

Section 2

Unsupervised Learning: Hierarchical Clustering



Universitat
de les Illes Balears

Departament
de Ciències Matemàtiques
i Informàtica

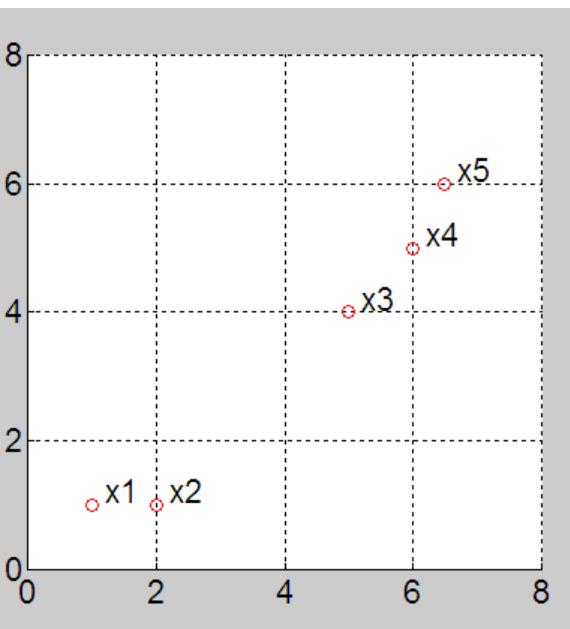
11752 Aprendizaje Automático
11752 Machine Learning
Máster Universitario
en Sistemas Inteligentes

Alberto ORTIZ RODRÍGUEZ

- Introduction
- Agglomerative clustering
- Divisive clustering
- Selection of a good clustering

Introduction

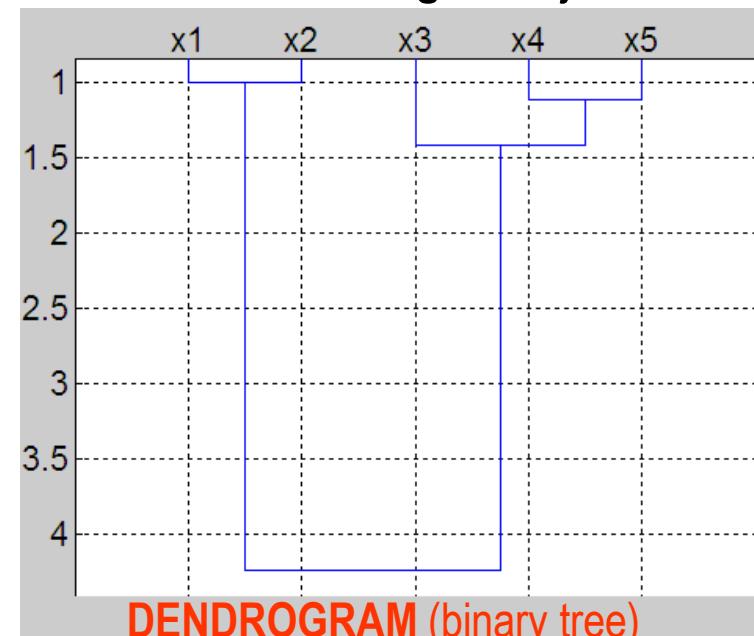
- Given a set of samples X , HC algorithms produce a **set of clusterings**, not just one



$$\begin{aligned}R_0 &: \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \\R_1 &: \{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \\R_2 &: \{x_1, x_2\}, \{x_3\}, \{x_4, x_5\} \\R_3 &: \{x_1, x_2\}, \{x_3, x_4, x_5\} \\R_4 &: \{x_1, x_2, x_3, x_4, x_5\}\end{aligned}$$

Actually, it is a **hierarchy of clusterings**, as they can be considered **nested**:

$$R_0 \subset R_1 \subset R_2 \subset R_3 \subset R_4 \subset R_5$$



- N samples $\Rightarrow N$ -level hierarchy – N execution steps
- Essentially, two sorts of hierarchical clustering algorithms:
 - Agglomerative** — $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \rightarrow \{x_1, x_2, x_3, x_4, x_5\}$ (bottom-up process)
 - Divisive** — $\{x_1, x_2, x_3, x_4, x_5\} \rightarrow \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}$ (top-down process)
 - both are just heuristic, **not optimal**, i.e. they do not optimize any objective function
 - a hierarchy of clusterings is produced even if **there is no structure** in the data

- Introduction
- Agglomerative clustering
- Divisive clustering
- Selection of a good clustering

Agglomerative clustering

- Generic algorithm:

(1) $\mathcal{R}_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$

(2) $t = 0$

(3) **repeat**

(3.1) Choose the pair of **clusters** $(C_r, C_s) \in \mathcal{R}_t$ ($r \neq s$) such that

$$\wp(C_r, C_s) = \begin{cases} \min_{i,j}\{\wp(C_i, C_j)\} & \wp \text{ is DM} \\ \max_{i,j}\{\wp(C_i, C_j)\} & \wp \text{ is SM} \end{cases}$$

(3.2) $\mathcal{R}_{t+1} = (\mathcal{R}_t - \{C_r, C_s\}) \cup \{C_r \cup C_s\}$

(3.3) $t = t + 1$

until all samples are in a single **cluster**

OBSERVATIONS:

- when two samples get in the same cluster, they keep together until the end
- there is no way to recover from a bad merge

Agglomerative clustering

- At the level t , there are $N - t$ clusters. therefore, one has to analyze:

$$\binom{N-t}{2} = \frac{(N-t)!}{2!(N-t-2)!} = \frac{(N-t)(N-t-1)}{2}$$

clusters to find the best merge for level $t + 1$.

- The number of merges which have to be considered up to the end of the process can be easily calculated:

$$\sum_{t=0}^{N-1} \binom{N-t}{2} = \sum_{k=1}^N \binom{k}{2} = \sum_{k=1}^N \frac{k(k-1)}{2} = \frac{1}{2} \left(\sum_{k=1}^N k^2 - \sum_{k=1}^N k \right)$$

$$= \frac{1}{2} \left(\frac{N(N-1)(2N-1)}{12} - \frac{N(N-1)}{2} \right)$$

$$= \mathcal{O}(N^3)$$

$$\begin{aligned} t &= 0, \dots, N-1 \\ \Rightarrow k &= N-t = 1, 2, \dots, N \end{aligned}$$

This gives an idea of the complexity of the process.

Agglomerative clustering

- From an implementation point of view, there are two main approaches of agglomerative clustering:



- based on **matrix** concepts
- based on **graph** theory

– We will consider the first approach. Some previous concepts first:

- data matrix:**

$$D(X) = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1L} \\ x_{21} & x_{22} & \dots & x_{2L} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NL} \end{bmatrix} \quad \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{matrix}$$

- proximity matrix:**

(e.g. dissimilarity
matrix)

– point of departure
for dendrogram
construction

$$P(X) = \begin{bmatrix} \wp(x_1, x_1) & \wp(x_1, x_2) & \dots & \wp(x_1, x_N) \\ \wp(x_2, x_1) & \wp(x_2, x_2) & \dots & \wp(x_2, x_N) \\ \vdots & \vdots & & \vdots \\ \wp(x_N, x_1) & \wp(x_N, x_2) & \dots & \wp(x_N, x_N) \end{bmatrix}$$

Agglomerative clustering

- **Example:**

- given $X = \{x_1, x_2, x_3, x_4, x_5\}$ such that $D(X) =$

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{bmatrix} \quad \sqrt{(5-1)^2 + (4-1)^2} = 5$$

- [1] Using the **Euclidean distance**, the proximity matrix (**dissimilarity**) is
 - notice the diagonal elements are **0**

$$P(X) = \begin{bmatrix} 0.00 & 1.00 & 5.00 & 6.40 & 7.43 \\ 1.00 & 0.00 & 4.24 & 5.66 & 6.73 \\ 5.00 & 4.24 & 0.00 & 1.41 & 2.50 \\ 6.40 & 5.66 & 1.41 & 0.00 & 1.12 \\ 7.43 & 6.73 & 2.50 & 1.12 & 0.00 \end{bmatrix}$$

- [2] Using the **Tanimoto measure**, the proximity matrix (**similarity**) is
 - notice the diagonal elements are **1**

$$P(X) = \begin{bmatrix} 1.00 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1.00 & 0.44 & 0.35 & 0.30 \\ 0.26 & 0.44 & 1.00 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1.00 & 0.98 \\ 0.18 & 0.30 & 0.90 & 0.98 & 1.00 \end{bmatrix}$$

Agglomerative clustering

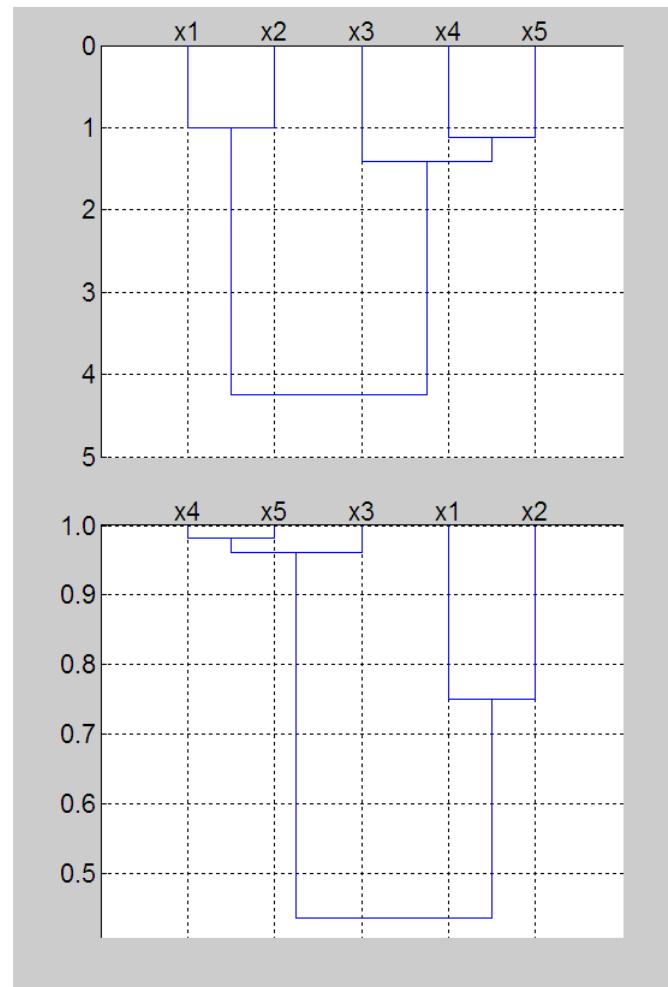
- **Example:** dendograms for $\wp(C_1, C_2) = \min/\max_{a \in C_1, b \in C_2} \wp(a, b)$

