Clustering algorithms

Konstantinos Koutroumbas

Unit 7

– Discussion on clustering CFO algorithms

– Hierarchical clustering (agglomerative case)

koutroum@noa.gr

CFO clustering algorithms: A unified view

Data

$$X = \{\boldsymbol{x}_j \in R^l, j = 1, \dots, N\}$$

Recall the general CFO framework

Basic parameters - notation

✓ $\Theta = \{\boldsymbol{\theta}_j, j = 1, \dots, m\}$ ($\boldsymbol{\theta}_j$ is the representative of cluster C_j).

- **Proximity** between $\boldsymbol{x}_i$ and C_j: $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$

CFO clustering algorithms: A unified view

Basic parameters – notation (cont.)

In the probabilistic case u_{ij} stands for $P(j|\boldsymbol{x}_i)$

Recall the general CFO framework

✓ $$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{Nm} \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_N \end{bmatrix}$$

- $u_{ij} \in [0,1]$ quantifies the "relation" between $\boldsymbol{x}_i$ and C_j.
- "Large" ("small") u_{ij} values indicate close (loose) relation between $\boldsymbol{x}_i$ and C_j.

$\Rightarrow$ u_{ij} varies **inversely proportional** wrt $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

- $\boldsymbol{u}_i$: vector containing the u_{ij}'s of $\boldsymbol{x}_i$ with all clusters.

(*) Unless otherwise stated, the case where **cluster representatives** are used is considered.

CFO clustering algorithms: A unified view

Aim:

✓ To place the representatives into dense in data regions (physical clusters).

Recall the general CFO framework

How this is achieved:

✓ Via the minimization of the following type of cost function (wrt Θ, U)

$$J(\Theta, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^{q} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \ (q \geq 1)$$

s.t. some **constraints** on U, $C(U)$.

For the probabilistic case $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ is embedded in the log-likelihood of suitably defined exponential distributions

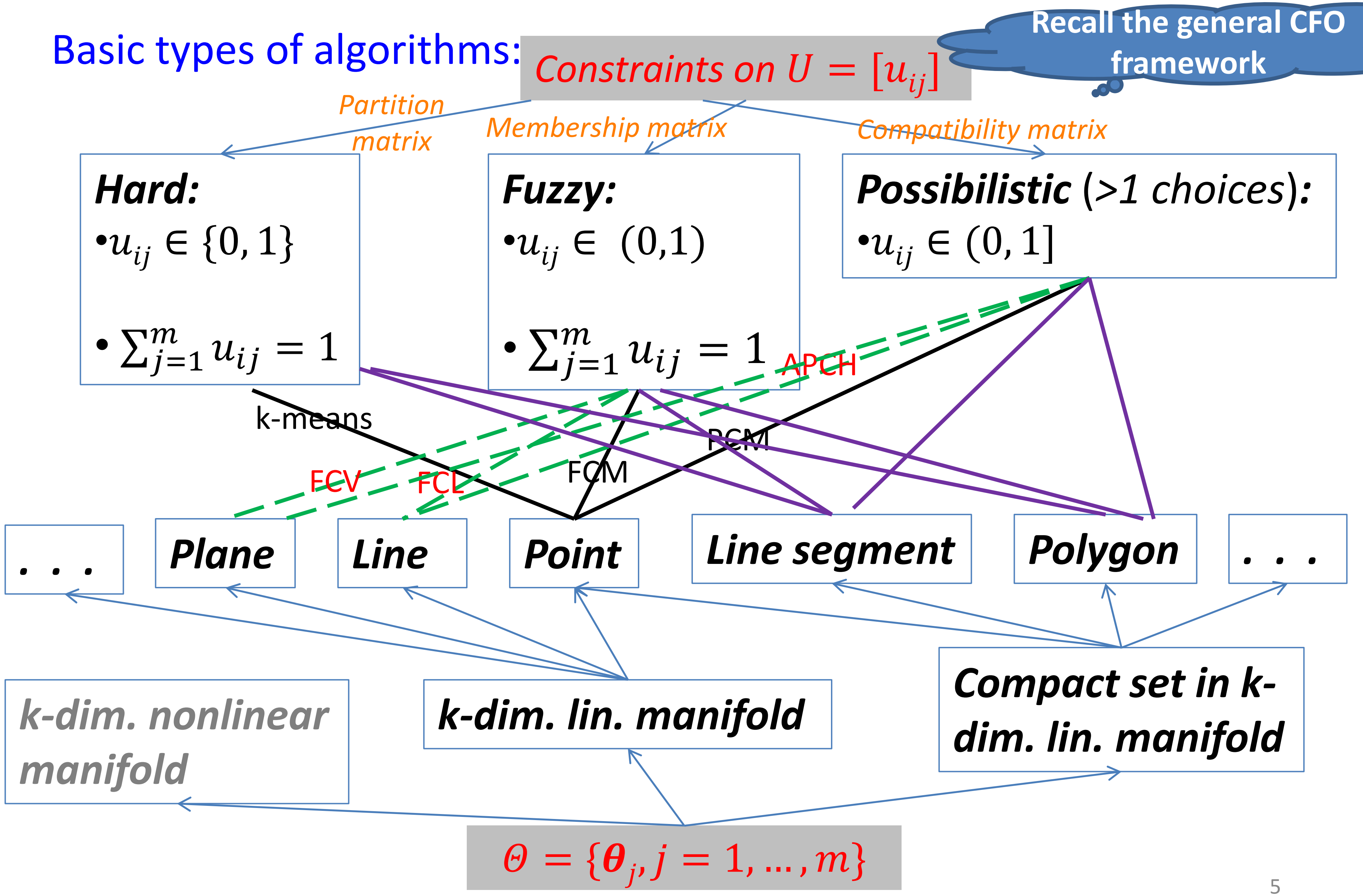
Intuition:

✓ For **fixed** $\boldsymbol{\theta}_j$'s, $J(\Theta, U)$ is a weighted sum of **fixed** distances $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

$\Rightarrow$ **Minimization** of $J(\Theta, U)$ wrt u_{ij} instructs for large weights (u_{ij}) for small distances $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

✓ For **fixed** u_{ij}'s, **minimization** of $J(\Theta, U)$ wrt $\boldsymbol{\theta}_j$'s leads $\boldsymbol{\theta}_j$'s closer to their most relative data points.

CFO clustering algorithms: A unified view



CFO clustering algorithms: A unified view

"Array of CFO algorithms"

algorithm

Recall the general CFO framework

$C(U)$

θ_j

	Hard Constr.	Fuzzy Constr.	Possi Constr.	
Point				
Line				
Hyperplane				
Hyperellipsoid				
. . .				

CFO scheme

There are **several** unexplored areas (groups of algorithms) in this array.

CFO clustering algorithms: A unified view

General cost function opt. (**CFO**) scheme:

Recall the general CFO framework

- ✓ *Initialize* $\Theta = \Theta(0)$
- ✓ $t = 0$
- ✓ **Repeat**
  - $U(t) = argmin_U \; J(\Theta(t), U)$, s.t. $C(U(t))$
  - $t = t + 1$
  - $\Theta(t) = argmin_\Theta \; J(\Theta, U(t-1))$
- ✓ **Until convergence**

CFO clustering algorithms: A unified view

Recall the general CFO framework

"Array of CFO algorithms"

$C(U)$

$\boldsymbol{\theta}_j$

	Hard Constr.	Fuzzy Constr.	Possib. Constr.	. . .
Point				
Line				
Hyperplane				
Hyperellipsoid				
. . .				

Hard CFO scheme

Fuzzy CFO scheme

Possib. CFO scheme

CFO scheme

CFO clustering algorithms: A unified view

"Array of CFO algorithms"

