x,y\} \subseteq B\} \\ &= \min\{r \mid (x,y) \in \eta_r\} = \min\{r \mid d(x,y) \leqslant r\} = d(x,y), \end{aligned}$$

for $x, y \in S$; in other words, we have $e = d_{\mathsf{H}}$.

Example

Starting from the ultrametric on the set $S = \{s, t, u, v, w, x, y\}$ defined before we obtain the following quotient sets:

Values of r	S/η_r
$[0, 3)$	$\{s\}, \{t\}, \{u\}, \{v\}, \{w\}, \{x\}, \{y\}$
$[3, 4)$	$\{s, t, u\}, \{v\}, \{w\}, \{x\}, \{y\}$
$[4, 5)$	$\{s, t, u\}, \{v\}, \{w, x\}, \{y\}$
$[5, 6)$	$\{s, t, u, v\}, \{w, x\}, \{y\}$
$[6, 7)$	$\{s, t, u, v\}, \{w, x, y\}$
$[7, \infty)$	$\{s, t, u, v, w, x, y\}$

As we saw, $d_{\mathrm{H}}(x, y)$ generated by a graded hierarchy H is the smallest height of a set of a hierarchy that contains both x and y. This allows us to “read” the value of the ultrametric generated by H directly from the dendrogram of the hierarchy.

Let $G = (V, E, c)$ be a weighted graph. We seek to determine a subgraph $G' = (V, E', c')$ over the same set of vertices such that

- $E' \subseteq E$;
- $c'(x, y) = c(x, y)$ for $\{x, y\} \in E'$, and
- $\sum\{c(x, y) \mid \{x, y\} \in E'\}$ is minimal.

In other words, we seek a subgraph G' of G such that every vertex of G occurs in G and the total cost of G' is minimal.

Theorem

The set of edges of a connected weighted graph $G = (V, E, c)$ that achieves a minimal cost defines a connected and acyclic graph.

Proof.

Let $G' = (V, E', c')$ be the subgraph of G that achieves a minimal cost. Since each vertex must be connected to the other vertices, G' is a connected graph.

G' contains no cycles. Indeed, suppose that G' contains a cycle C and $e = (v_i, v_j)$ would be an edge on this cycle. Then, $(V, E' - \{e\}, c')$ is still connected because we could go from v_i to v_j using the remaining edges of this cycle and this would result into a graph with a lower cost. Thus, G' is both connected and acyclic. □

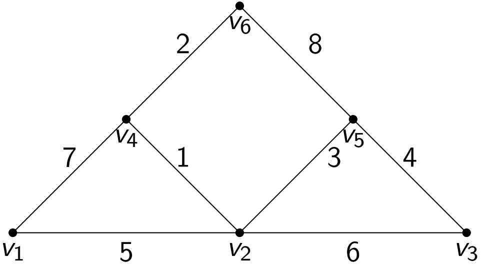
Thus, the graph of minimal cost is a tree (a connected and acyclic graph) and also, a spanning subgraph.

The *cost of a spanning tree* $T \in \mathsf{ST}(G)$ is the sum of the costs of its edges. We seek to determine spanning trees of a graph that have the lowest cost (the *minimum spanning tree problem*). We will use the acronym MST to refer to minimum spanning tree.
If the graph G is not connected we seek a *minimal spanning forest*, that is, a collection of spanning tree for each of its components such that the total cost of all these trees is minimal.

Kruskal's Algorithm:

Data: A weighted graph $G = (V, E, c)$
Result: A minimum spanning tree $T = (V, E', c')$ of G
initialize the set of edges U as $U \leftarrow \emptyset$
insert in U successive edges in the order of increasing weight *provided that the insertion does not create a cycle*; if it does, skip the edge;
stop when all nodes are connected
return: $T = (V, U, c \upharpoonright_U)$

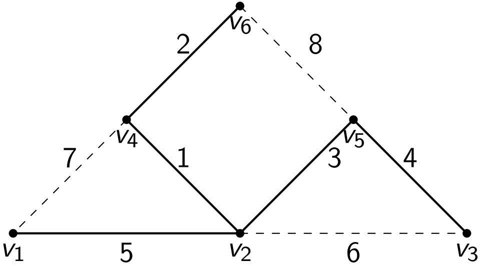
Let $G = (\{v_i \mid 1 \leqslant i \leqslant 6\}, E, c)$ a the weighted graph shown below.



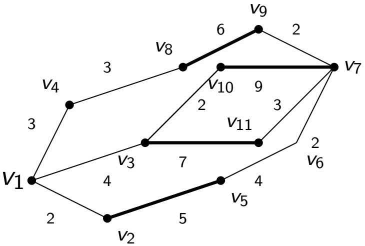
The successive values of the set U are:

$$
\begin{aligned}
&\emptyset\\
&\{\{v_2, v_4\}\}\\
&\{\{v_2, v_4\}, \{v_4, v_6\}\}\\
&\{\{v_2, v_4\}, \{v_4, v_6\}, \{v_2, v_5\}\}\\
&\{\{v_2, v_4\}, \{v_4, v_6\}, \{v_2, v_5\}, \{v_5, v_3\}\}\\
&\{\{v_2, v_4\}, \{v_4, v_6\}, \{v_2, v_5\}, \{v_5, v_3\}, \{v_2, v_1\}\}
\end{aligned}
$$

The weight of the minimum spanning tree shown is 15.



Suppose that a climber needs to climb a difficult terain. For example, to reach point v_7 starting from point v_1 in the map shown below



several paths are possible:

$$(v_1, v_4, v_8, v_9, v_7), (v_1, v_3, v_{10}, v_7), (v_1, v_3, v_{11}, v_7), (v_1, v_2, v_5, v_6, v_7).$$

The choice of a trail is dictated by the most difficult segment of each trail, and the climber chooses the trail having the least maximum difficulty as shown. Segments of maximum difficulty have been marked with a thick line. Thus, the trail with minimum maximal difficulty is $(v_1, v_2, v_5, v_6, v_7)$.