[1] proximity function: **Euclidean distance** (d_2)

$$P(X) = \begin{bmatrix} 0.00 & 1.00 & 5.00 & 6.40 & 7.43 \\ 1.00 & 0.00 & 4.24 & 5.66 & 6.73 \\ 5.00 & 4.24 & 0.00 & 1.41 & 2.50 \\ 6.40 & 5.66 & 1.41 & 0.00 & 1.12 \\ 7.43 & 6.73 & 2.50 & 1.12 & 0.00 \end{bmatrix}$$

[2] proximity function: **Tanimoto similarity** (s_T)

$$P(X) = \begin{bmatrix} 1.00 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1.00 & 0.44 & 0.35 & 0.30 \\ 0.26 & 0.44 & 1.00 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1.00 & 0.98 \\ 0.18 & 0.30 & 0.90 & 0.98 & 1.00 \end{bmatrix}$$



**dissimilarity
dendrogram**



the sequence
of mergings is
different because
 $d_2 \neq 1 - s_T$!!



**similarity
dendrogram**

Agglomerative clustering

- Example: dendrogram for

$$\wp(C_1, C_2) = \min_{a \in C_1, b \in C_2} \wp(a, b)$$

$$P_0 = \begin{bmatrix} 0.00 & 1.00 & 5.00 & 6.40 & 7.43 \\ 1.00 & 0.00 & 4.24 & 5.66 & 6.73 \\ 5.00 & 4.24 & 0.00 & 1.41 & 2.50 \\ 6.40 & 5.66 & 1.41 & 0.00 & 1.12 \\ 7.43 & 6.73 & 2.50 & 1.12 & 0.00 \end{bmatrix}$$

$\rightarrow C_6 = \{x_1, x_2\}$

$$P_1 = \begin{bmatrix} 0.00 & 1.00 & 5.00 & 6.40 & 7.43 & -- \\ 1.00 & 0.00 & 4.24 & 5.66 & 6.73 & -- \\ 5.00 & 4.24 & 0.00 & 1.41 & 2.50 & 4.24 \\ 6.40 & 5.66 & 1.41 & 0.00 & 1.12 & 5.66 \\ 7.43 & 6.73 & 2.50 & 1.12 & 0.00 & 6.73 \\ -- & -- & 4.24 & 5.66 & 6.73 & 0.00 \end{bmatrix}$$

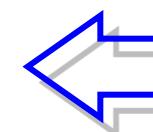
(a) $d(C_3, C_6)$
 (b) $d(C_4, C_6)$
 (c) $d(C_5, C_6)$

$\rightarrow C_7 = \{x_4, x_5\}$

(a) = $\min\{d(x_3, x_1)5.00, d(x_3, x_2)4.24\}$

(b) = $\min\{d(x_4, x_1)6.40, d(x_4, x_2)5.66\}$

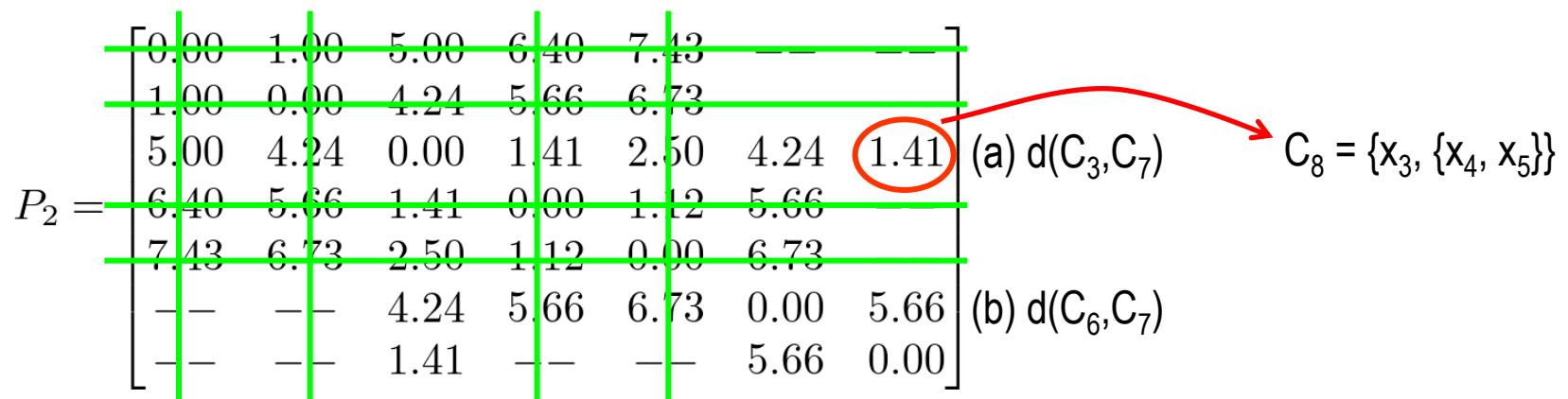
(c) = $\min\{d(x_5, x_1)7.43, d(x_5, x_2)6.73\}$



if this information
is available from P_0 !!

Agglomerative clustering

- **Example:** dendrogram for $\wp(C_1, C_2) = \min_{a \in C_1, b \in C_2} \wp(a, b)$

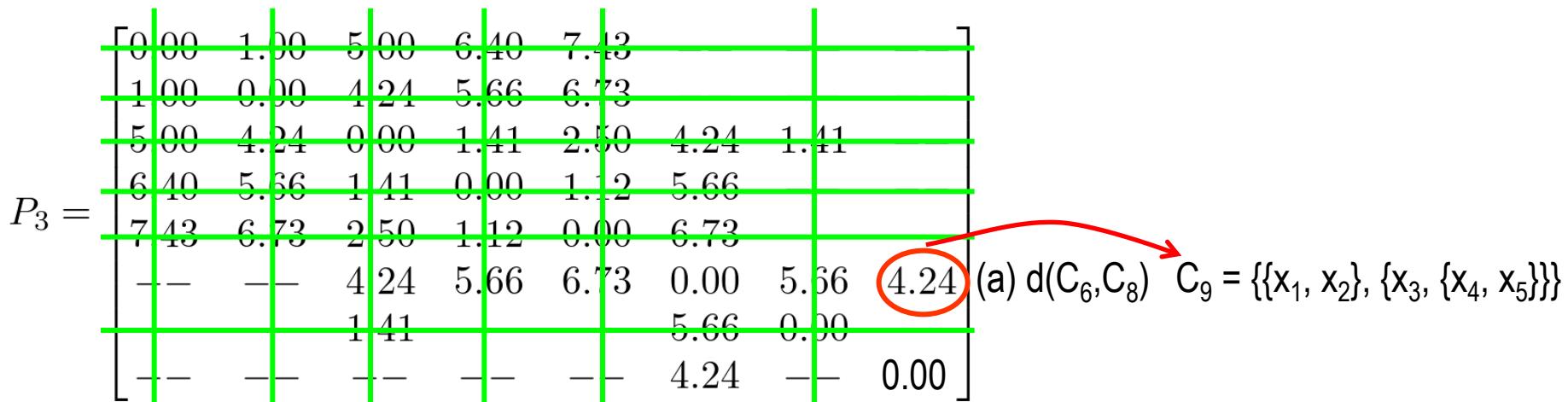


$$(a) = \min\{d(x_3, x_4) 1.41, d(x_3, x_5) 2.50\}$$

$$(b) = \min\{d(x_1, x_4) 6.40, d(x_1, x_5) 7.43, d(x_2, x_4) 5.66, d(x_2, x_5) 6.73\}$$

Agglomerative clustering

- **Example:** dendrogram for $\wp(C_1, C_2) = \min_{a \in C_1, b \in C_2} \wp(a, b)$



$$(a) = \min\{d(x_1, x_3)5.00, d(x_1, x_4)6.40, d(x_1, x_5)7.43, \\ d(x_2, x_3)4.24, d(x_2, x_4)5.66, d(x_2, x_5)6.73\}$$