Recall the general CFO framework

$C(U)$

$\boldsymbol{\theta}_j$

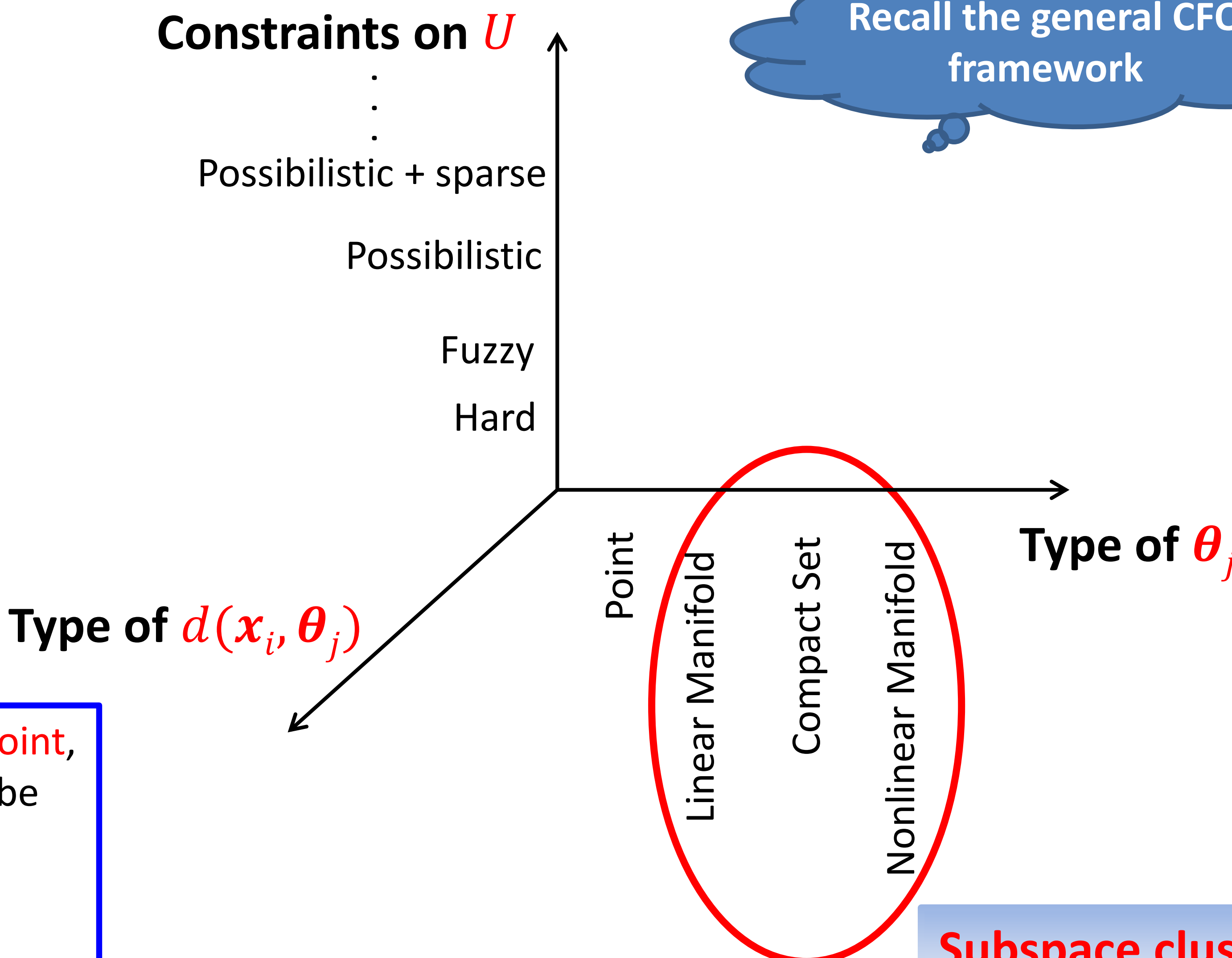
	Hard Constr.	Fuzzy Constr.	Possib. Constr.	. . .
Point	c-means scheme			
Line	c-lines scheme			
Hyperplane	c-hyperplanes scheme			
Hyperellipsoid	c-hyperellipsoids scheme			
. . .				

CFO scheme

CFO clustering algorithms: A unified view

CFO clustering algorithms: A loose presentation

Recall the general CFO framework



E.g.:If $\boldsymbol{\theta}_j$ is a point, $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ may be
- Sq. Euclidean
- l_p norm
- Mahalanobis

Subspace clustering

CFO clustering algorithms: Final remarks (1)

Relating hard, fuzzy and probabilistic clustering

(point representatives, squared Euclidean distance)

A. Generalized Hard Algorithmic Scheme (GHAS) – k-means algorithm

$$minimize_{U,\Theta} J(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij} \,||\boldsymbol{x}_i - \boldsymbol{\theta}_j||^2$$

subject to **(a)** $u_{ij} \in \{0,1\}$, $i = 1, \ldots, N, j = 1, \ldots, m$, and **(b)** $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \ldots, N$.

The Isodata or k-Means or c-Means algorithm

- Choose arbitrary initial estimates $\boldsymbol{\theta}_j(0)$ for the $\boldsymbol{\theta}_j$’ s, j=1, …,m.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of the partition*
    - o For $j = 1$ to m

$$u_{ij}(t) = \begin{cases} 1, & if \; ||\boldsymbol{x_i} - \boldsymbol{\theta}_j(t)||^2 = min_{q=1,\ldots,m} ||\boldsymbol{x_i} - \boldsymbol{\theta}_q(t)||^2 \\ 0, & otherwise \end{cases}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j(t) = \frac{\sum_{i=1}^{N} u_{ij}(t-1)\boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}(t-1)}, j = 1, \ldots, m$$

  - End {For-j}
- **Until** no change in $\boldsymbol{\theta}_j$’ s occurs between two successive iterations

CFO clustering algorithms: Final remarks (1)

Relating hard, fuzzy and probabilistic clustering

(point representatives, squared Euclidean distance)

B. Generalized Fuzzy Algorithmic Scheme (GFAS) – Fuzzy c-means algorithm

$$minimize_{U,\Theta} J(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^q \left|\left|\boldsymbol{x}_i - \boldsymbol{\theta}_j\right|\right|^2$$

subject to **(a)** $u_{ij} \in (0,1)$, $i = 1,\dots,N, j = 1,\dots,m$, and **(b)** $\sum_{j=1}^{m} u_{ij} = 1, i = 1,\dots,N$.

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, j=1,…,m.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of $u_{ij}'s$*
    - o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$$

    - o End {For-j}
  - – End {For-i}
  - –$t = t + 1$
  - – For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j(t) = \frac{\sum_{i=1}^{N} {u_{ij}}^q(t-1)\boldsymbol{x}_i}{\sum_{i=1}^{N} {u_{ij}}^q(t-1)}, j = 1,\dots,m$$

  - – End {For-j}
- **Until** a termination criterion is met.

CFO clustering algorithms: Final remarks (1)

Relating hard, fuzzy and probabilistic clustering

(point representatives, squared Euclidean distance)

C. Generalized Probabilistic Algorithmic Scheme (GPrAS) – the normal pdfs case

$$minimize_{\Theta,P} J(\Theta,P) = -\sum_{i=1}^{N}\sum_{j=1}^{m} P(j|\boldsymbol{x}_i)\ln\left(p\left(\boldsymbol{x}_i\middle|j;\boldsymbol{\theta}_j\right)P_j\right)$$

It is **(a)** $P(j|\boldsymbol{x}_i) \in (0,1),\ i = 1,\dots,N, j = 1,\dots,m$, and **(b)** $\sum_{j=1}^{m} P(j|\boldsymbol{x}_i) = 1, i = 1,\dots,N$.

- **Choose** $\boldsymbol{\mu}_j(0)$, $\Sigma_j(0)$, $P_j(0)$ as initial estimates for $\boldsymbol{\mu}_j, \Sigma_j, P_j$, resp., $j = 1,\dots,m$
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Expectation step*
    - o For $j = 1$ to m

$$P(j|\boldsymbol{x}_i;\Theta^{(t)},P^{(t)}) = \frac{p(x_i|j;\theta_j{}^{(t)})P_j{}^{(t)}}{\sum_{q=1}^{m} p(x_i|q;\theta_q{}^{(t)})P_q{}^{(t)}} \equiv \gamma_{ji}{}^{(t)}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating – Maximization step*
    - o Set

$$\boldsymbol{\mu}_j{}^{(t)} = \frac{\sum_{i=1}^{N}\gamma_{ji}{}^{(t-1)}\boldsymbol{x_i}}{\sum_{i=1}^{N}\gamma_{ji}{}^{(t-1)}}, \qquad \Sigma_j{}^{(t)} = \frac{\sum_{i=1}^{N}\gamma_{ji}{}^{(t-1)}(\boldsymbol{x_i}-\boldsymbol{\mu_j})(\boldsymbol{x_i}-\boldsymbol{\mu_j})^{\boldsymbol{T}}}{\sum_{i=1}^{N}\gamma_{ji}{}^{(t-1)}}\ j = 1,\dots,m$$

$$P_j{}^{(t)} = \frac{1}{N}\sum_{i=1}^{N}\gamma_{ji}{}^{(t-1)}, j = 1,\dots,m$$

  - End {For-j}
- **Until** a termination criterion is met.

CFO clustering algorithms: Final remarks (1)

Relating hard, fuzzy and probabilistic clustering

(point representatives, squared Euclidean distance)

Consider the **GPrAS cost function**

$$J(\Theta, P) = -\sum_{i=1}^{N}\sum_{j=1}^{m} P(j|\boldsymbol{x}_i)\ln\big(p(\boldsymbol{x}_i|j;\boldsymbol{\theta}_j)P_j\big)$$

with

$\boldsymbol{\theta}_j = \{\boldsymbol{\mu}_j, \Sigma_j\}$

$$p(\boldsymbol{x}_i|j;\boldsymbol{\theta}_j) = \frac{1}{(2\pi)^{\frac{l}{2}}|\Sigma_j|^{\frac{1}{2}}} exp\left(-\frac{(\boldsymbol{x}_i-\boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_j)}{2}\right)$$

It is $J(\Theta, P) = -\sum_{i=1}^{N}\sum_{j=1}^{m} P(j|\boldsymbol{x}_i)\ln\left(\frac{1}{(2\pi)^{\frac{l}{2}}|\Sigma_j|^{\frac{1}{2}}} exp\left(-\frac{(\boldsymbol{x}_i-\boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_j)}{2}\right)P_j\right) =$

Term **A** $$-\sum_{i=1}^{N}\sum_{j=1}^{m} P(j|\boldsymbol{x}_i)\ln\left(\frac{1}{(2\pi)^{\frac{l}{2}}|\Sigma_j|^{\frac{1}{2}}}\right)$$

Term **B** $$+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{m} P(j|\boldsymbol{x}_i)(\boldsymbol{x}_i-\boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_j)$$

Term **C** $$-\sum_{i=1}^{N}\sum_{j=1}^{m} P(j|\boldsymbol{x}_i)\ln P_j$$

CFO clustering algorithms: Final remarks (1)

Relating hard, fuzzy and probabilistic clustering

(point representatives, squared Euclidean distance)