Lemma

Let $K = (V, E, c)$ *be a complete weighted graph having a positive weight function* c*. For* $\wp \in \mathit{paths}_K(u, v)$*, let*

$$M(\wp) = \max\{c(x, y) \mid \{x, y\} \text{ is an edge on } \wp\},$$

and $d(u, v) = \min\{M(\wp) \mid \wp \in \mathit{paths}_K(u, v)\}$*.*
The function $d : V \times V \longrightarrow \mathbb{R}_{\geqslant 0}$ *is an ultrametric on* V*.*

Proof

We have $d(u,u) = 0$ because the unique path between u and u is $\wp_0 = (u)$ and $M(\wp_0) = 0$. Conversely, if $d(u,v) = 0$, then $u = v$.
It is immediate that $d(u,v) = d(v,u)$ for $u, v \in S$. We claim that d satisfies the ultrametric inequality.
Let $u, v, w \in V$, $d(u,v) = p$, and $d(v,w) = q$. There exists a path $\wp \in paths_K(u,v)$ such that $d(u,v) = M(\wp)$ and $M(\wp) \leqslant M(\wp')$ for every path $\wp'$ that joins u and v. Similarly, there exists a path $\wp_1 \in paths_K(v,w)$ such that $d(v,w) = M(\wp_1)$ and $M(\wp_1) \leqslant M(\wp_1')$ for every path $\wp_1' \in paths_K(v,w)$.

Proof (cont'd)

Note that $\wp\wp_1$ is a path that joins u to w and the largest value of $c(x, y)$ for an edge $\{x, y\}$ on this path is $\max\{M(\wp), M(\wp_1)\} = \max\{d(u, v), d(v, w)\}$. Since $d(u, w)$ is the least of the maximal weights that occur on a path that joins u to w it follows that

$$d(u, w) \leqslant \max\{d(u, v), d(v, w)\}.$$

In other words, d is an ultrametric on V.

An extension of the lemma

Let $\mathcal{G} = (V, E, c)$ be a weighted graph. Define $\tilde{c} : V \times V \longrightarrow \hat{\mathbb{R}}_{\geqslant 0}$ as

$$\tilde{c}(u, v) = \begin{cases} c(u, v) & \text{if } \{u, v\} \in E, \\ \infty & \text{otherwise.} \end{cases}$$

Theorem

Let $\mathcal{G} = (V, E, c)$ be a weighted graph. For $\wp \in paths_K(u, v)$, let

$$M(\wp) = \max\{\tilde{c}(x, y) \mid \{x, y\} \textit{ is an edge on } \wp\},$$

and $d(u, v) = \min\{M(\wp) \mid \wp \in paths_K(u, v)\}$.
The function $d : V \times V \longrightarrow \hat{\mathbb{R}}_{\geqslant 0}$ is a quasi-ultrametric on V.

Proof

If there is no path between the vertices u, w, we have $d(u, w) = \infty$. Note that in this case, for every $v \in V$, either there is no path between u and v, or there is no path between v and w. In either case, the ultrametric inequality holds. If there is a path between u and w, the argument of the Lemma applies.

Corollary

Let $T = (V, E, c)$ be a weighted tree. Since each pair of distinct vertices v_i, v_j is connected by a unique path, we can define the function $d : V \times V \longrightarrow \mathbb{R}_{\geqslant 0}$ such that $d(v_i, v_j)$ is 0 if $v_i = v_j$ and $d(v_i, v_j)$ is the largest weight assigned to an edge on the path between v_i and v_j. Then d is an ultrametric on V.

Theorem

The ultrametric d_T generated by a minimum spanning tree T of a weighted graph $\mathcal{G} = (V, E, c)$ is the same for every MST of $\mathcal{G}$.

Proof

Let T and T' be two minimal spanning trees for $\mathcal{G}$ and let d, d' be the ultrametrics generated by T and T', respectively.
Suppose that there is a pair of vertices $\{x, y\}$ such that

$$d_T(x, y) < d'_T(x, y).$$

Removing from T' the most expensive edge that occurs on the path between x and y in this tree we get two connected components of T', that can be connected by some edge on a path from x to y in T. Thus, the cost of T' is reduced, which contradicts the minimality of T'.

Definition

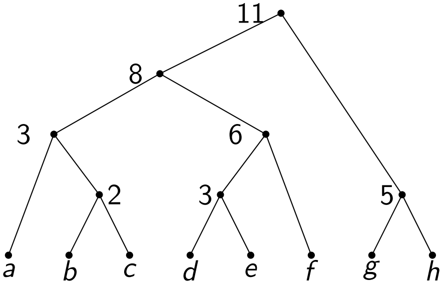
Let $D \in \mathbb{R}^{n \times n}$ be a matrix with non-negative entries. An *ultrametric tree* for D is a rooted tree $T = (V, E, v_0)$ that satisfies the following conditions:

- T contains n leaves, each labeled by a unique row in D;
- each internal node of T is labeled by one entry from D, and has at least two children;
- along any path from the root to a leaf, the numbers labeling the internal nodes strictly decrease;
- for any two leaves u, v of T, d_{uv} is the label of the closest common ancestor of u and v in T.

The rows of the matrix $D \in \mathbb{R}^{8\times 8}$ given below are labeled by a, b, c, d, e, f, g, h:

$$D = \begin{array}{c|cccccccc}
 & a & b & c & d & e & f & g & h \\
\hline
a & 0 & 3 & 3 & 8 & 8 & 8 & 11 & 11 \\
b & 3 & 0 & 2 & 8 & 8 & 8 & 11 & 11 \\
c & 3 & 2 & 0 & 8 & 8 & 8 & 11 & 11 \\
d & 8 & 8 & 8 & 0 & 3 & 6 & 11 & 11 \\
e & 8 & 8 & 8 & 3 & 0 & 6 & 11 & 11 \\
f & 8 & 8 & 8 & 6 & 6 & 0 & 11 & 11 \\
g & 11 & 11 & 11 & 11 & 11 & 11 & 0 & 5 \\
h & 11 & 11 & 11 & 11 & 11 & 11 & 5 & 0
\end{array}$$

An ultrametric tree that represents this matrix:



Since a binary tree with n leaves has $n-1$ interior nodes, the existence of an ultrametric tree for a matrix $D \in \mathbb{R}^{n \times n}$ implies that the matrix D has at most $n-1$ distinct non-zero entries.