Agglomerative clustering

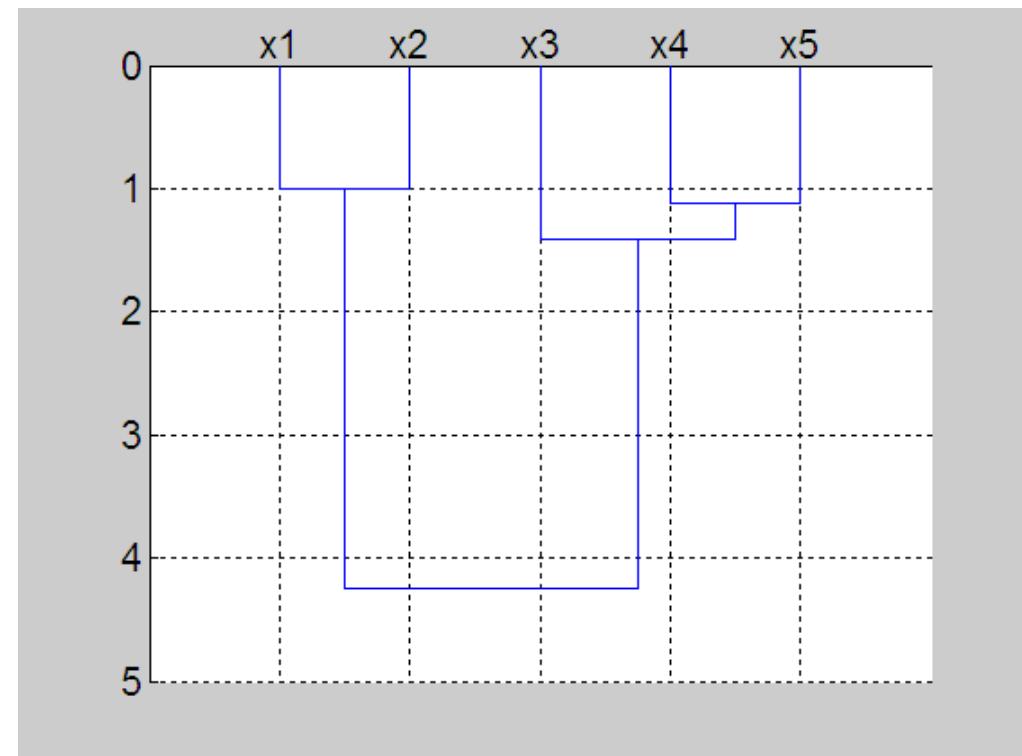
- **Example:** dendrogram for $\wp(C_1, C_2) = \min_{a \in C_1, b \in C_2} \wp(a, b)$

$$C_6 = \{x_1, x_2\} (1.00)$$

$$C_7 = \{x_4, x_5\} (1.12)$$

$$C_8 = \{x_3, \{x_4, x_5\}\} (1.41)$$

$$C_9 = \{\{x_1, x_2\}, \{x_3, \{x_4, x_5\}\}\} (4.24)$$



Agglomerative clustering

- In this way, the clustering algorithm turns out to be as follows:

(1) $\mathcal{R}_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$

(2) $t = 0$

(3) **repetir**

(3.1) Choose the pair of **clusters** $(C_r, C_s) \in \mathcal{R}_t$ ($r \neq s$) such that:

$$P_t(r, s) = \begin{cases} \min_{i,j} P_t(i, j) & P_t \text{ is DM} \\ \max_{i,j} P_t(i, j) & P_t \text{ is SM} \end{cases}$$

(3.2) $\mathcal{R}_{t+1} = (\mathcal{R}_t - \{C_r, C_s\}) \cup \{C_r \cup C_s\}$

(3.3) Calculate P_{t+1} from P_t :

$$P_{t+1} = P_t$$

remove row and column r of P_{t+1}

remove row and column s of P_{t+1}

open new row and column P_{t+1} for $C_q = \{C_r \cup C_s\}$

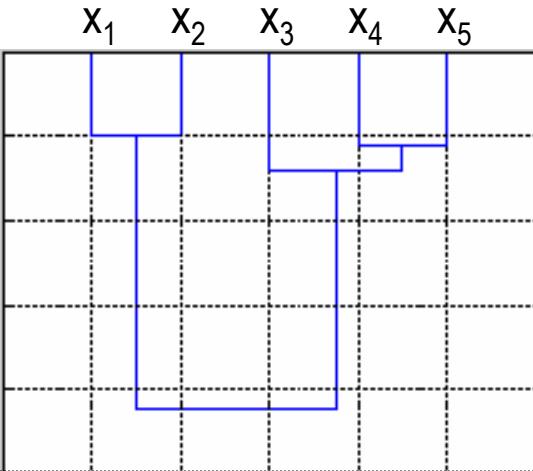
calculate $P_{t+1}(\cdot, q)$ and $P_{t+1}(q, \cdot)$

(3.4) $t = t + 1$

until all examples are in a single **cluster**

Agglomerative clustering

- Additional remarks:



(1) $\mathcal{R}_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$

(2) $t = 0$

(3) repeat

(3.1) Choose the pair of **clusters** $(C_r, C_s) \in \mathcal{R}_t$ ($r \neq s$) such that

$$\wp(C_r, C_s) = \begin{cases} \min_{i,j} \{\wp(C_i, C_j)\} & \wp \text{ is DM} \\ \max_{i,j} \{\wp(C_i, C_j)\} & \wp \text{ is SM} \end{cases}$$

(3.2) $\mathcal{R}_{t+1} = (\mathcal{R}_t - \{C_r, C_s\}) \cup \{C_r \cup C_s\}$

(3.3) $t = t + 1$

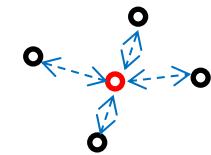
until all samples are in a single **cluster**

- the dendrogram strongly depends on the chosen proximity measure between clusters $\wp(C_i, C_j)$
- the shape of the dendrogram itself indicates whether the clustering at a certain level is natural or not:
 - a great “jump” between levels suggests the existence of a **non-natural merging**
 - it is not necessary to build completely the dendrogram: one can stop at a certain level, before **reaching a non-natural clustering**, or when a **certain number of clusters has been found**
- regarding the adequateness of a merge, every step of the algorithm increases the **total variance of the clustering** E_t , defined by:

$$E_t = \sum_{r \in \mathcal{R}_t} e_r^2 \quad [\text{total variance of clustering } \mathcal{R}_t]$$

$$e_r^2 = \sum_{a \in C_r} \|a - \mu_r\|^2, \quad \mu_r = \frac{1}{n_r} \sum_{a \in C_r} a$$

(Euclidean distance
assumed)



Agglomerative clustering

- **Increase of total variance** from level t to level t+1:
 - Let us consider the merge between clusters C_i and C_j into cluster C_q at level t+1
 - Then, the increment of total variance can be stated as:
- Taking into account that:

$$\Delta E_{t+1}^{ij} = E_{t+1}^{ij} - E_t = \left(\sum_{r \neq q} e_r^2 \right) + e_q^2 - \left(\left(\sum_{r \neq i,j} e_r^2 \right) + e_i^2 + e_j^2 \right) = e_q^2 - e_i^2 - e_j^2$$

$$\begin{aligned} e_r^2 &= \sum_{a \in C_r} \|a - \mu_r\|^2 = \sum_{a \in C_r} (a - \mu_r)^T (a - \mu_r) \\ &= \sum_{a \in C_r} a^T a - 2 \sum_{a \in C_r} \mu_r^T a + \sum_{a \in C_r} \mu_r^T \mu_r \quad \text{--- } \mu_r = \frac{1}{n_r} \sum_{a \in C_r} a \Rightarrow \sum_{a \in C_r} a = n_r \mu_r \\ &= \sum_{a \in C_r} a^T a - 2\mu_r^T (n_r \mu_r) + n_r (\mu_r^T \mu_r) \quad \text{--- } \\ &= \sum_{a \in C_r} a^T a - n_r (\mu_r^T \mu_r) = \sum_{a \in C_r} \|a\|^2 - n_r \|\mu_r\|^2 \end{aligned}$$

Agglomerative clustering

- **Increase of total variance** from level t to level $t + 1$:
 - Since C_q is the merge of C_i and C_j , the increment of total variance is given by:

$$\begin{aligned}\Delta E_{t+1}^{ij} &= \sum_{a \in C_q} \|a\|^2 - n_q \|\mu_q\|^2 - \left(\sum_{a \in C_i} \|a\|^2 - n_i \|\mu_i\|^2 \right) - \left(\sum_{a \in C_j} \|a\|^2 - n_j \|\mu_j\|^2 \right) \\ &= n_i \|\mu_i\|^2 + n_j \|\mu_j\|^2 - n_q \|\mu_q\|^2 = n_i (\mu_i^T \mu_i) + n_j (\mu_j^T \mu_j) - n_q (\mu_q^T \mu_q)\end{aligned}$$

- Taking into account that:

$$n_q \mu_q = \sum_{a \in C_i} a + \sum_{b \in C_j} b = n_i \mu_i + n_j \mu_j \quad \text{and} \quad n_q = n_i + n_j$$

then we obtain:

$$\begin{aligned}\Delta E_{t+1}^{ij} &= n_i (\mu_i^T \mu_i) + n_j (\mu_j^T \mu_j) - n_q \left(\frac{n_i}{n_q} \mu_i + \frac{n_j}{n_q} \mu_j \right)^T \left(\frac{n_i}{n_q} \mu_i + \frac{n_j}{n_q} \mu_j \right) \\ &= \frac{n_i n_j}{n_i + n_j} \mu_i^T \mu_i + \frac{n_i n_j}{n_i + n_j} \mu_j^T \mu_j - 2 \frac{n_i n_j}{n_i + n_j} \mu_i^T \mu_j \\ &= \frac{n_i n_j}{n_i + n_j} (\mu_i - \mu_j)^T (\mu_i - \mu_j) = \boxed{\frac{n_i n_j}{n_i + n_j} \|\mu_i - \mu_j\|^2 = \Delta E_{t+1}^{ij} > 0}\end{aligned}$$