<u>Assumption 1:</u> $\Sigma_j = \Sigma = constant,\ j = 1, \dots, m$. Then

$$Term\ \boldsymbol{A} = -\sum_{i=1}^{N} \sum_{j=1}^{m} P(j|\boldsymbol{x}_i) \ln\left(\frac{1}{(2\pi)^{\frac{l}{2}} |\Sigma|^{\frac{1}{2}}}\right)$$

$$= -\ln\left(\frac{1}{(2\pi)^{\frac{l}{2}} |\Sigma|^{\frac{1}{2}}}\right) \sum_{i=1}^{N} \sum_{j=1}^{m} P(j|\boldsymbol{x}_i) = -\ln\left(\frac{1}{(2\pi)^{\frac{l}{2}} |\Sigma|^{\frac{1}{2}}}\right) \sum_{i=1}^{N} 1$$

$$= -N \ln\left(\frac{1}{(2\pi)^{\frac{l}{2}} |\Sigma|^{\frac{1}{2}}}\right) = constant$$

<u>Assumption 2:</u> $P_j = \frac{1}{m},\ j = 1, \dots, m$. Then

$Term\ \boldsymbol{C}$

$$= -\sum_{i=1}^{N} \sum_{j=1}^{m} P(j|\boldsymbol{x}_i) \ln \frac{1}{m} = -\ln \frac{1}{m} \sum_{i=1}^{N} \sum_{j=1}^{m} P(j|\boldsymbol{x}_i) = -N \ln \frac{1}{m} = constant$$

CFO clustering algorithms: Final remarks (1)

Relating hard, fuzzy and probabilistic clustering

(point representatives, squared Euclidean distance)

Based on the previous two results, it follows that

$$minimize\left(-\sum_{i=1}^{N}\sum_{j=1}^{m}P(j|\boldsymbol{x}_i)\ln\left(p\left(\boldsymbol{x}_i\middle|j;\boldsymbol{\theta}_j\right)P_j\right)\right)$$

$$\Updownarrow$$

$$minimize\left(\sum_{i=1}^{N}\sum_{j=1}^{m}P(j|\boldsymbol{x}_i)\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^T\Sigma^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)\right)$$

$\Sigma_j = \Sigma$

<u>Assumption 3(a):</u> Approximate $P(j|\boldsymbol{x}_i)$ as

$$P(j|\boldsymbol{x}_i) = \begin{cases}1, & P(j|\boldsymbol{x}_i) = max_{s=1,\ldots,m}P(s|\boldsymbol{x}_i)\\ 0, & otherwise\end{cases} \quad (\equiv u_{ij})$$

In this case, $GPrAS \Leftrightarrow k-means$ (for $\Sigma = \sigma^2 I$)

WARNING: Valid ONLY from a mathematical formulation point of view. NOT from a conceptual point of view.

<u>Assumption 3(b):</u> Approximate $P(j|\boldsymbol{x}_i)$ as

$$P(j|\boldsymbol{x}_i) = \frac{1}{\sum_{k=1}^{m}\left(\frac{d(\boldsymbol{x}_i,\boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i,\boldsymbol{\theta}_k(t))}\right)^{\frac{1}{q-1}}}$$

In this case, $GPrAS \Leftrightarrow fuzzy\ c-means$ (for $\Sigma = \sigma^2 I$)

CFO clustering algorithms: Final remarks (1)

Relating hard, fuzzy and probabilistic clustering
(point representatives, squared Euclidean distance)

Remarks:
The **hard**, **fuzzy** and **probabilistic CFO** clustering algorithms (with point representatives and squared Euclidean distance) :

- are **partition algorithms**.
- they **share** the "**sum-to-one**" constraint.
- they <u>can be related to each other</u> (through the "sum-to-one" constraint).

The **possibilistic** CFO clustering algorithms (point representatives and squared Euclidean distance) :

- are **mode seeking algorithms**
- **<u>no</u> "sum-to-one"** constraint is associated with them
- they <u>**can not** be related to the hard, fuzzy and probabilistic CFO clustering algorithms</u> (due to the absence of the sum-to-one constraint).

CFO clustering algorithms: Final remarks (2)

The role of q in the fuzzy clustering

Consider the minimization problem for fuzzy clustering

$$minimize_{U,\Theta} J(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^{q}\, d_{ij}$$

$d_{ij} = d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$

subject to **(a)** $u_{ij} \in (0,1),\ i = 1, \ldots, N, j = 1, \ldots, m$, and **(b)** $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \ldots, N$.

Expanding $J(U,\Theta)$, we have

$$J(U,\Theta) = \begin{array}{cccc} {u_{11}}^{q} d_{11} + & {u_{12}}^{q} d_{12} + & \ldots & {u_{1m}}^{q} d_{1m} \\ {u_{21}}^{q} d_{21} + & {u_{22}}^{q} d_{22} + & \ldots & {u_{2m}}^{q} d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ {u_{N1}}^{q} d_{N1} + & {u_{N2}}^{q} d_{N2} + & \cdots & {u_{Nm}}^{q} d_{Nm} \end{array}$$

<u>Assumption:</u> d_{ij}'s are fixed.

Then, due to the sum-to-one constraint, $J(U,\Theta)$ is **minimized** if each of the summation in the rows of the above expansion is minimized.

Let s_i: $d_{is_i} = min_{j=1,\ldots,m} d_{ij}, i = 1, \ldots, N$

Then,

$${u_{i1}}^{q} d_{i1} + \ldots + {u_{im}}^{q} d_{im} \geq \left(\sum_{j=1}^{m} {u_{ij}}^{q}\right) d_{is_i}$$

CFO clustering algorithms: Final remarks (2)

The role of q in the fuzzy clustering

$$A_i = {u_{i1}}^q d_{i1} + \ldots + {u_{im}}^q d_{im} \geq \left(\sum_{j=1}^{m} {u_{ij}}^q\right) d_{is_i}$$

For $q = 1$, it is $\sum_{j=1}^{m} u_{ij} = 1$. Thus

$$A_i = u_{i1} d_{i1} + \ldots + u_{im} d_{im} \geq d_{is_i}$$

Clearly, the **equality holds** for $u_{is_i} = 1$ and $u_{ij} = 0$, for $j = 1, \ldots, m, j \neq s_i$

In other words the minimum possible value of A_i is achieved for the hard cluster solution. Thus, **no** fuzzy clustering (where more than one u_{ij}'s are positive) **minimizes** the A_i.

For $q > 1$, in the hard clustering case, the minimum possible value of A_i is still d_{is_i}.

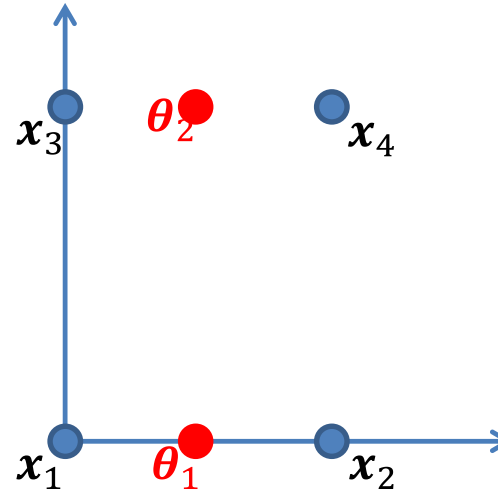
For $q > 1$, in the fuzzy clustering case, it is $\sum_{j=1}^{m} {u_{ij}}^q < 1$. Thus

$$d_{is_i} > \left(\sum_{j=1}^{m} {u_{ij}}^q\right) d_{is_i}$$

Thus, in this case, there are choices for u_{ij}'s with more than one of them being positive (fuzzy case) that achieve lower value for A_i than the best hard clustering. The larger the value of q, the more fuzzy clusterings **achieve** for A_i value $< d_{is_i}$.

CFO clustering algorithms: Final remarks (2)

The role of q in the fuzzy clustering



Example: $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}$

$\boldsymbol{x}_1 = [0,0]^T$, $\boldsymbol{x}_2 = [2,0]^T$, $\boldsymbol{x}_3 = [0,3]^T$, $\boldsymbol{x}_4 = [2,3]^T$

$\boldsymbol{\theta}_1 = [1,0]^T$, $\boldsymbol{\theta}_2 = [1,3]^T$ **(fixed)**

$\boldsymbol{q} = \mathbf{1}$ (hard case): **Best solution** $U_{hard} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$, $J_{hard} = \mathbf{4}$

$\boldsymbol{q} = \mathbf{2}$ (fuzzy case): **Focus** on $\boldsymbol{x}_1$:

Question: Is it possible to have

$$u_{11}{}^2 \cdot d(\boldsymbol{x}_1, \boldsymbol{\theta}_1) + u_{12}{}^2 \cdot d(\boldsymbol{x}_1, \boldsymbol{\theta}_2) < d(\boldsymbol{x}_1, \boldsymbol{\theta}_1) \Rightarrow u_{11}{}^2 \cdot 1 + u_{12}{}^2 \cdot \sqrt{10} < 1? \quad \textbf{(A)}$$

Since $u_{12} = 1 - u_{11}$, **(A)** becomes

$$u_{11}{}^2 \cdot 1 + (1 - u_{11})^2 \cdot \sqrt{10} < 1 \Leftrightarrow$$

$$\left(\sqrt{10} + 1\right)u_{11}{}^2 - 2\sqrt{10}u_{11} + \sqrt{10} - 1 < 0 \Leftrightarrow$$

$$u_{11} \in (0.52,1) \Rightarrow u_{12} \in (0,0.48)$$

$d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$	$\boldsymbol{\theta}_1 = (1,0)$	$\boldsymbol{\theta}_2 = (1,3)$
$\boldsymbol{x}_1 = (0,0)$	$d_{11} = 1$	$d_{12} = \sqrt{10}$
$\boldsymbol{x}_2 = (2,0)$	$d_{21} = 1$	$d_{22} = \sqrt{10}$
$\boldsymbol{x}_3 = (0,3)$	$d_{31} = \sqrt{10}$	$d_{32} = 1$
$\boldsymbol{x}_4 = (2,3)$	$d_{41} = \sqrt{10}$	$d_{42} = 1$

For example, if $u_{11} = 0.7$ ($u_{12} = 0.3$), it is

$$u_{11}{}^2 \cdot 1 + u_{12}{}^2 \cdot \sqrt{10} = 0.7^2 \cdot 1 + 0.3^2 \cdot \sqrt{10} = 0.77 < 1$$

CFO clustering algorithms: Final remarks (3)

The role of q in the possibilistic clustering

Consider the minimization problem for possibilistic clustering

$$minimize_{U,\Theta} J(\boldsymbol{u}_j, \boldsymbol{\theta}_j) = \sum_{i=1}^{N} {u_{ij}}^q d_{ij} + \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q$$

subject to $u_{ij} \in (0,1)$, $i = 1, \dots, N, j = 1, \dots, m$.