Definition

An *ultrametric matrix* is a symmetric non-negative matrix $D \in \mathbb{R}^{n \times n}$ with $d_{ii} = 0$ for $1 \leqslant i \leqslant n$ and $d_{ij} \leqslant \max\{d_{ik}, d_{kj}\}$ for $1 \leqslant i, j, k \leqslant n$

Theorem

A symmetric matrix $D \in \mathbb{R}^{n \times n}$ has an ultrametric tree if and only if D is an ultrametric matrix.

Proof

Suppose that D has an ultrametric tree and consider the minimal subtree that contains the distinct leaves i, j, k. Let v be the closest common ancestor of i and j and let u be the closest common ancestor of i, j, k. Clearly, u is an ancestor of v, which means that $u > v$ because the numbers labeling the internal nodes strictly decrease along any path from the root to a leaf. The number at v is d_{ij}, while the number at u is $d_{ik} = d_{jk}$. Therefore, $d_{ik} \leqslant \max\{d_{ij}, d_{jk}\}$, so D is an ultrametric matrix.

Proof (cont'd)

Conversely, suppose that D is an ultrametric matrix. Note that $d_{ii} = 0 \leqslant d_{ij}$ for all $j \neq i$. If there are m non-zero distinct entries in the row i of D, then any ultrametric tree T for D must contain a path $\wp_i$ from i to the root with exactly m nodes and each node on this path must be labeled by one of the distinct m entries on row i; these labels must appear in decreasing order on the path. Thus, the nodes and the labels on the path to leaf i are determined only by the entries of the row i of D. Any internal node v on that path labeled by d_{ij} must be the closest common ancestor of leaves i and j. This determines where leaf j must occur in T relative to $\wp_i$. Thus, $\wp_i$ partitions the remaining $n-1$ nodes into $m-1$ classes resulting into a partition π_i. Leaves j and k are together in the same class of π_i if $d_{ij} = d_{ik}$. It follows that each block B_ℓ of π_i is defined by a distinct node in $\wp_i$.

The construction of an ultrametric tree can be done recursively, for each of the classes B_ℓ; after each of these trees is obtained they can be connected to yield the full ultrametric tree.
Let v be an internal node and let leaf j be contained in this class. Let k be some other leaf. Three cases may occur:

- k is in the same class as j;
- k is in a class contained between the leaf i and node v;
- k is in a class contained between node v and the root of the tree.

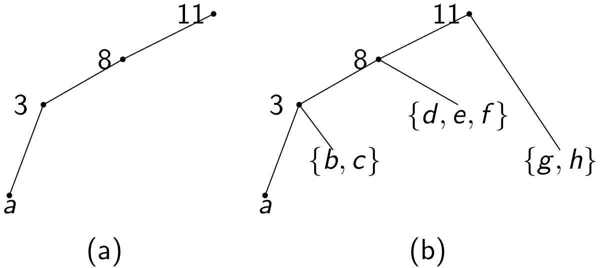
In the first case $d_{ij} = d_{ik}$, so $d_{jk} \leqslant d_{ij}$ because D is an ultrametric matrix. This means that if a subtree containing leaves j and k is attached to v, then d_{jk} is correctly represented in the new tree and the tree has the required property that node numbers stricly decrease along any path from the root.

In the second case $d_{ik} > d_{ij}$, so $d_{jk} = d_{ik}$. Therefore, if an ultrametric tree for the class containing j is connected at v, then d_{jk} will be correctly computed at the least common ancestor of j and k.

In the third case, $d_{ik} < d_{ij}$, so $d_{jk} = d_{ij}$ and v must be the least common ancestor of j and k.

Thus, in all cases, the ultrametric tree of the class defined by v can be correctly attached to v and the procedure correctly constructs an ultrametric tree for D.

We construct an ultrametric tree for the matrix D before. There are three distinct non-zero entries in the first row of D, namely, 3, 8, and 11.

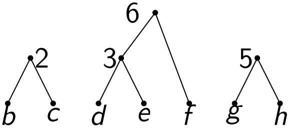


The construction begins with the path of length 3 shown in Figure (a).

The vertex sets that correspond to the vertices encountered an this path are $\{b, c\}, \{d, e, f\}$, and $\{g, h\}$ and their ultrametric matrices are:

$$\begin{array}{c|cc} & b & c \\ \hline b & 0 & 2 \\ c & 2 & 0 \end{array}, \qquad \begin{array}{c|ccc} & d & e & f \\ \hline d & 0 & 3 & 6 \\ e & 3 & 0 & 6 \\ f & 6 & 6 & 0 \end{array}, \qquad \begin{array}{c|cc} & g & h \\ \hline g & 0 & 5 \\ h & 5 & 0 \end{array}.$$

The ultrametric trees of these matrices are shown in Figures (a)-(c).



(a) (b) (c)

Single-linkage clustering begins by initializing each point as its own cluster, and then repeatedly merging the pair of clusters whose distance to one another (as measured from their closest points of approach) is minimal. The merging of clusters has a local behavior: the regions where the clusters are closest have a greater influence on this process than the global structure of the dissimilarity graph of the objects. The effect is to produce elongated clusters.

Single-linkage Clustering Algorithm

```
Data: A dissimilarity space $(S, d)$;
Result: A single-linkage clustering;
initialize $\pi \leftarrow \{\{x\} \mid x \in S\}$;
while {stopping condition is not met}
{ seek a pair of clusters $C, C' \in \pi$ such that
      $\delta(C, C') = \min\{d(x, y) \mid x \in C, y \in C'\}$
  is minimal;
  fuse the clusters $C$ and $C'$ into the cluster $C \cup C'$, that is,
  $\pi \leftarrow \pi - \{C, C'\} \cup \{C \cup C'\}$;
}
return $\pi$
```

The most common stopping condition, which we adopt unless specified otherwise is that $\pi = \omega_S$, that is, only one cluster exists.