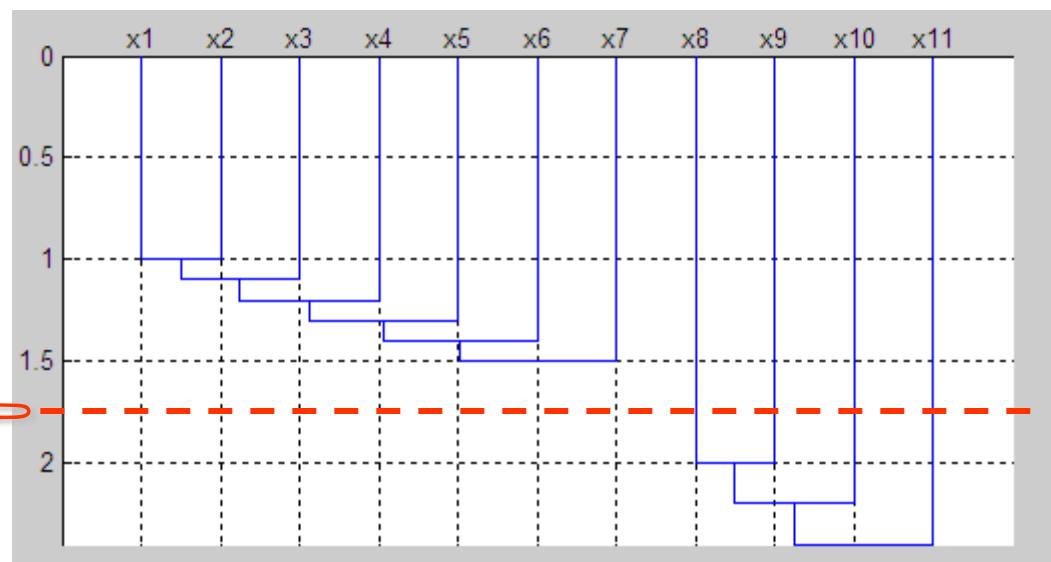
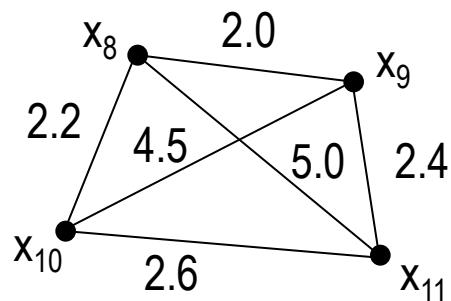
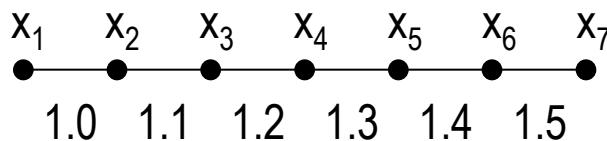
Agglomerative clustering

- Agglomerative algorithms **depending on the proximity function between clusters**
 - Nearest neighbour (or *single-link* / *single-linkage*)
 - (3.1) Choose the pair of **clusters** $(C_r, C_s) \in \mathcal{R}_t$ ($r \neq s$) such that:

$$\wp(C_r, C_s) = \begin{cases} \min_{i,j} \{\wp_{\min}(C_i, C_j) = \min_{a \in C_i, b \in C_j} \wp(a, b)\} & (\wp \text{ is DM}) \\ \max_{i,j} \{\wp_{\max}(C_i, C_j) = \max_{a \in C_i, b \in C_j} \wp(a, b)\} & (\wp \text{ is SM}) \end{cases}$$

– favour elongated clusters (chain effect)

- **Example of application:**



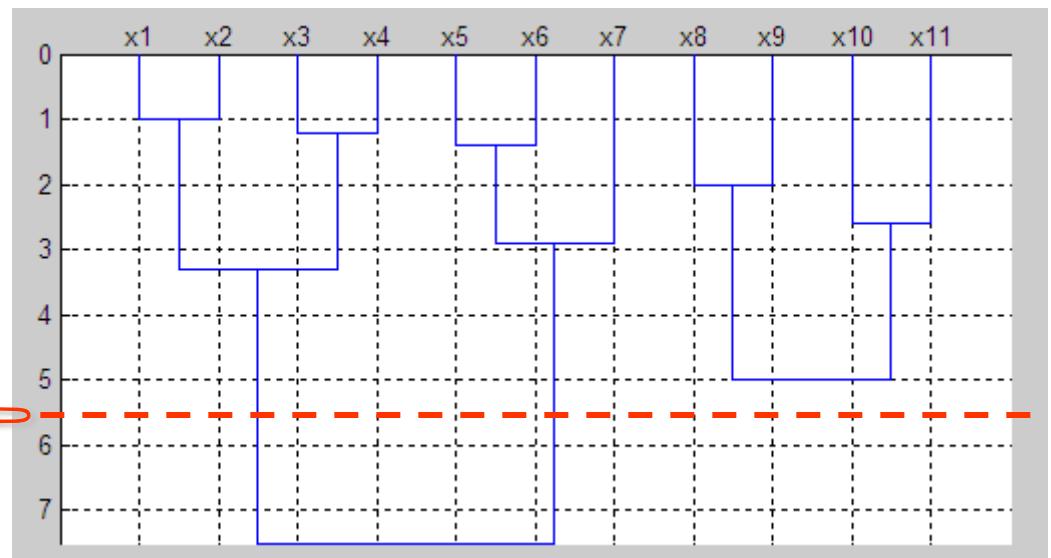
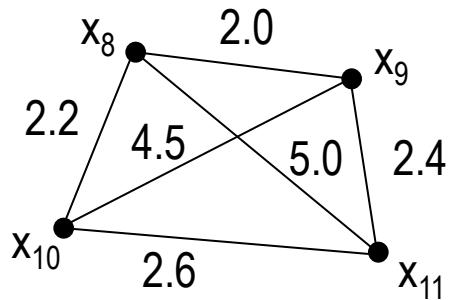
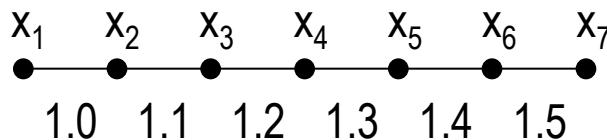
Agglomerative clustering

- Agglomerative algorithms **depending on the proximity function between clusters**
 - **Farthest neighbour** (or *complete-link / complete-linkage*)
 - (3.1) Choose the pair of **clusters** $(C_r, C_s) \in \mathcal{R}_t$ ($r \neq s$) such that:

$$\wp(C_r, C_s) = \begin{cases} \min_{i,j} \{\wp_{\max}(C_i, C_j) = \max_{a \in C_i, b \in C_j} \wp(a, b)\} & (\wp \text{ is DM}) \\ \max_{i,j} \{\wp_{\min}(C_i, C_j) = \min_{a \in C_i, b \in C_j} \wp(a, b)\} & (\wp \text{ is SM}) \end{cases}$$

– **favour compact clusters** (reduced diameter clusters)

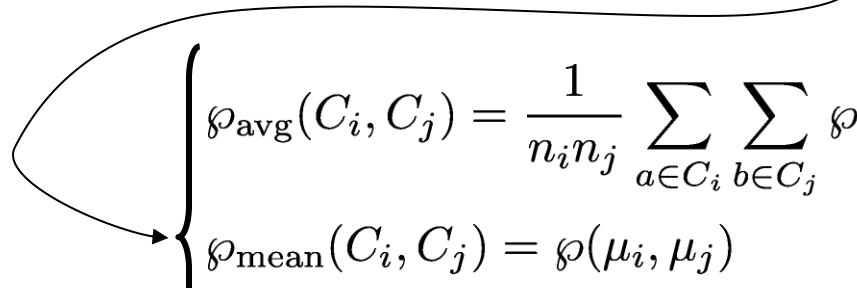
- **Example of application:**



Agglomerative clustering

- Agglomerative algorithms **depending on the proximity function between clusters**
 - NN and FN algorithms are at the **opposite ends of the spectrum** of measures cluster-to-cluster $\wp(C_i, C_j)$
 - Other algorithms, with **intermediate behaviours**, result for other distances:
 - (3.1) Choose the pair of **clusters** $(C_r, C_s) \in \mathcal{R}_t$ ($r \neq s$) such that:

$$\wp(C_r, C_s) = \begin{cases} \min_{i,j} \{\wp(C_i, C_j)\} & (\wp \text{ is DM}) \\ \max_{i,j} \{\wp(C_i, C_j)\} & (\wp \text{ is SM}) \end{cases}$$


$$\left\{ \begin{array}{l} \wp_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{a \in C_i} \sum_{b \in C_j} \wp(a, b) \quad (\text{average linkage alg.}) \\ \wp_{\text{mean}}(C_i, C_j) = \wp(\mu_i, \mu_j) \\ \wp_{\text{ward}}(C_i, C_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}} \wp(\mu_i, \mu_j) \end{array} \right.$$

Agglomerative clustering

- Agglomerative algorithms **depending on the proximity function between clusters**
 - In particular:

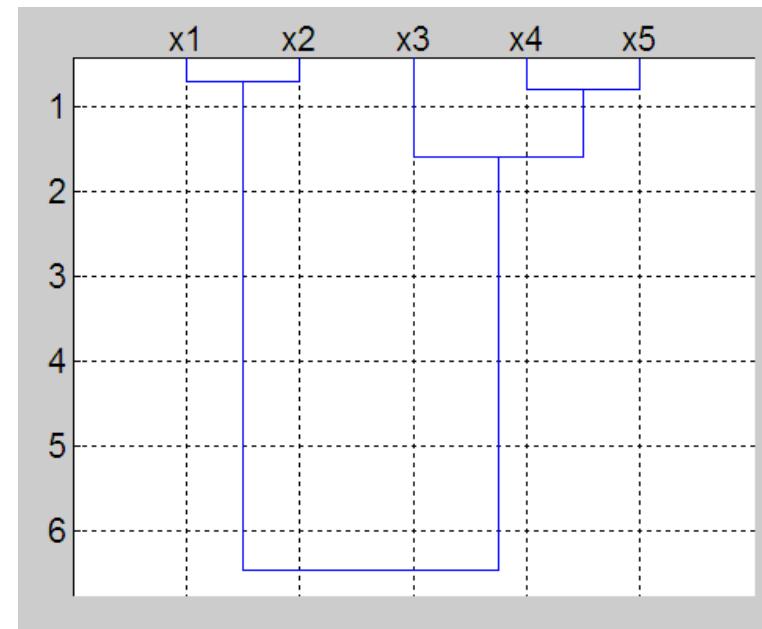
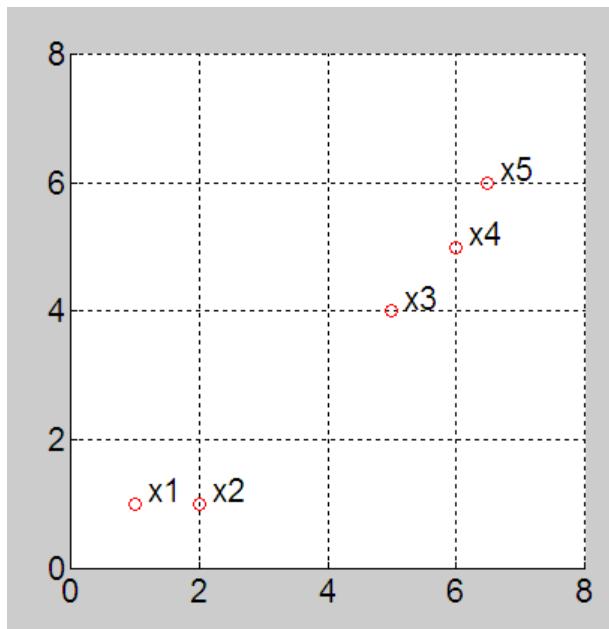
$$\phi_{\text{ward}}(C_i, C_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}} \phi(\mu_i, \mu_j)$$

leads to the **minimum total variance increase** between steps if $\phi(\mu_i, \mu_j) = \|\mu_i - \mu_j\|$:

- The algorithm chooses the pair of clusters that minimize $\sqrt{\frac{n_i n_j}{n_i + n_j}} \|\mu_i - \mu_j\|$ and we have already seen that $\Delta E_{t+1}^{ij} = \frac{n_i n_j}{n_i + n_j} \|\mu_i - \mu_j\|^2$.
- Because of this, this algorithm is termed **algorithm (of agglomerative hierarchical clustering) of minimum variance**.
- At a practical level, this algorithm **favours the fusion of small clusters with large clusters** against fusing large or medium-size *clusters*, because of the involvement of cluster sizes in the proximity measure.