For $q = 1$, $J(\boldsymbol{u}_j, \boldsymbol{\theta}_j)$ is written as

$$J(\boldsymbol{u}_j, \boldsymbol{\theta}_j) = \sum_{i=1}^{N} [u_{ij}(d_{ij} - \eta_j) + \eta_j]$$

Thus, minimizing $J(\boldsymbol{u}_j, \boldsymbol{\theta}_j)$ is equivalent to minimizing

$$\sum_{i=1}^{N} u_{ij}(d_{ij} - \eta_j)$$

For fixed $\boldsymbol{\theta}_j$($\Rightarrow$ fixed $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \equiv d_{ij}$), the latter achieves it **minimum** (negative) value by selecting $u_{ij} = 1$, for $d_{ij} < \eta_j$
and $u_{ij} = 0$, for $d_{ij} > \eta_j$.

However, in the above situation, all points having distance less than η_j from $\boldsymbol{\theta}_j$ share the same weight in the determination of $\boldsymbol{\theta}_j$ ($u_{ij} = 1$), while all the other points have no influence in the determination of $\boldsymbol{\theta}_j$ ($u_{ij} = 0$).

CFO clustering algorithms: Final remarks (3)

The role of q in the possibilistic clustering

Consider the minimization problem for possibilistic clustering

$$minimize_{U,\Theta} J(\boldsymbol{u}_j, \boldsymbol{\theta}_j) = \sum_{i=1}^{N} {u_{ij}}^q d_{ij} + \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q$$

subject to $u_{ij} \in (0,1),\ i = 1, \dots, N, j = 1, \dots, m.$

- For $q > 1$, (for fixed $\boldsymbol{\theta}_j (\Rightarrow$ fixed $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \equiv d_{ij}$)) it is

$$u_{ij} = \frac{1}{1 + \left(\frac{d_{ij}}{\eta_j}\right)^{\frac{1}{q-1}}}$$

Thus, points for which $d_{ij} > \eta_j$ have $(0 <) u_{ij} < \frac{1}{2}$.

- Furthermore, as $q \to \infty$, (for fixed $\boldsymbol{\theta}_j (\Rightarrow$ fixed $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \equiv d_{ij}$)) it is

$$u_{ij} \to \frac{1}{2}$$

Thus, **all points** have the **same degree of compatibility** with **all clusters**.

CFO clustering algorithms: Final remarks (4)

The role of q in the parameters updating in fuzzy and possibilistic clustering

Consider the updating equation for the point representative case and the squared Euclidean distance case (fuzzy and 1st possibilistic clust. algorithms)

$$\boldsymbol{\theta}_j(t) = \frac{\sum_{i=1}^{N} u_{ij}{}^q(t-1)\boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}{}^q(t-1)}, j = 1, \dots, m$$

For $q > 1$, and since $u_{ij} \in (0,1)$, the previous observation indicates that the $\boldsymbol{x}_i$'s with high (low) u_{ij}, will have more (much less) significant contribution to the estimation of $\boldsymbol{\theta}_j(t)$, compared with the $q = 1$ case.

Example: Let $\boldsymbol{x}_1 = [0, 0]^T$ and $\boldsymbol{x}_2 = [10, 10]^T$, and $u_{1j} = 0.1$, $u_{2j} = 0.9$. Then

$$\boldsymbol{\theta}_j = \frac{u_{1j}\boldsymbol{x}_1 + u_{2j}\boldsymbol{x}_2}{u_{1j} + u_{2j}} = \begin{bmatrix} 9 \\ 9 \end{bmatrix} \quad (q = 1)$$

and

$$\boldsymbol{\theta}_j = \frac{u_{1j}{}^q\boldsymbol{x}_1 + u_{2j}{}^q\boldsymbol{x}_2}{u_{1j}{}^q + u_{2j}{}^q} = \begin{bmatrix} 9.9 \\ 9.9 \end{bmatrix} \quad (q = 2)$$

Hierarchical Clustering Algorithms

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

- ✓ They find applications in:
  - Social sciences
  - Biological taxonomy
  - Modern biology
  - Medicine
  - Archaeology
  - Computer science and engineering

Hierarchical Clustering Algorithms

Let $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_i = [x_{i1}, \dots, x_{il}]^T$.

Recall that:

- In hard clustering each vector belongs exclusively to a single cluster.
- An m-(hard) clustering of X, $\Re$, is a partition of X into m sets (clusters) $C_1, \dots, C_m$, so that:
  - $C_j \neq \emptyset, j = 1, \dots, m$
  - $\cup_{j=1}^{m} C_j = X$
  - $C_i \cap C_j = \emptyset, i \neq j, i, j = 1,2,, \dots, m$

By the definition: $\Re = \{C_j, j = 1, \dots m\}$

Hierarchical Clustering Algorithms

- **Definition:** A clustering $\Re_1$ consisting of k clusters is said to be nested in the clustering $\Re_2$ consisting of r $(< k)$ clusters, if **each** cluster in $\Re_1$ is a subset of a cluster in $\Re_2$.
  We write $\Re_1 \angle \Re_2$

Example: Let $\Re_1 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_3\}, \{\boldsymbol{x}_4\}, \{\boldsymbol{x}_2, \boldsymbol{x}_5\}\}$, $\Re_2 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_4\}, \{\boldsymbol{x}_2, \boldsymbol{x}_5\}\}$,

$\Re_3 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_4\}, \{\boldsymbol{x}_3\}, \{\boldsymbol{x}_2, \boldsymbol{x}_5\}\}$, $\Re_4 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4\}, \{\boldsymbol{x}_3, \boldsymbol{x}_5\}\}$.

It is $\Re_1 \angle \Re_2$, but not $\Re_1 \angle \Re_3$, $\Re_1 \angle \Re_4$, $\Re_1 \angle \Re_1$.

Hierarchical Clustering Algorithms

Remarks:

- Hierarchical clustering algorithms produce a hierarchy of nested clusterings.
- They involve N steps at the most.
- At each step t, the clustering $\Re_t$ is produced by $\Re_{t-1}$.

➢ Main strategies:

Agglomerative hierarchical clustering algorithms	**Divisive hierarchical clustering algorithms**
$\Re_0 = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$	$\Re_0 = \{\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}\}$
. . .	. . .
$\Re_{N-1} = \{\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}\}$	$\Re_{N-1} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$
$\Re_0 \angle \ \ldots \ \angle \Re_{N-1}$	$\Re_{N-1} \angle \ \ldots \ \angle \Re_0$

Agglomerative Clustering Algorithms

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

Generalized Agglomerative Scheme (GAS)

- **Initialization**
  - **Choose** $\Re_0 = \{\{\boldsymbol{x}_1\}, \dots, \{\boldsymbol{x}_N\}\}$
  - $t = 0$
- **Repeat**
  - $t = t + 1$
  - **Choose** (C_i, C_j) in $\Re_{t-1}$ such that

$$g(C_i, C_j) = \begin{cases} min_{r,s} g(C_r, C_s), & if\ \ g\ is\ a\ disim.function \\ max_{r,s} g(C_r, C_s), & if\ \ g\ is\ a\ sim.function \end{cases}$$

  - Define $C_q = C_i \cup C_j$ and produce $\Re_t = \left(\Re_{t-1} - \{C_i, C_j\}\right) \cup \{C_q\}$
- **Until** all vectors lie in a single cluster.

Agglomerative Clustering Algorithms

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: $O(N^3)$

Agglomerative Clustering Algorithms

Definitions of some useful quantities:

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$, with $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{il}]^T$.

- Pattern matrix $(D(X))$: An Nxl matrix whose i-th row is $\boldsymbol{x}_i$ (transposed).

- Proximity (similarity or dissimilarity) matrix $(P(X))$: An NxN matrix whose (i,j) element equals the proximity $\wp(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (similarity $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$, dissimilarity $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$).

Example 1: Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5\}$, with

$$\boldsymbol{x}_1 = [1,1]^T, \boldsymbol{x}_2 = [2,1]^T, \boldsymbol{x}_3 = [5,4]^T, \boldsymbol{x}_4 = [6,5]^T, \boldsymbol{x}_5 = [6.5,6]^T$$

Pattern matrix / Euclidean distance / Tanimoto distance

$$D(X) = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{bmatrix} \quad P(X) = \begin{bmatrix} 0 & 1 & 5 & 6.4 & 7.4 \\ 1 & 0 & 4.2 & 5.7 & 6.7 \\ 5 & 4.2 & 0 & 1.4 & 2.5 \\ 6.4 & 5.7 & 1.4 & 0 & 1.1 \\ 7.4 & 6.7 & 2.5 & 1.1 & 0 \end{bmatrix} \quad P'(X) = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1 & 0.44 & 0.35 & 0.20 \\ 0.26 & 0.44 & 1 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1 & 0.98 \\ 0.18 & 0.20 & 0.90 & 0.98 & 1 \end{bmatrix}$$

Agglomerative Clustering Algorithms

Definitions of some useful quantities:

➢Threshold dendrogram (or dendrorgram): It is an effective way of **representing the sequence of clusterings**, which are produced by an agglomerative algorithm.