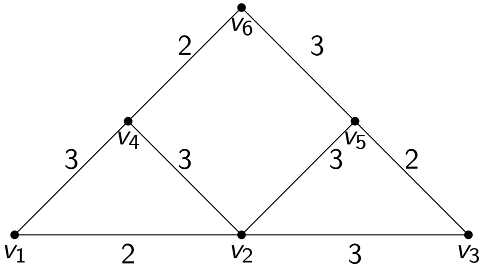
Other stopping conditions:

- k`-cluster stopping condition:` Stop adding edges when the partition first consists of k blocks. (This condition is well-defined when the number of points is at least k.)
- `dissimilarity-`r `stopping condition:` Only fuse clusters C, C' such that $\delta(C, C') \leqslant r$;
- `scale-`α `stopping condition:` Let d^* denote the maximum pairwise distance; i.e. $d^* = max\{d(i,j) \mid (i,j) \in V\}$. Only fuse clusters C, C' if $\delta(C, C') \leqslant \alpha d^*$.

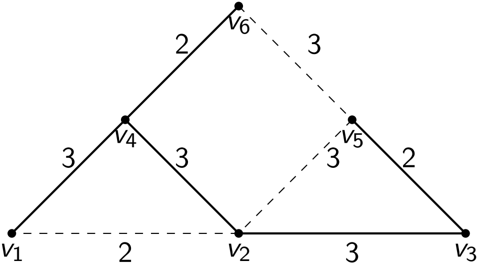
The single-linkage algorithm can be presented from the perspective of a minimum spanning tree of the weighted complete graph $\mathcal{G}_d$ whose vertex set is S and for which the weight of edge $\{i, j\}$ is $d(i, j)$.

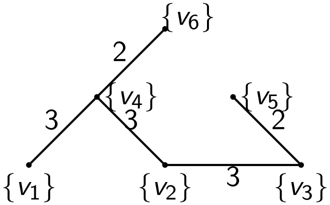
- The process starts with the partition of S that consists of singletons and from an MST T of the graph $\mathcal{G}_{S,d}$ labeled by these singletons.
- At each step the algorithm replaces edges in the tree by blocks obtained by fusing the extremities of the edges that have the lowest weight, until a single block partition is obtained.
- As before, the most common stopping condition, which we adopt unless specified otherwise is that $\pi = \omega_S$, that is, only one cluster exists.

Consider the graph

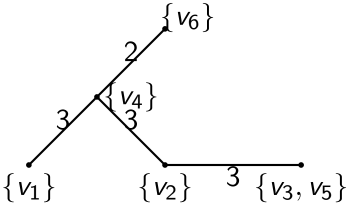


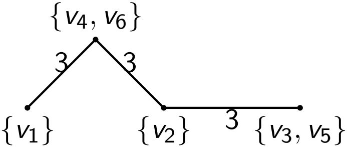
Starting from the MST tree for the previous graph the construction of the single-link clustering proceeds along the sequence of graphs next.

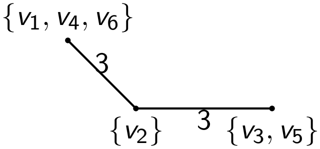


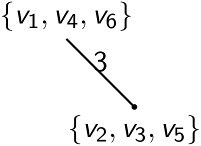


$\{v_6\}$
2
$\{v_4\}$
$\{v_5\}$
3
3
2
$\{v_1\}$
$\{v_2\}$
3
$\{v_3\}$


$\{v_6\}$
2
$\{v_4\}$
3
3
$\{v_1\}$
$\{v_2\}$
3
$\{v_3, v_5\}$


$\{v_4, v_6\}$
3
3
$\{v_1\}$
$\{v_2\}$
3
$\{v_3, v_5\}$


$\{v_1, v_4, v_6\}$
3
$\{v_2\}$
3
$\{v_3, v_5\}$


$\{v_1, v_4, v_6\}$
3
$\{v_2, v_3, v_5\}$

$$\{v_2, v_3, v_5, v_1, v_4, v_6\}$$

Hierarchical clustering can be regarded as a recursive process that begins with a metric space of objects (S, d) and results in a chain of partitions of the set of objects. In each of the partitions, similar objects belong to the same block and objects that belong to distinct blocks tend to be dissimilar. In agglomerative hierarchical clustering, the construction of this chain begins with the unit partition $\pi^1 = \alpha_S$. If the partition constructed at step k is

$$\pi^k = \{U_1^k, \ldots, U_{m_k}^k\},$$

then two distinct blocks U_p^k and U_q^k of this partition are selected using a *selection criterion*. These blocks are fused and a new partition

$$\pi^{k+1} = \{U_1^k, \ldots, U_{p-1}^k, U_{p+1}^k, \ldots, U_{q-1}^k, U_{q+1}^k, \ldots, U_p^k \cup U_q^k\}$$

is formed. Clearly, we have $\pi^k \prec \pi^{k+1}$. The process must end because the poset $(PART(S), \leqslant)$ is of finite height. The algorithm halts when the one-block partition ω_S is reached.

If two blocks U and V of a partition π are fused into a new block W to yield a new partition π' that covers π, then the variation of the sum of squared errors was shown to be

$$sse(\pi') - sse(\pi) = \frac{|U||V|}{|W|} \parallel \mathbf{c}_V - \mathbf{c}_U \parallel^2 .$$

In each phase of hierarchical clustering two of the "closest" clusters are merged. The notion of closest clusters is dependent on the specific dissimilarity between clusters considered in each variant of the clustering algorithm. If U and V are two clusters, the dissimilarity between them is be defined using one of the following real-valued, two-argument functions defined on the set of subsets of S:

$$\begin{aligned}
sl(U,V) &= \min\{d(u,v) | u \in U, v \in V\};\\
cl(U,V) &= \max\{d(u,v) | u \in U, v \in V\};\\
gav(U,V) &= \frac{\sum\{d(u,v) | u \in U, v \in V\}}{|U| \cdot |V|};\\
cen(U,V) &= \parallel \mathbf{c}_U - \mathbf{c}_V \parallel^2;\\
ward(U,V) &= \frac{|U||V|}{|U|+|V|} \parallel \mathbf{c}_V - \mathbf{c}_U \parallel^2 .
\end{aligned}$$

The names of the functions *sl*, *cl*, *gav*, and *cen* defined above are acronyms of the terms “single link”, “complete link”, “group average”, and “centroid”, respectively.