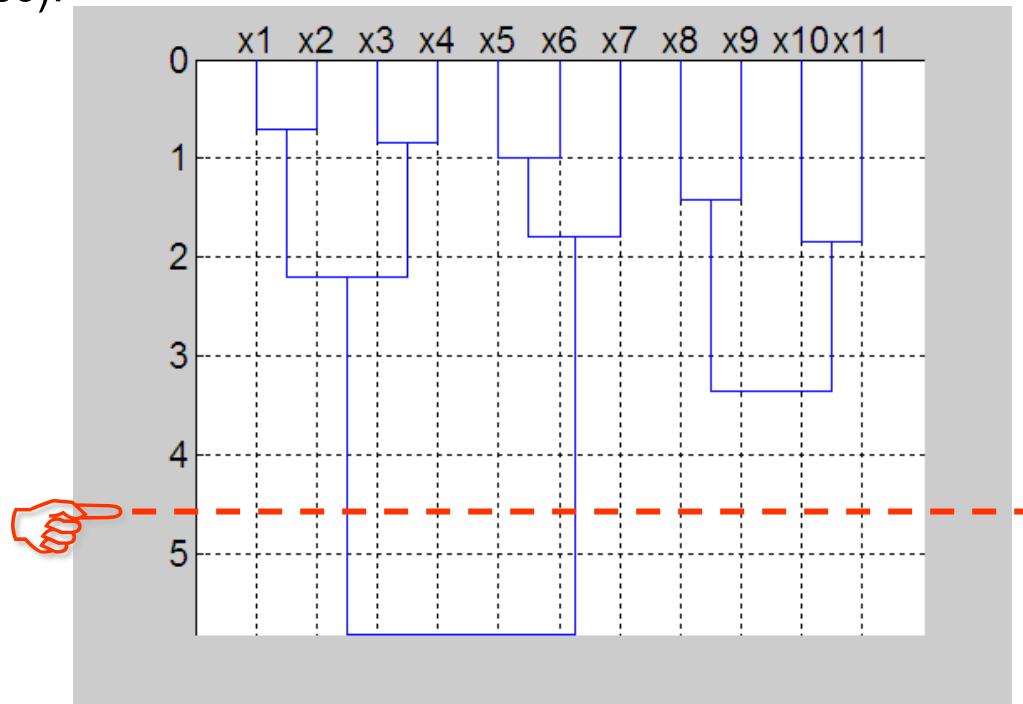
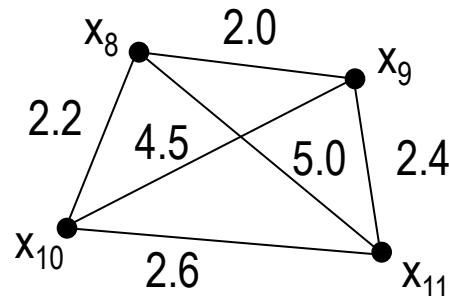
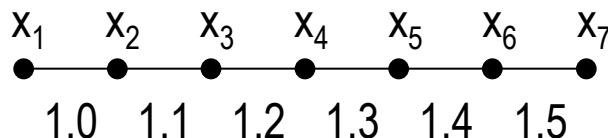
Agglomerative clustering

- Agglomerative algorithms **depending on the proximity function between clusters**
 - **Example** (of Ward distance use):



Agglomerative clustering

- Agglomerative algorithms **depending on the proximity function between clusters**
 - **Example** (of Ward distance use):

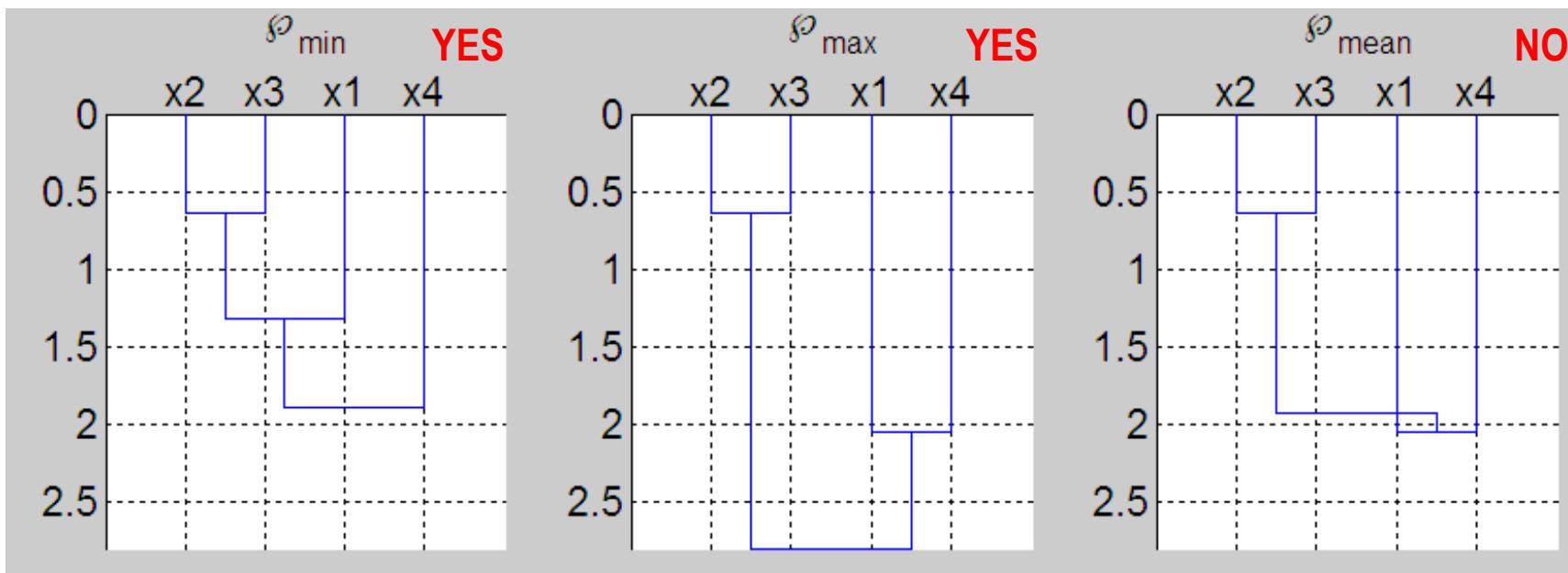


- In clear cases (compact and well separated clusters), all alternatives lead to the same results. **Differences appear for particular cases.**

Agglomerative clustering

- **Monotonicity**

- If the clustering algorithm selects clusters C_i y C_j to build a new cluster C_q so that $d(C_q, C_k) \geq d(C_i, C_j)$, $\forall k \neq i, j, q$, then the resulting dendrogram is said to be **monotonous**
- If monotonicity holds, clusters are created at higher dissimilarity levels than its constituents

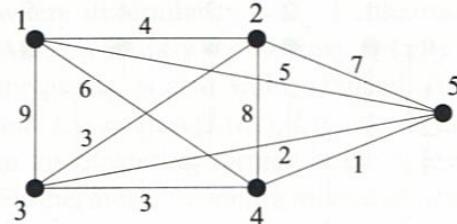


- ρ_{\min} , ρ_{\max} , ρ_{avg} and ρ_{ward} can be proved to always give rise to monotonous dendrograms.
- Monotonicity has been considered to be necessary for a clustering algorithm to be useful.