Example 1 (cont.): If $d_{min}^{ss}(C_i, C_j)$ 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:

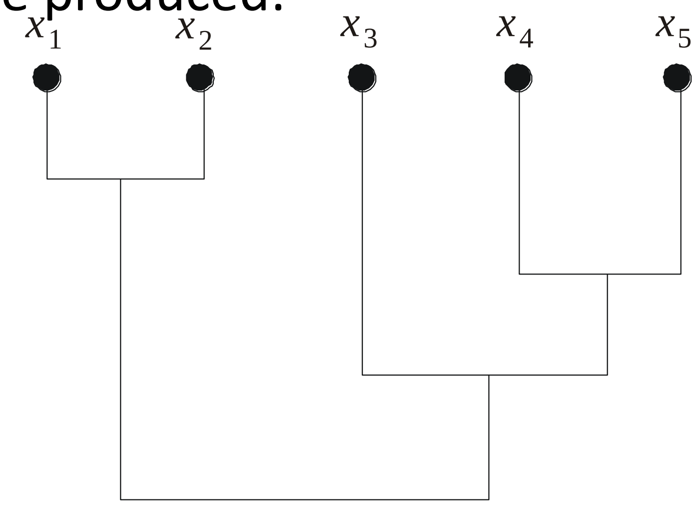
$$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$$

$$\{\{x_1,x_2\},\{x_3\},\{x_4\},\{x_5\}\}$$

$$\{\{x_1,x_2\},\{x_3\},\{x_4,x_5\}\}$$

$$\{\{x_1,x_2\},\{x_3,x_4,x_5\}\}$$

$$\{\{x_1,x_2,x_3,x_4,x_5\}\}$$



$$D(X)=\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{bmatrix}$$

$$P(X)=\begin{bmatrix} 0 & 1 & 5 & 6.4 & 7.4 \\ 1 & 0 & 4.2 & 5.7 & 6.7 \\ 5 & 4.2 & 0 & 1.4 & 2.5 \\ 6.4 & 5.7 & 1.4 & 0 & 1.1 \\ 7.4 & 6.7 & 2.5 & 1.1 & 0 \end{bmatrix}$$

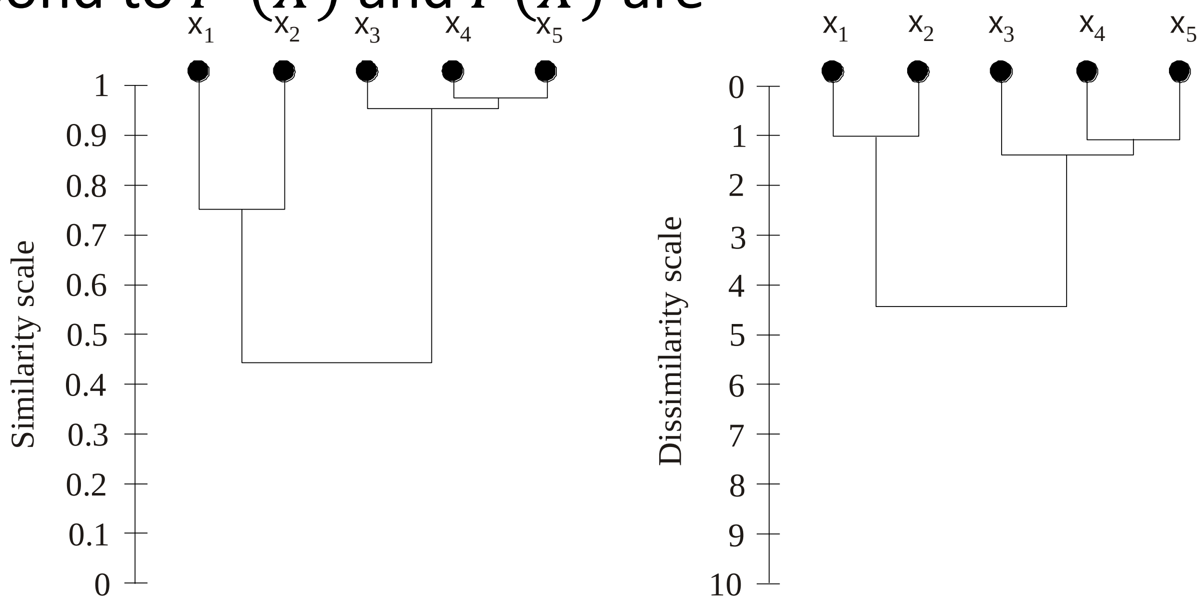
Agglomerative Clustering Algorithms

Definitions of some useful quantities:

- Proximity (dissimilarity or similarity) dendrogram: A **dendrogram** that takes into account the **level of proximity** (dissimilarity or similarity) where two clusters are merged for the first time.

Example 1 (cont.): In terms of the previous example, the proximity dendrograms that correspond to $P'(X)$ and $P(X)$ are

$$P(X) = \begin{bmatrix} 0 & 1 & 5 & 6.4 & 7.4 \\ 1 & 0 & 4.2 & 5.7 & 6.7 \\ 5 & 4.2 & 0 & 1.4 & 2.5 \\ 6.4 & 5.7 & 1.4 & 0 & 1.1 \\ 7.4 & 6.7 & 2.5 & 1.1 & 0 \end{bmatrix}$$



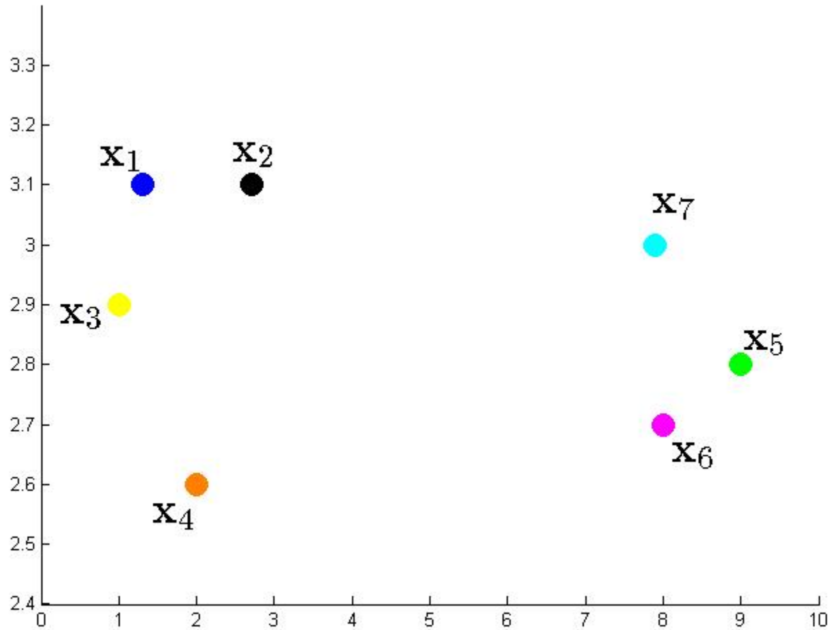
(a) (b)

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 Clustering Algorithms

Example:

$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_4$ $\mathbf{x}_5$ $\mathbf{x}_6$ $\mathbf{x}_7$

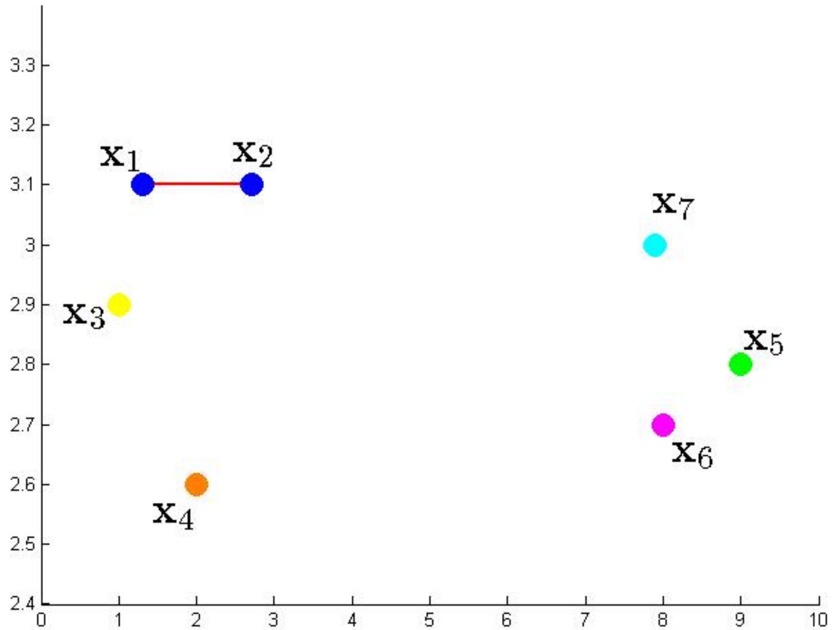


Agglomerative philosophy:

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

Example:

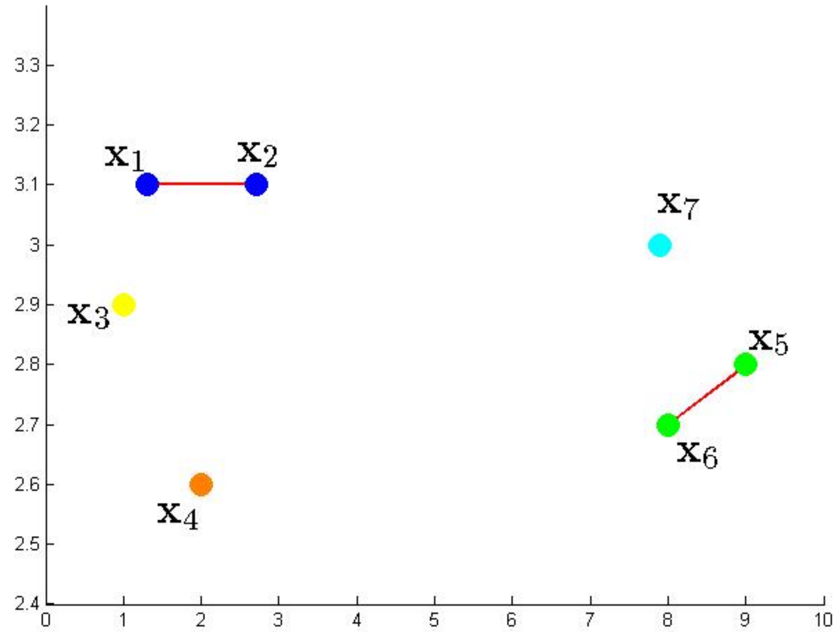


<u>Agglomerative philosophy:</u>

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

Example:

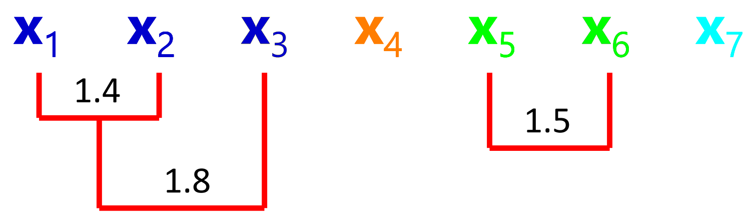


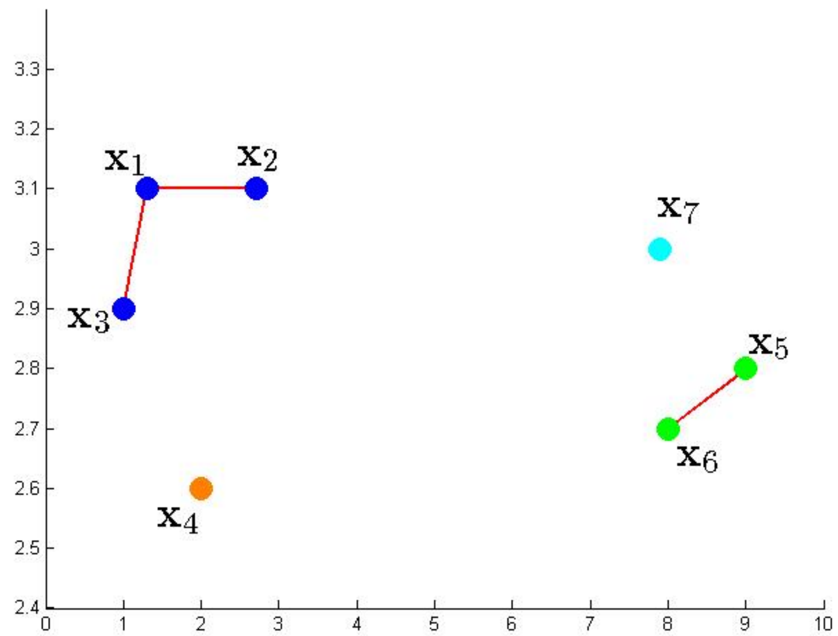
<u>Agglomerative philosophy:</u>

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

Example:



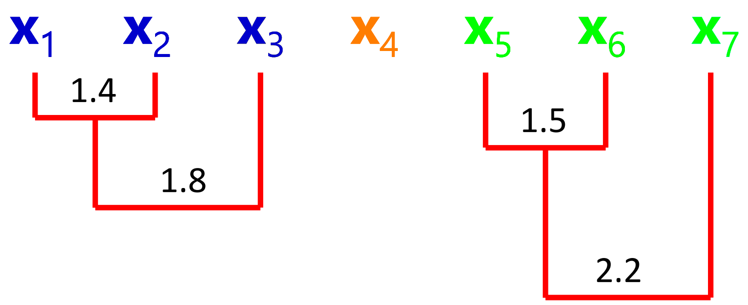


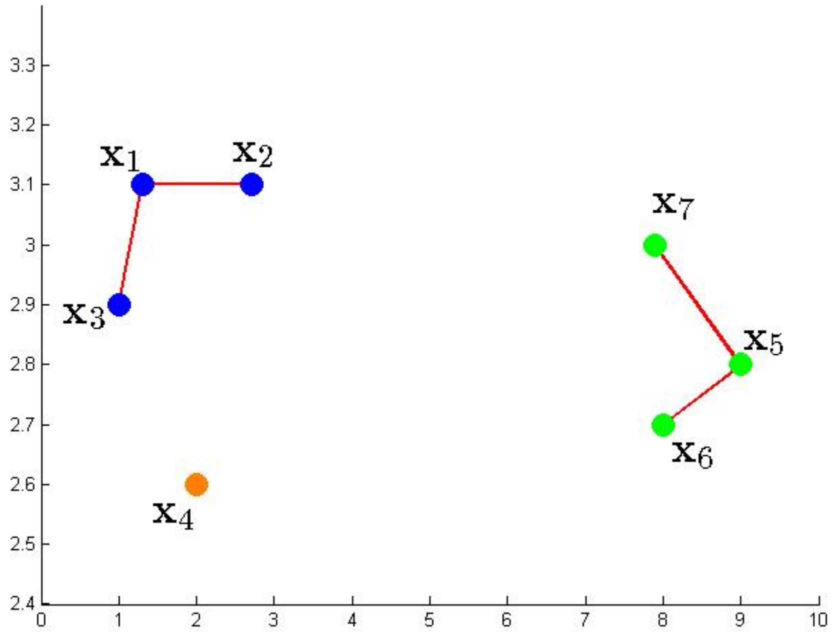
<u>Agglomerative philosophy:</u>

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

Example:



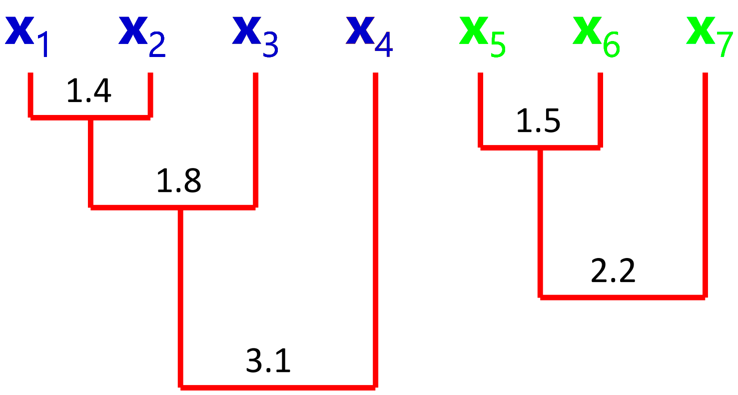


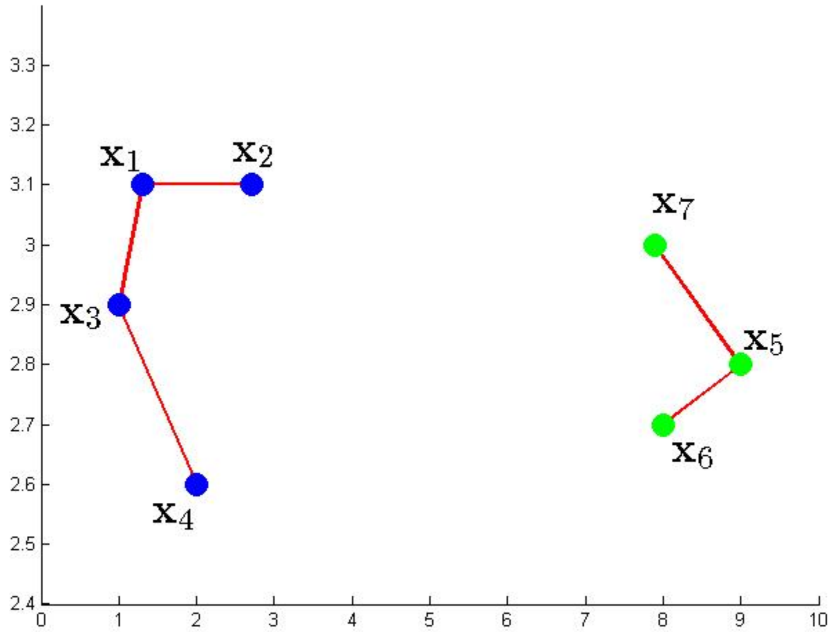
<u>Agglomerative philosophy:</u>

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

Example:



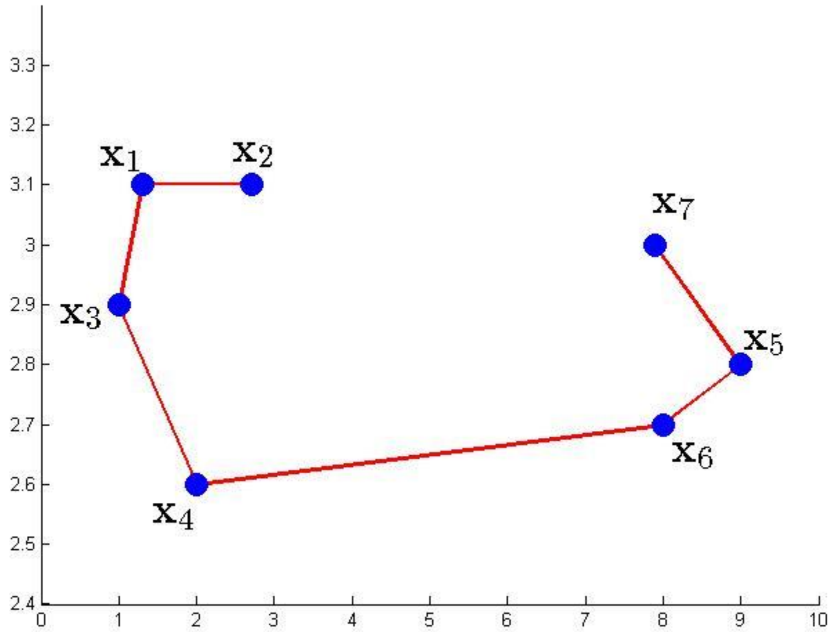


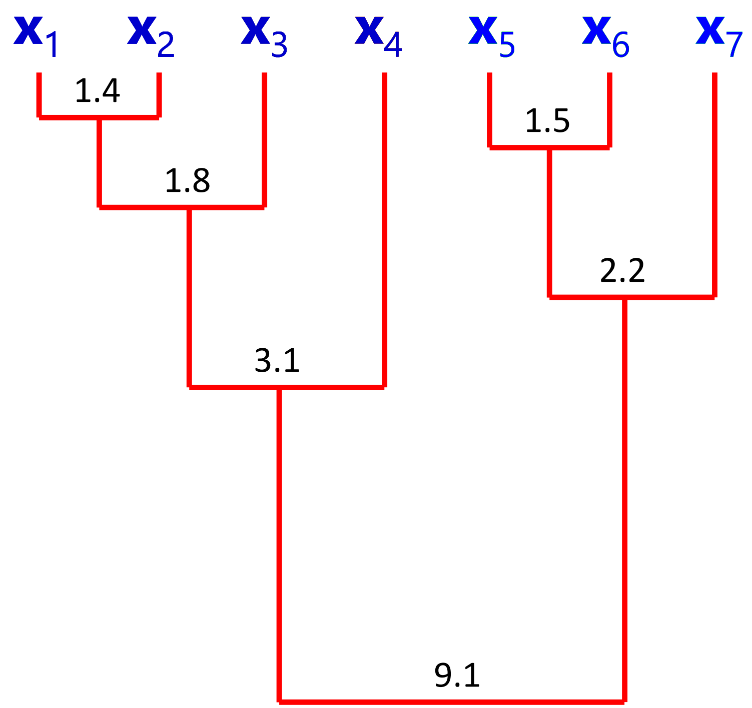
Agglomerative philosophy:

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

Example:



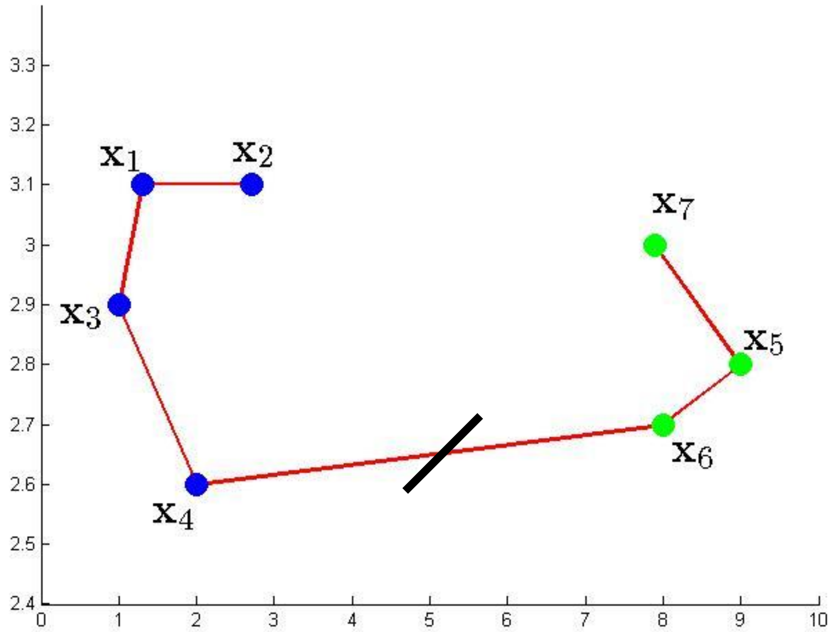


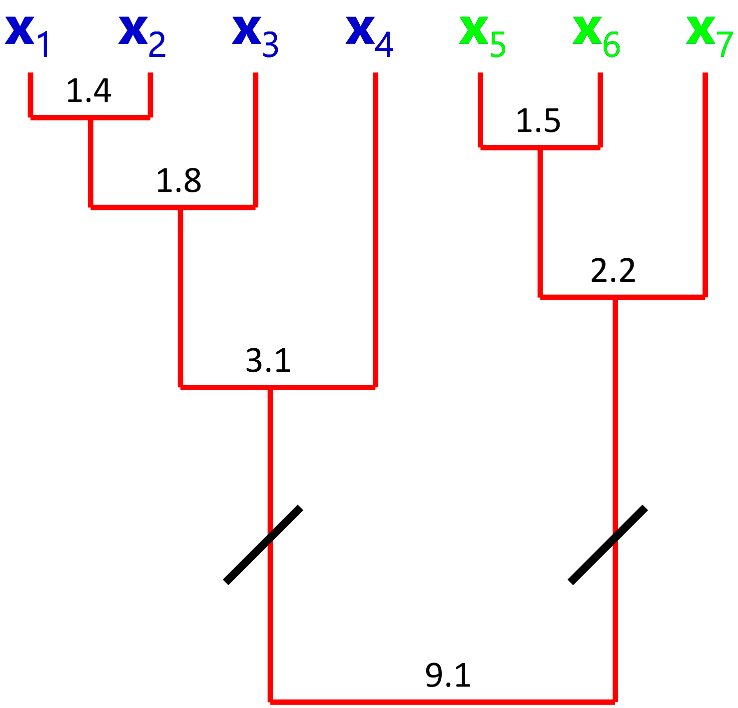
Agglomerative philosophy:

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

Example:





Agglomerative philosophy:

- In the initial clustering all data vectors **belong** to different clusters.
- At each step a new clustering is defined by **merging** the two most similar clusters to one.
- At the final clustering all vectors **belong** to the same cluster.

Agglomerative Clustering Algorithms

According to the mathematical tools used for their expression, **agglomerative algorithms** are divided into:

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

NOTE: In the sequel we consider only 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 each of the remaining clusters C_s, via a relation of the form

$$d(C_q, C_s) = f(d(C_i, C_s), d(C_j, C_s), d(C_i, C_j))$$

Agglomerative matrix theory based Clustering Algorithms

- A number of distance functions comply with the following update equation

$C_q = C_i \cup C_j$

$$d(C_q, C_s) = a_i d(C_i, C_s) + a_j (d(C_j, C_s) + b d(C_i, C_j) + c|d(C_i, C_s) - d(C_j, C_s)| \quad \textbf{(1)}$$

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

$$d(C_q, C_s) = \min\{d(C_i, C_s), d(C_j, C_s)\} \quad \textbf{(2)}$$

➢ Complete link (CL) algorithm ($a_i = 1/2, a_j = 1/2, b = 0, c = 1/2$). In this case

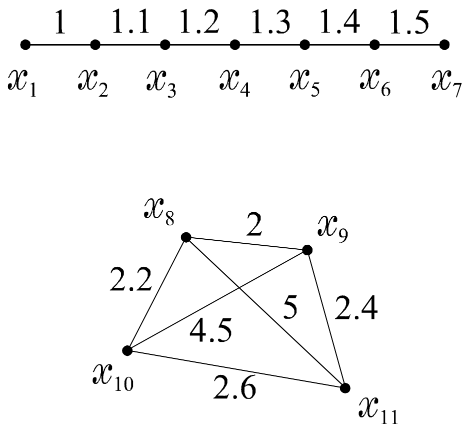
$$d(C_q, C_s) = \max\{d(C_i, C_s), d(C_j, C_s)\}$$

Remarks:

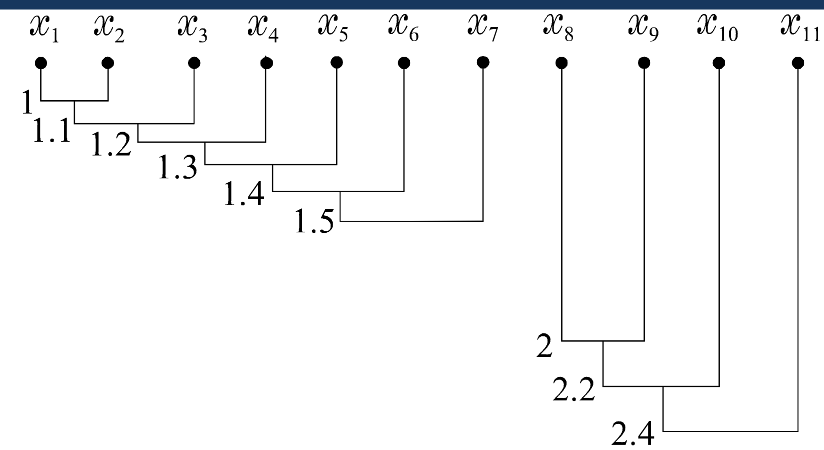
- Single link forms clusters at low dissimilarities while complete link forms clusters at high dissimilarities.
- Single link tends to form elongated clusters (*chaining effect*) while complete link tends to form compact clusters.
- The rest algorithms are compromises between these two extremes.