They are linked to variants of the hierarchical clustering algorithms that we discuss later. Note that in the case of the *ward* function the value equals the increase in the sum of the square errors when the clusters U, V are replaced with their union.

The specific selection criterion for fusing blocks defines the clustering algorithm. All algorithms store the dissimilarities between the current clusters $\pi^k = \{U_1^k, \ldots, U_{m_k}^k\}$ in an $m_k \times m_k$-matrix $D^k = (d_{ij}^k)$, where d_{ij}^k is the dissimilarity between the clusters U_i^k and U_j^k. As new clusters are created by merging two existing clusters, the distance matrix must be adjusted to reflect the dissimilarities between the new cluster and existing clusters.

The general form of the hierararchical clustering algorithm is

```
Data: the initial dissimilarity matrix $D^1$;
Result: the cluster hierarchy on the set of objects $S$, where $|S| = n$;
$k = 1$;
initialize clustering: $\pi^1 = \alpha_S$;
while $\{\pi^k$ contains more than one block$\}$
{   merge a pair of two of the closest clusters;
    output new cluster;
    $k++$;
    compute the dissimilarity matrix $D^k$;
}
```

To evaluate the space and time complexity of hierarchical clustering, note that the algorithm must handle the matrix of the dissimilarities between objects, and this is a symmetric $n \times n$-matrix having all elements on its main diagonal equal to 0; in other words, the algorithm needs to store $\frac{n(n-1)}{2}$ numbers. To keep track of the clusters, an extra space that does not exceed $n-1$ is required. Thus, the total space required is $O(n^2)$.

The computation of the dissimilarity between a new cluster and existing clusters is described next.

Theorem

Let U and V be two clusters of the clustering π that are joined into a new cluster W. Then, if $Q \in \pi - \{U, V\}$, we have:

$$
\begin{aligned}
sl(W,Q) &= \frac{1}{2}sl(U,Q) + \frac{1}{2}sl(V,Q) - \frac{1}{2}\Big|sl(U,Q) - sl(V,Q)\Big|;\\
cl(W,Q) &= \frac{1}{2}cl(U,Q) + \frac{1}{2}cl(V,Q) + \frac{1}{2}\Big|cl(U,Q) - cl(V,Q)\Big|;\\
gav(W,Q) &= \frac{|U|}{|U|+|V|}gav(U,Q) + \frac{|V|}{|U|+|V|}gav(V,Q);\\
cen(W,Q) &= \frac{|U|}{|U|+|V|}cen(U,Q) + \frac{|V|}{|U|+|V|}cen(V,Q)\\
&\quad -\frac{|U||V|}{(|U|+|V|)^2}cen(U,V);\\
ward(W,Q) &= \frac{|U|+|Q|}{|U|+|V|+|Q|}ward(U,Q) + \frac{|V|+|Q|}{|U|+|V|+|Q|}ward(V,Q)
\end{aligned}
$$

Proof

The first two equalities follow from the fact that

$$\begin{aligned} \min\{a, b\} &= \frac{1}{2}(a + b) - \frac{1}{2}|a - b|, \\ \max\{a, b\} &= \frac{1}{2}(a + b) + \frac{1}{2}|a - b|, \end{aligned}$$

for every $a, b \in \mathbb{R}$.

Proof (cont'd)

For the third equality, we have

$$
\begin{aligned}
& gav(W,Q) \\
&= \frac{\sum\{d(w,q) | w \in W, q \in Q\}}{|W| \cdot |Q|} \\
&= \frac{\sum\{d(u,q) | u \in U, q \in Q\}}{|W| \cdot |Q|} + \frac{\sum\{d(v,q) | v \in V, q \in Q\}}{|W| \cdot |Q|} \\
&= \frac{|U|}{|W|} \frac{\sum\{d(u,q) | u \in U, q \in Q\}}{|U| \cdot |Q|} + \frac{|V|}{|W|} \frac{\sum\{d(v,q) | v \in V, q \in Q\}}{|V| \cdot |Q|} \\
&= \frac{|U|}{|U| + |V|} gav(U,Q) + \frac{|V|}{|U| + |V|} gav(V,Q).
\end{aligned}
$$

Proof (cont'd)

The equality involving the function *cen* is immediate. The last equality can be easily translated into

$$
\begin{aligned}
cen(W,Q) &= \frac{|Q||W|}{|Q|+|W|} \parallel \mathbf{c}_Q - \mathbf{c}_W \parallel^2 \\
= & \quad \frac{|U|+|Q|}{|U|+|V|+|Q|}\frac{|U||Q|}{|U|+|Q|} \parallel \mathbf{c}_Q - \mathbf{c}_U \parallel^2 \\
& + \frac{|V|+|Q|}{|U|+|V|+|Q|}\frac{|V||Q|}{|V|+|Q|} \parallel \mathbf{c}_Q - \mathbf{c}_V \parallel^2 \\
& - \frac{|Q|}{|U|+|V|+|Q|}\frac{|U||V|}{|U|+|V|} \parallel \mathbf{c}_V - \mathbf{c}_U \parallel^2,
\end{aligned}
$$

which can be verified replacing $|W| = |U| + |V|$ and $\mathbf{c}_W = \frac{|U|}{|W|}\mathbf{c}_U + \frac{|V|}{|W|}\mathbf{c}_V$.