Agglomerative clustering

- Ties in the proximity matrix $P(X)$

$$P_0 = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 \\ 4 & 0 & 3 & 8 & 7 \\ 9 & 3 & 0 & 3 & 2 \\ 6 & 8 & 3 & 0 & 1 \\ 5 & 7 & 2 & 1 & 0 \end{bmatrix}$$

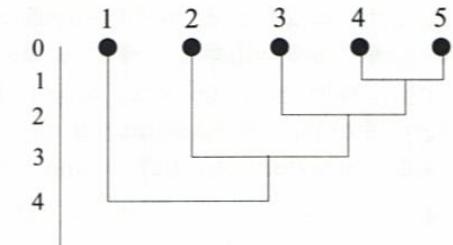


$$P_1^{\phi_{\min}} = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 & 5 \\ 4 & 0 & 3 & 8 & 7 & 7 \\ 9 & 3 & 0 & 3 & 2 & 2 \\ 6 & 8 & 3 & 0 & 1 & 1 \\ 5 & 7 & 2 & 1 & 0 & 0 \\ 5 & 7 & 2 & 1 & 0 & 0 \end{bmatrix}$$

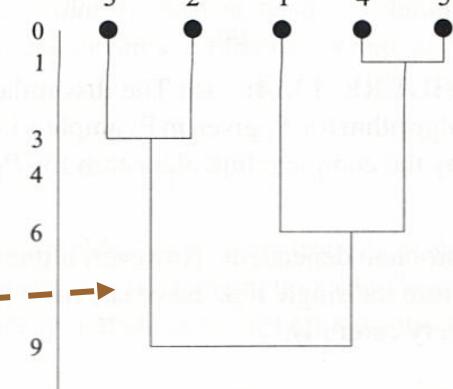
(a)

$$P_1^{\phi_{\max}} = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 & 6 \\ 4 & 0 & 3 & 8 & 7 & 8 \\ 9 & 3 & 0 & 3 & 2 & 3 \\ 6 & 8 & 3 & 0 & 1 & 1 \\ 5 & 7 & 2 & 1 & 0 & 0 \\ 6 & 8 & 3 & 0 & 1 & 0 \end{bmatrix}$$

(c)



(b)



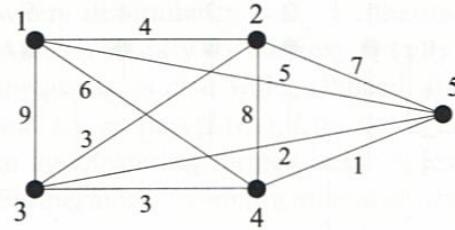
(d)

FIGURE 13.13: (a) The dissimilarity graph ($G(9)$) for the dissimilarity matrix given in Example 13.8. (b) The dissimilarity dendrogram obtained by the single link algorithm. (c) The dissimilarity dendrogram obtained by the complete link algorithm when edge $(3, 4)$ is considered first. (d) The dissimilarity dendrogram obtained by the complete link algorithm when edge $(2, 3)$ is considered first.

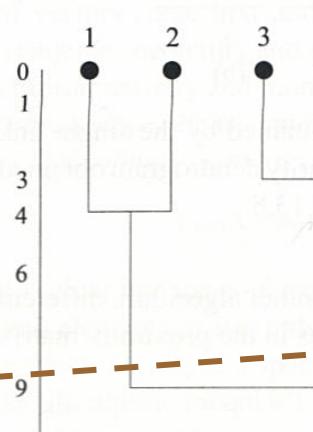
Agglomerative clustering

- Ties in the proximity matrix $P(X)$

$$P_0 = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 \\ 4 & 0 & 3 & 8 & 7 \\ 9 & 3 & 0 & 3 & 2 \\ 6 & 8 & 3 & 0 & 1 \\ 5 & 7 & 2 & 1 & 0 \end{bmatrix}$$



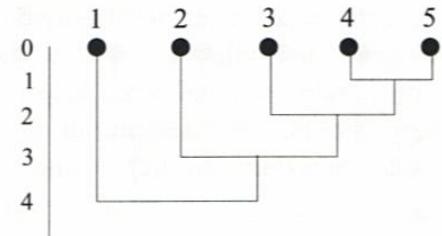
$$P_1^{\phi_{\min}} = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 & 5 \\ 4 & 0 & 3 & 8 & 7 & 7 \\ 9 & 3 & 0 & 3 & 2 & 2 \\ 6 & 8 & 3 & 0 & 1 & 1 \\ 5 & 7 & 2 & 1 & 0 & 0 \\ 5 & 7 & 2 & 1 & 0 & 0 \end{bmatrix}$$



$$P_1^{\phi_{\max}} = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 & 6 \\ 4 & 0 & 3 & 8 & 7 & 8 \\ 9 & 3 & 0 & 3 & 2 & 3 \\ 6 & 8 & 3 & 0 & 1 & 1 \\ 5 & 7 & 2 & 1 & 0 & 0 \\ 6 & 8 & 3 & 1 & 0 & 0 \end{bmatrix}$$



• All agglomerative algorithms are more or less affected by this defect, although the **nearest neighbour algorithm** seems to be the least affected.



- Introduction
- Agglomerative clustering
- Divisive clustering
- Selection of a good clustering

Divisive Clustering

- Generic **algorithm**:

- (1) $\mathcal{R}_0 = \{X\}$

- (2) $t = 0$

- (3) **repeat**

- (3.1) **for** all **clusters** C_q of \mathcal{R}_t :

- Find C_r^q and C_s^q s.t. $C_q = C_r^q \cup C_s^q$, $C_r^q \cap C_s^q = \emptyset$ and

$$\wp(C_r^q, C_s^q) = \begin{cases} \max_{i,j}\{\wp(C_i, C_j)\} & (\wp \text{ is DM}) \\ \min_{i,j}\{\wp(C_i, C_j)\} & (\wp \text{ is SM}) \end{cases}$$

- end**

- (3.2) Find the **cluster** C_{q*} whose splitting is the best

- (3.3) $\mathcal{R}_{t+1} = (\mathcal{R}_t - \{C_{q*}\}) \cup \{C_r^{q*}, C_s^{q*}\}$

- (3.4) $t = t + 1$

- until** all samples are in a different **cluster**

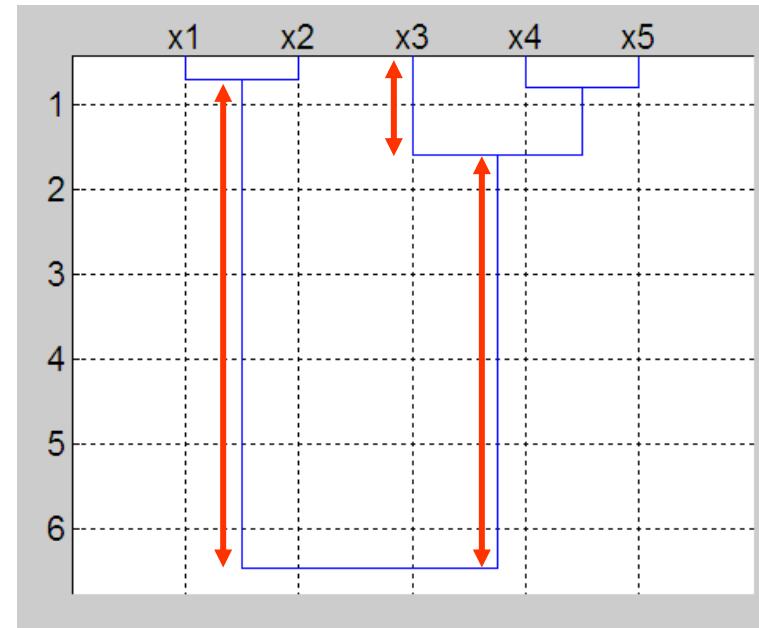
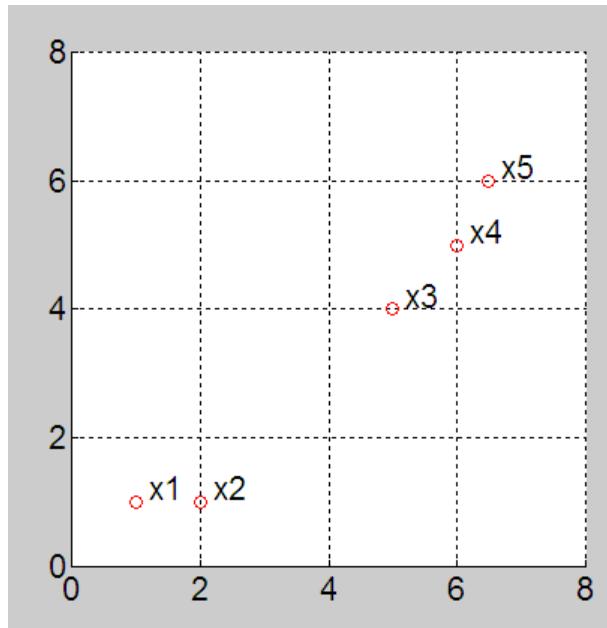
- Very high **cost** at the computational level

- Different alternatives available, which try to reduce the cost of (3.1)

- Introduction
- Agglomerative clustering
- Divisive clustering
- Selection of a good clustering

Selection of a good clustering (in the hierarchy)

- We turn our attention to the determination of a **good clustering within a given hierarchy**. This can help to identify the natural structure of the data.
1. Find in the dendrogram clusters with a **long lifetime**:
 - **lifetime of a cluster** \equiv absolute difference between the proximity level at which a cluster is generated and the level at which it is absorbed

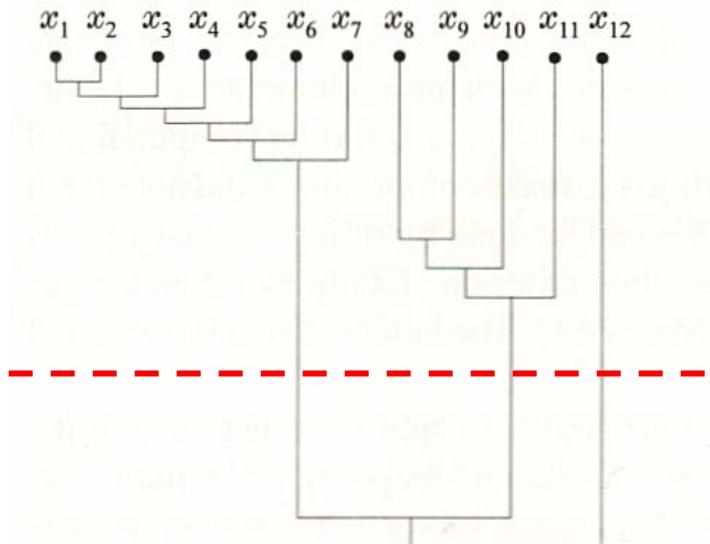


\Rightarrow final clustering should be: $\{x_1, x_2\}, \{x_3, x_4, x_5\}$

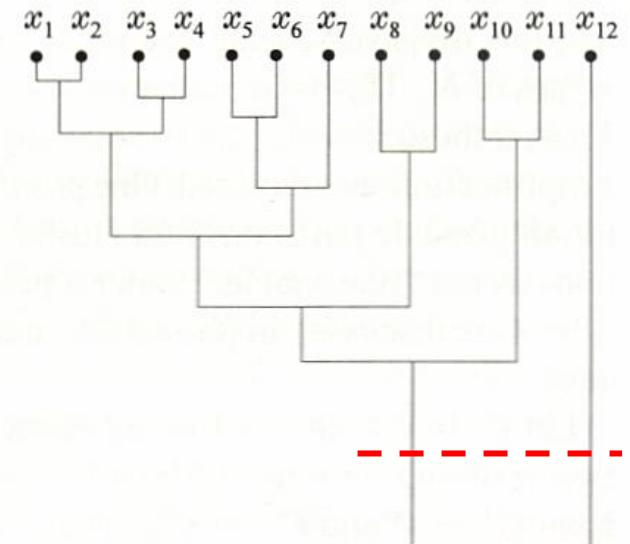
Selection of best clustering

1. Find in the dendrogram clusters with a **long lifetime**:

- **lifetime of a cluster** \equiv absolute difference between the proximity level at which a cluster is generated and the level at which it is absorbed
- another example:



(a)

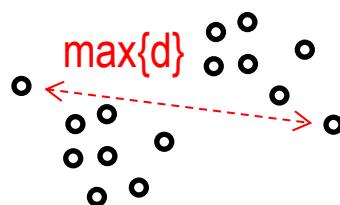


(b)

Selection of best clustering

2. Stop before a **low-quality cluster** results:

- Determine the disimilarity within a cluster by means of an appropriate measure **h**. Several possibilities:



$$h_1(C) = \max\{d(x, y), x, y \in C\} \quad \text{— diameter of } C$$

$$h_2(C) = \text{median}\{d(x, y), x, y \in C\} \quad \text{— robust to outliers}$$

$$h_3(C) = \frac{2}{n_C(n_C - 1)} \sum_{x \neq y \in C} d(x, y)$$

- In this way, every time a cluster is going to be created, one can check its internal dissimilarity and decide not to create it if it is above a threshold τ , and the process is stopped:

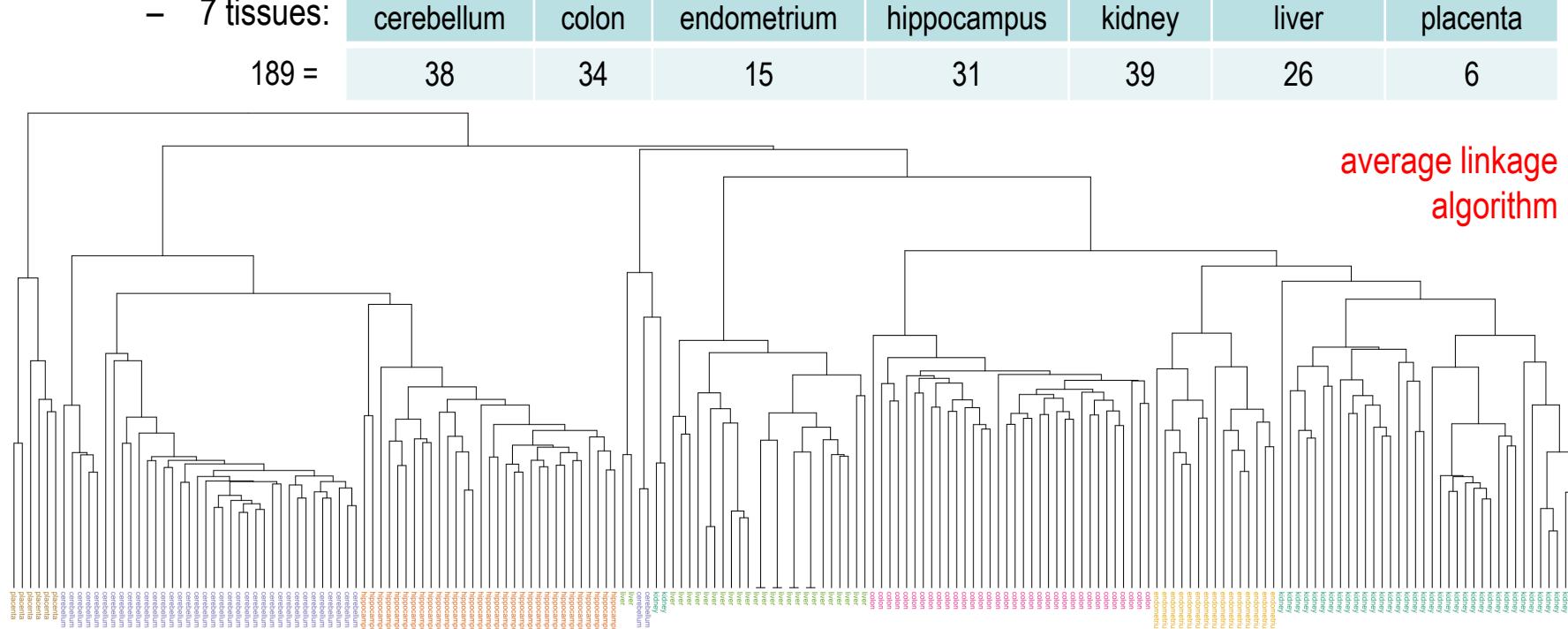
$$C_q = C_i \bigcup C_j \in \mathcal{R}_{t+1} : h(C_q) > \tau \Rightarrow \text{STOP at level } t$$

- It is usually useful to define $\tau = \mu + \lambda\sigma$, where μ is the average distance among elements of X and σ is the standard deviation. (The proximity measure is assumed a dissimilarity.)
 - Typically, it is easier to set λ than τ .