Corollary (The Lance-Williams Formula)

Let U and V be two clusters of the clustering π that are joined into a new cluster W. Then, if $Q \in \pi - \{U, V\}$, the dissimilarity between W and Q can be expressed as

$$d(W, Q) = a_U d(U, Q) + a_V d(V, Q) + bd(U, V) + c|d(U, Q) - d(V, Q)|,$$

where the coefficients a_U, a_V, b, c are given by the following table:

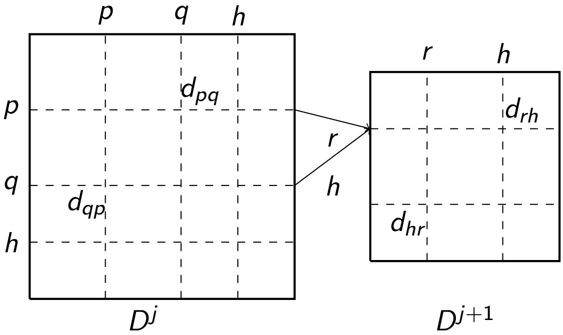
Function	a_U	a_V	b	c
sl	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
cl	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
gav	$\frac{\lvert U\rvert}{\lvert U\rvert+\lvert V\rvert}$	$\frac{\lvert V\rvert}{\lvert U\rvert+\lvert V\rvert}$	0	0
cen	$\frac{\lvert U\rvert}{\lvert U\rvert+\lvert V\rvert}$	$\frac{\lvert V\rvert}{\lvert U\rvert+\lvert V\rvert}$	$-\frac{\lvert U\rvert\lvert V\rvert}{(\lvert U\rvert+\lvert V\rvert)^2}$	0
ward	$\frac{\lvert U\rvert+\lvert Q\rvert}{\lvert U\rvert+\lvert V\rvert+\lvert Q\rvert}$	$\frac{\lvert V\rvert+\lvert Q\rvert}{\lvert U\rvert+\lvert V\rvert+\lvert Q\rvert}$	$-\frac{\lvert Q\rvert}{\lvert U\rvert+\lvert V\rvert+\lvert Q\rvert}$	0

The variant of the algorithm that makes use of the function *sl* is known as the *single-link* clustering. It tends to favor elongated clusters.
The *group average method*, which makes use of the *gav* function generates an intermediate approach between the single-link and the complete-link method.

Some of the methods mentioned so far have in common the *monotonicity property* expressed by the following statement.

Theorem

Let (S, d) be a finite metric space and let $D^1, \ldots, D^m$ be the sequence of matrices constructed by any of the first three hierarchical methods (single, complete, or average link), where $m = |S|$. If $\min D^j$ is the smallest entry of the matrix D^j for $1 \leqslant j \leqslant m$, then $\min D^1 \leqslant \min D^2 \leqslant \cdots \leqslant \min D^m$. In other words, the dissimilarity between clusters that are merged at each step is nondecreasing.


p
q
h
p
q
h
d_{pq}
d_{qp}
D^j
r
h
r
h
d_{rh}
d_{hr}
D^{j+1}

Proof

Let D^{j+1} be the matrix is obtained from the matrix D^j by merging the clusters C_p and C_q that correspond to the lines p and q and to columns p, q of D^j. This happens because $d_{pq} = d_{qp}$ is one of the minimal elements of the matrix D^j. Then, these lines and columns are replaced with a line and column that corresponds to the new cluster C_r and the dissimilarities between this new cluster and the previous clusters C_h, where $h \neq p, q$. The elements d_{rh}^{j+1} of the new line (and column) are obtained either as $\min\{d_{ph}^j, d_{qh}^j\}$, $\max\{d_{ph}^j, d_{qh}^j\}$, or $\frac{|C_p|}{|C_r|} d_{ph}^j + \frac{|C_q|}{|C_r|} d_{qh}^j$, for the single-link, complete-link, or group average methods, respectively. Note that any of the values that d_{rh}^{j+1} may take is greater or equal than $\min D^j$, so $\min D^{j+1} \geqslant \min D^j$.

The last two methods captured by the Lance-Williams formula are the centroid method and the Ward method of clustering.

As we observed before, the dissimilarity of two clusters in the case of Ward's method equals the increase in the sum of the squared errors that results when the clusters are merged.

The centroid method adopts the distance between the centroids as the distance between the corresponding clusters. Either method lacks the monotonicity properties.

We use the function setofpoints2 defined by

```
setofpoints2 <- function(n,center,stdev){
                              return(cbind(rnorm(n,center[1],stdev[1]),
                                             rnorm(n,center[2],stdev[2])))
}
```

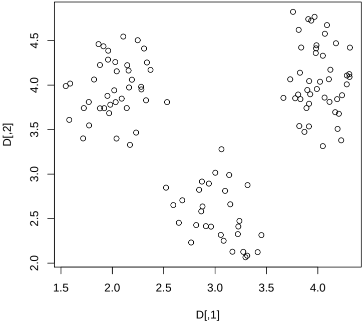
to generate n points in $\mathbb{R}^2$ normally distributed around the vector center and having the standard deviations specified by the vector stdev.

We begin by producing three sets of points A, B and D, and, then by joining these sets into the set D and naming the columns of this matrix as x and y:

```
A <- setofpoints2(40,c(2,4),c(0.2,0.3))
B <- setofpoints2(30,c(3,2.5),c(0.3,0.3))
C <- setofpoints2(45,c(4,4),c(0.2,0.4))
D <- rbind(A,B,C)
```

Then, the columns of D are named "x" and "y" by assigning to colnames(D) the array c("x","y").

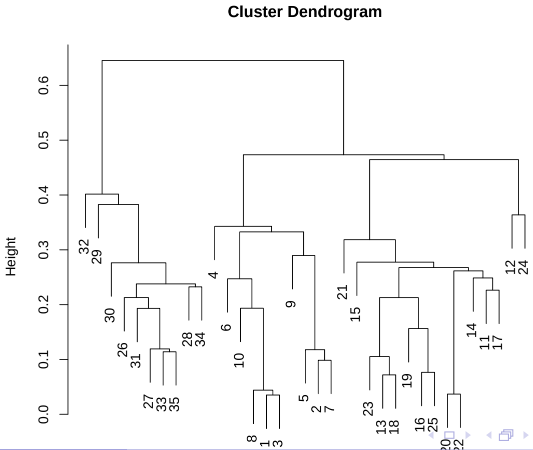
The plot of the set D is shown here



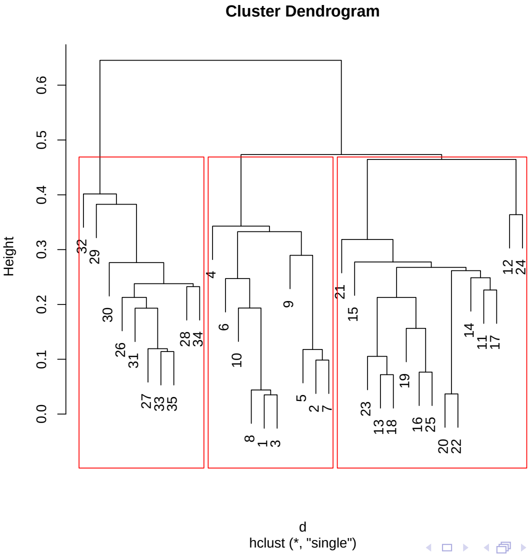
Starting from the matrix S a `dist` object is produced by `d<-dist(S)`. Next, the function `hclust` is applied in order to produce the single-link hierarchical clustering `sLink`:

```
sLink <- hclust(d,method=''single'')
```

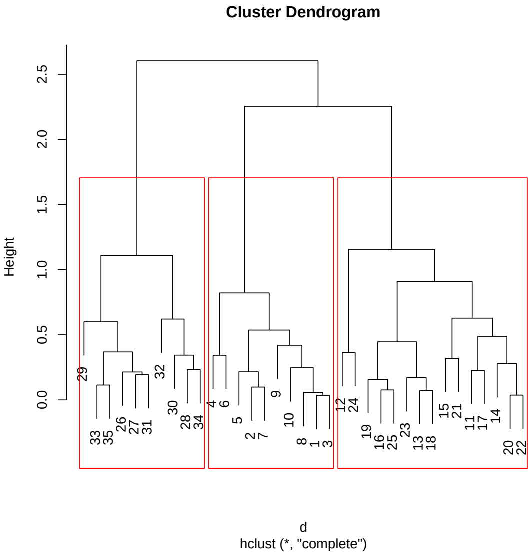
The dendrogram of the clustering is visualized using `plot(sLink)` and its representation is shown in below:



To obtain three clusters, the dendrogram is “cut” at an appropriate level using the function call `rect.hclust(sLink,3)` which generates the representation shown below:

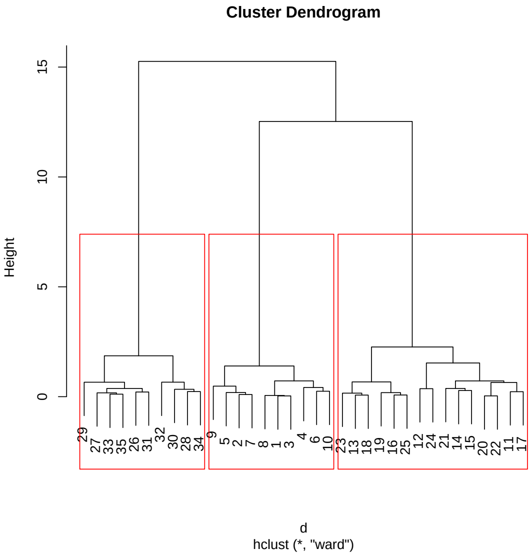


A clustering obtained by the application of the complete-link is:



Clusters delimited by rectangles obtained through the complete-link method.

The Ward method produces the following clustering:



Note that at the leaf-level, when clusters are sigletons, all methods produce exactly the same result. At higher levels the results diverge. The vertical axis shows the fusion level.

Divisive hierarchical clustering begins with a set of objects and recursively splits it to build a clustering.

- If there is no need to generate a complete hierarchy it may be more efficient that the previously discussed hierarchical algorithms.
- Since a divisive approach begins with the entire set of objects, it may be possible to take into account global characteristics of the set of objects.

The initial cluster to be split equals to the entire set of objects. During the evolution of a divisive algorithm, once a cluster C to be selected to be split we have several choices of bipartitions $\pi = \{C', C''\}$ of C. The choice of π can be made using the same criteria as in the case of agglomerative clustering: single-link, complete link, average link, or a variant of the Ward criterion.

The main issue of divisive hierarchical clustering is the difficulty of choosing a bipartition. Note that

- an agglomerative hierarchical algorithm applied to a set of n objects has $\binom{n}{2}$ choices for creating a two-object cluster;
- a divisive hierarchical algorithm that considers all bipartitions has $2^{n-1} - 1$ choices, a considerably larger number.

The DIANA Algorithm

DIANA (an acronym for DIvisive ANAlysis) is a divisive clustering algorithm introduced by Kaufman and Rousseeuw.
The algorithm splits sets according to an iterative process which mimmics the way a political party would split: first the most discontent member of the party leaves and starts a new party, then some other follow him until an equilibrium is achieved.

Definition

For a subset T of a dissimilarity space (S, d) and an object $t \in T$ the *average dissimilarity of t in T* is

$$\mathrm{ad}_T(t) = \frac{1}{|T| - 1} \sum \{d(t, u) \mid u \in T - \{t\}\}.$$

To split a set T we construct a sequence of pairs of sets $(U_0, V_0), (U_1, V_1), \ldots$ such that $U_0 = \emptyset$ and $V_0 = T$. For $i \geqslant 1$ each pair (U_i, V_i) is a bipartition of T.

Suppose that we constructed the bipartition $\{U_i, V_i\}$ and that $|V_i| > 1$. For each $o \in V_i$ we compute

$$d_i(o) = \mathrm{ad}_{V_i - \{o\}}(o) - ad_{U_i}(o)$$

for each object o of V_i. Let $o_i = \arg\max_x d_i(o)$ be the object that maximizes this difference.
If $d_i(o_i) > 0$ we move o_i from V_i to U_i, that is,

$$U_{i+1} = U_i \cup \{o_i\} \text{ and } V_{i+1} = V_i - \{o_i\}.$$

When $d_i(o) \leqslant 0$ is negative for all objects $o \in V_i$ we stop the process and the partitioning of T is completed.