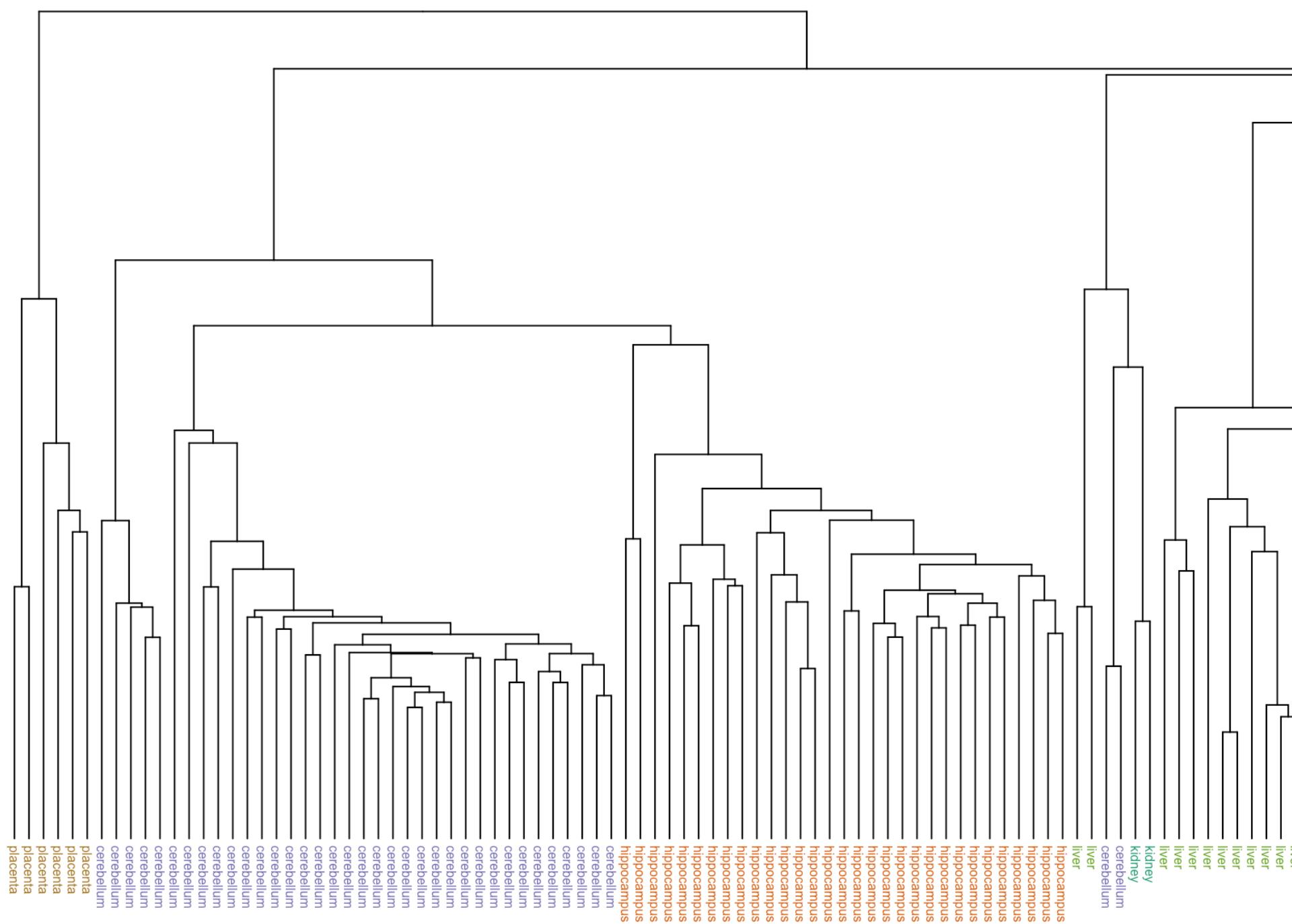
Example

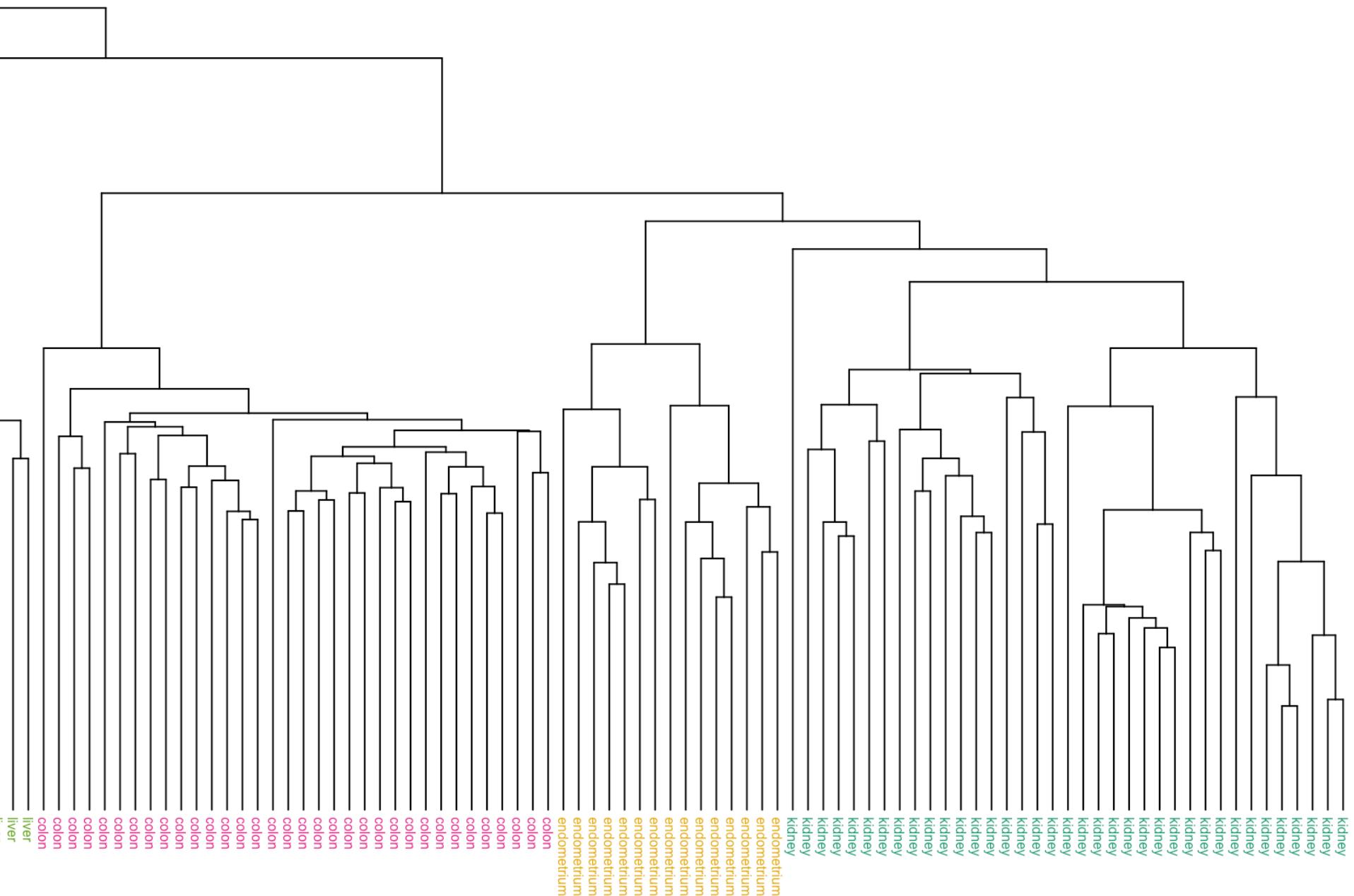
- Example: gene expression profiling of human tissues
 - Each row is a gene expression profile and each column is a different gene. The column names are the gene symbols. The outcome is a character vector representing the tissue.
 - Subset of the original dataset comprising 22.215 samples
 - 189 samples chosen at random, with $L = 500$
 - 7 tissues:

	cerebellum	colon	endometrium	hippocampus	kidney	liver	placenta
189 =	38	34	15	31	39	26	6



- source: <https://github.com/genomicsclass/tissuesGeneExpression>



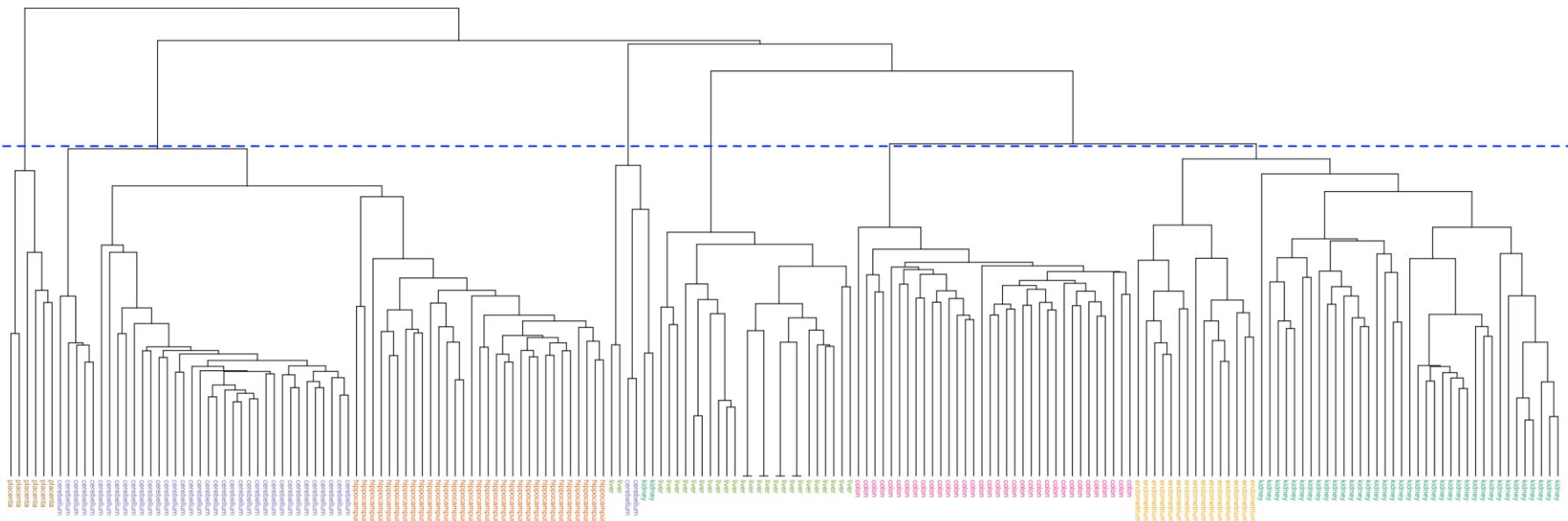


Example

- “Cutting” the dendrogram at height = 125:

contingency table →
(or contingency matrix)

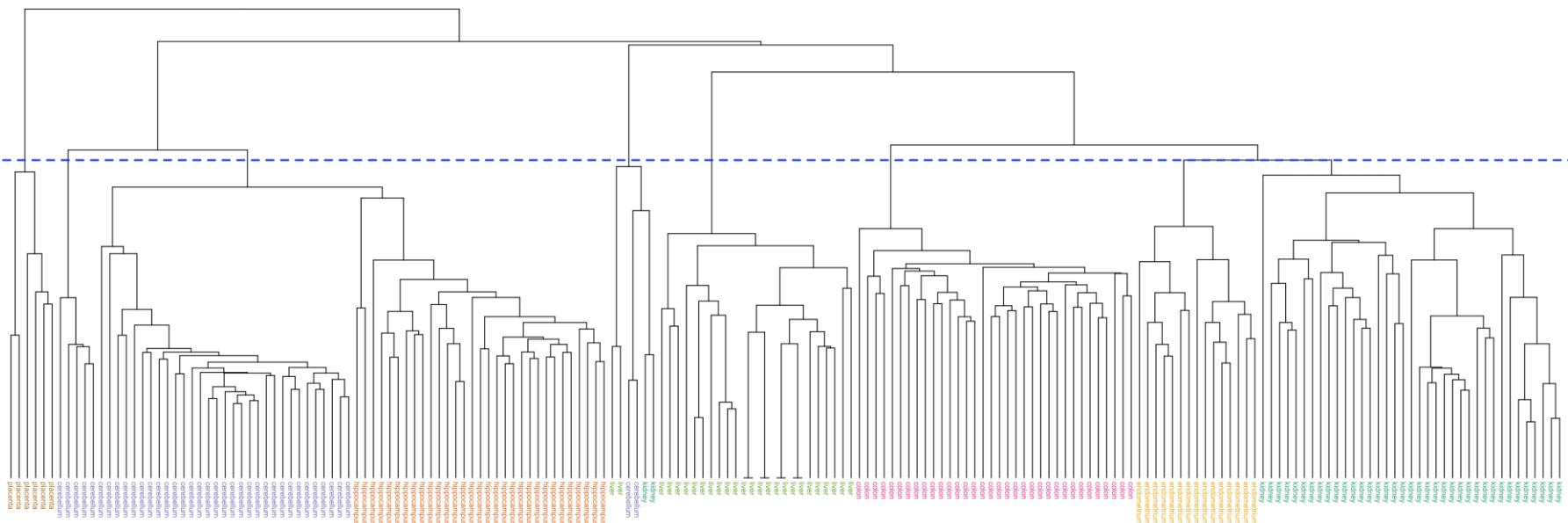
	cluster					
## tissue	1	2	3	4	5	6
## 1.cerebellum	0	36	0	0	2	0
## 2.colon	0	0	34	0	0	0
## 3.endometrium	15	0	0	0	0	0
## 4.hippocampus	0	31	0	0	0	0
## 5.kidney	37	0	0	0	2	0
## 6.liver	0	0	0	24	2	0
## 7.placenta	0	0	0	0	0	6



Example

- “Cutting” the dendrogram so as to have 8 clusters:

		cluster							
	tissue	1	2	3	4	5	6	7	8
##	1.cerebellum	0	31	0	0	2	0	5	0
##	2.colon	0	0	34	0	0	0	0	0
##	3.endometrium	0	0	0	0	0	15	0	0
##	4.hippocampus	0	31	0	0	0	0	0	0
##	5.kidney	37	0	0	0	2	0	0	0
##	6.liver	0	0	0	24	2	0	0	0
##	7.placenta	0	0	0	0	0	0	0	6



Section 2

Unsupervised Learning: Hierarchical Clustering



Universitat
de les Illes Balears

Departament
de Ciències Matemàtiques
i Informàtica

11752 Aprendizaje Automático
11752 Machine Learning
Máster Universitario
en Sistemas Inteligentes

Alberto ORTIZ RODRÍGUEZ