The splitting process at each step involves a cluster that has the largest diameter and consists in initiating a “splinter group” and then in reassigning those objects closer to the splinter group” than to the remaining objects in the cluster.

The *divisive coefficient* measures the clustering structure of the dataset. For each observation u, denote by $d(u)$ the diameter of the last cluster to which it belongs (before being split off as a single observation), divided by the diameter of the whole dataset. The d_c is the average of all numbers $1 - d(u)$. Because d_c grows with the number of observations, this measure should not be used to compare datasets of very different sizes.

Example

Consider the set of objects $T = \{x, y, z, u, v\}$ and the matrix of dissimilarities defined as

$$D = \begin{array}{c|ccccc} & x & y & z & u & v \\ \hline x & 0 & 7 & 4 & 7 & 8 \\ y & 7 & 0 & 9 & 3 & 4 \\ z & 4 & 9 & 0 & 8 & 10 \\ u & 7 & 3 & 8 & 0 & 5 \\ v & 8 & 4 & 10 & 5 & 0 \end{array}$$

The average dissimilarities are:

object o	x	y	z	u	v
$\text{ad}_T(o)$	6.5	5.75	7.75	5.75	6.75

The object having the largest average dissimilarity is z, so $U_1 = \{z\}$ and $V_1 = \{x, y, u, v\}$.

Example

Next, we compute the function $d_1(o)$ for $o \in V_1$:

object o	x	y	u	v
$\mathrm{ad}_{V_1-\{o\}}(o)$	7.33	4.66	5.33	5.66
$\mathrm{ad}_{U_1}(o)$	4	9	8	10
$d_1(o)$	3.33	-4.34	-2.67	-4.34

Thus, x is shifted from the second set to the first and we have $U_2 = \{x, z\}$ and $V_2 = \{y, u, v\}$.

Example

In the next phase we compute $d_2(o)$ for $o \in V_2$:

object o	y	u	v
$\text{ad}_{V_2-\{o\}}(o)$	3.5	4	4.5
$\text{ad}_{U_2}(o)$	8	7.5	9
$d_2(o)$	-4.5	-3.5	-4.5

Since all values of $d_2(o)$ are negative the process halts and we have the partition $\{\{x, z\}, \{y, u, v\}\}$ of T.

Example

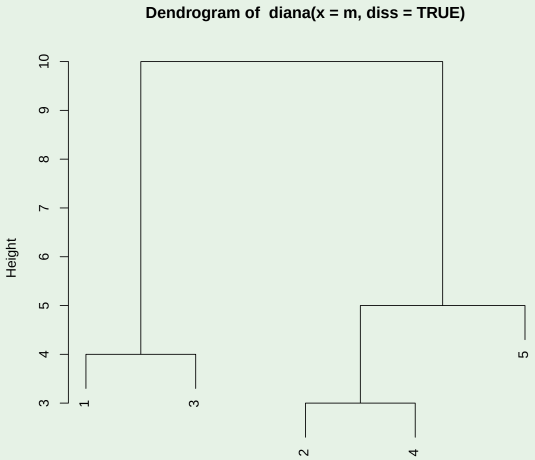
The diameters of these clusters are $diam(\{x, z\}) = 4$ and $diam(\{y, u, v\}) = 5$. Thus, in the next phase we split the cluster $V_3 = \{y, u, v\}$. The function $d_3(o)$ is computed next:

object o	y	u	v
$\text{ad}_{V_3 - \{o\}}(o)$	3.5	4	4.5

The object that has the largest average dissimilarity is v, so V_3 splits into the clusters $U_4 = \{v\}$ and $V_4 = \{y, u\}$. Note that two-element clusters are split automatically into one-element clusters.

Example

The result of applying the divisive clustering algorithm is shown below where the numbers $1, \ldots, 5$ correspond to the elements x, y, z, u, v, respectively.



DIANA is implemented in the package cluster. The syntax of the **R** command is

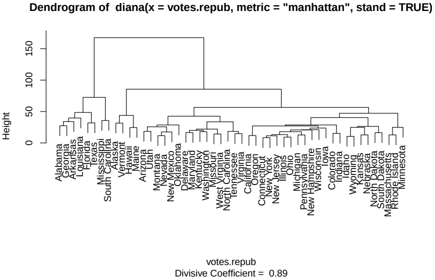
```
diana(x, diss, metric = ...,
      stand = FALSE, stop.at.k = FALSE,
      keep.diss = n < 100, keep.data = !diss, trace.lev = 0)
```

x is a data matrix, a data frame, a dissimilarity matrix or object depending on the value of the diss argument.

A matrix or a data frame is treated like a sample matrix if the logical variable diss is FALSE. If diss is set to TRUE then x is regarded as a dissimilarity matrix.

- The metric to be used can be Euclidean or the d_1 metric (the Manhattan metric); these choices are codified as `euclidean` or `manhattan`, respectively.
- If the logical parameter `stand` is set to `TRUE`, the measurements of `x` are standardized before calculating the dissimilarities. Recall that this standardization takes place for each variable (column) by subtracting the mean value and dividing by the variable standard deviation. If `x` is already a dissimilarity matrix this parameter is ignored.

A data frame named `votes.repub` contains the percents of votes given to the republican candidate in presidential elections from 1856 to 1976. Rows represent the fifty states, and columns the 31 elections.


Dendrogram of diana(x = votes.repub, metric = "manhattan", stand = TRUE)
Height
Alabama
Georgia
Arkansas
Louisiana
Florida
Texas
Mississippi
South Carolina
Alaska
Vermont
Hawaii
Maine
Arizona
Utah
Montana
Nevada
New Mexico
Oklahoma
Delaware
Maryland
Kentucky
Washington
Missouri
West Virginia
North Carolina
Tennessee
Virginia
California
Oregon
Connecticut
New York
New Jersey
Illinois
Ohio
Michigan
Pennsylvania
New Hampshire
Wisconsin
Iowa
Colorado
Indiana
Idaho
Wyoming
Kansas
Nebraska
North Dakota
South Dakota
Massachusetts
Rhode Island
Minnesota
votes.repub
Divisive Coefficient = 0.89