Agglomerative matrix theory based Clustering Algorithms

Example:



(a)

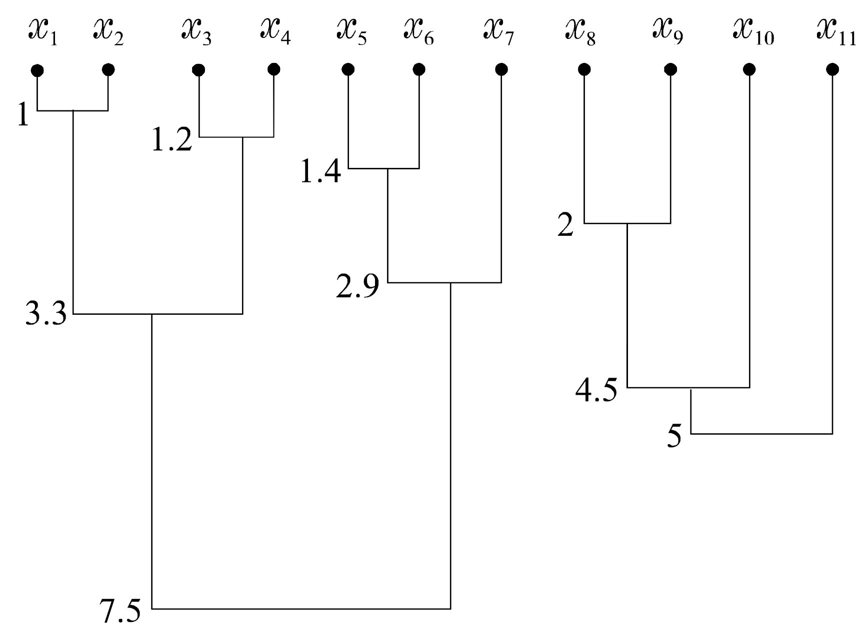


(b)

(a) The data set X.

(b) The single link algorithm dissimilarity dendrogram.

(c) The complete link algorithm dissimilarity dendrogram.



(c)

Agglomerative matrix theory based Clustering Algorithms

- Weighted Pair Group Method Average (WPGMA) ($a_i = 1/2, a_j = 1/2, b = 0, c = 0$). In this case:

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

$$d(C_q, C_s) = \frac{1}{2}(d(C_i, C_s) + d(C_j, C_s))$$

- 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:

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

- 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, **if** d_{ij} is defined as the squared Euclidean distance between the means of C_i and C_j,
then it holds that $d_{qs} = ||\boldsymbol{m}_q - \boldsymbol{m}_s||^2$, where $\boldsymbol{m}_q$, $\boldsymbol{m}_s$ are the means of C_q, C_s, respectively.

Agglomerative matrix theory based Clustering Algorithms

- Weighted Pair Group Method Centroid (WPGMC) ($a_i = 1/2, a_j = 1/2, b = -1/4, c = 0$). In this case

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

$$d_{qs} = \frac{1}{2} d_{is} + \frac{1}{2}\, d_{js} - \frac{1}{4} d_{ij}$$

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

$$d'_{ij} = \frac{n_i n_j}{n_i + n_j}\, ||\boldsymbol{m}_i - \boldsymbol{m}_j||^2 \qquad \textbf{(3)}$$

d'_{qs} can be expressed in terms of $d'_{is}, d'_{js}, d'_{ij}$ as

$$d'_{qs} = \frac{n_i + n_s}{n_i + n_j + n_s} d'_{is} + \frac{n_j + n_s}{n_i + n_j + n_s} d'_{js} - \frac{n_s}{n_i + n_j + n_s} d'_{ij}$$

Remark: Ward's algorithm forms $\Re_{t+1}$ by merging the two clusters that lead to the smallest possible increase of the total variance, i.e.,

$$E_t = \sum_{r=1}^{N-t} \sum_{\boldsymbol{x} \in C_r} ||\boldsymbol{x} - \boldsymbol{m}_r||^2$$

Agglomerative matrix theory based Clustering Algorithms

Example 3: Consider the following dissimilarity matrix (Euclidean distance)

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

$\Re_0=\{\{\underline{x}_1\}, \{\underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4\}, \{\underline{x}_5\}\}$,
$\Re_1=\{\{\underline{x}_1, \underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4\}, \{\underline{x}_5\}\}$,
$\Re_2=\{\{\underline{x}_1, \underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4, \underline{x}_5\}\}$,
$\Re_3=\{\{\underline{x}_1, \underline{x}_2, \underline{x}_3\}, \{\underline{x}_4, \underline{x}_5\}\}$,
$\Re_4=\{\{\underline{x}_1, \underline{x}_2, \underline{x}_3, \underline{x}_4, \underline{x}_5\}\}$

All the algorithms produce the same sequence of clusterings shown above, yet at different proximity levels:

	SL	*CL*	*WPGMA*	*UPGMA*	*WPGMC*	*UPGMC*	*Ward*
$\Re_0$	0	0	0	0	0	0	0
$\Re_1$	1	1	1	1	1	1	0.5
$\Re_2$	1.5	1.5	1.5	1.5	1.5	1.5	0.75
$\Re_3$	2	3	2.5	2.5	2.25	2.25	1.5
$\Re_4$	16	37	25.75	27.5	24.69	26.46	31.75

Agglomerative matrix theory based Clustering Algorithms

Example 3 (in detail): (a) The single-link case

$(C_q = C_i \cup C_j,\ d(C_q, C_s) = \min(d(C_i, C_s), d(C_j, C_s))$

P_0:

	$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1\}$	0	**1**	2	26	37
$\{x_2\}$	**1**	0	3	25	36
$\{x_3\}$	2	3	0	16	25
$\{x_4\}$	26	25	16	0	1.5
$\{x_5\}$	37	36	25	1.5	0

	$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1\}$	0	**1**	2	26	37
$\{x_2\}$	**1**	0	3	25	36
$\{x_3\}$	2	3	0	16	25
$\{x_4\}$	26	25	16	0	1.5
$\{x_5\}$	37	36	25	1.5	0

$d(\{x_1, x_2\}, \{x_3\}) = \min(d(\{x_1\}, \{x_3\}), d(\{x_2\}, \{x_3\}))$
$= \min(2,3) = 2$

$d(\{x_1, x_2\}, \{x_4\}) =$
$\min(26,25) = 25$

$d(\{x_1, x_2\}, \{x_5\}) =$
$\min(37,36) = 36$

P_1:

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1, x_2\}$	0	2	25	36
$\{x_3\}$	2	0	16	25
$\{x_4\}$	25	16	0	**1.5**
$\{x_5\}$	36	25	**1.5**	0

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1, x_2\}$	0	2	25	36
$\{x_3\}$	2	0	16	25
$\{x_4\}$	25	16	0	**1.5**
$\{x_5\}$	36	25	**1.5**	0

$d(\{x_1, x_2\}, \{x_4, x_5\}) =$
$\min(25,36) = 25$

$d(\{x_3\}, \{x_4, x_5\}) =$
$\min(16,25) = 16$

Agglomerative matrix theory based Clustering Algorithms

Example 3 (in detail): (a) The single-link case

$(C_q = C_i \cup C_j,\ d(C_q, C_s) = \min(d(C_i, C_s), d(C_j, C_s))$

P_2:

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2\}$	0	**2**	25
$\{x_3\}$	**2**	0	16
$\{x_4, x_5\}$	25	16	0

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2\}$	0	**2**	25
$\{x_3\}$	**2**	0	16
$\{x_4, x_5\}$	25	16	0

$d(\{x_1, x_2, x_3\}, \{x_4, x_5\}) =$
$= \min(25{,}16) = 16$

P_3:

	$\{x_1, x_2, x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2, x_3\}$	0	**16**
$\{x_4, x_5\}$	**16**	0

	$\{x_1, x_2, x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2, x_3\}$	0	**16**
$\{x_4, x_5\}$	**16**	0

P_4:

	$\{x_1, x_2, x_3, x_4, x_5\}$
$\{x_1, x_2, x_3, x_4, x_5\}$	0

$\mathcal{R}_0 = \{\{\underline{x}_1\}, \{\underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4\}, \{\underline{x}_5\}\}$, $(\mathbf{0})$

$\mathcal{R}_1 = \{\{\underline{x}_1, \underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4\}, \{\underline{x}_5\}\}$, $(\mathbf{1})$

$\mathcal{R}_2 = \{\{\underline{x}_1, \underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4, \underline{x}_5\}\}$, $(\mathbf{1.5})$

$\mathcal{R}_3 = \{\{\underline{x}_1, \underline{x}_2, \underline{x}_3\}, \{\underline{x}_4, \underline{x}_5\}\}$, $(\mathbf{2})$

$\mathcal{R}_4 = \{\{\underline{x}_1, \underline{x}_2, \underline{x}_3, \underline{x}_4, \underline{x}_5\}\}$, $(\mathbf{16})$

Agglomerative matrix theory based Clustering Algorithms

Example 3 (in detail): (b) The complete-link case

$(C_q = C_i \cup C_j,\ d(C_q, C_s) = \max(d(C_i, C_s), d(C_j, C_s))$

P_0:

	$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1\}$	0	**1**	2	26	37
$\{x_2\}$	**1**	0	3	25	36
$\{x_3\}$	2	3	0	16	25
$\{x_4\}$	26	25	16	0	1.5
$\{x_5\}$	37	36	25	1.5	0

	$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1\}$	0	**1**	2	26	37
$\{x_2\}$	**1**	0	3	25	36
$\{x_3\}$	2	3	0	16	25
$\{x_4\}$	26	25	16	0	1.5
$\{x_5\}$	37	36	25	1.5	0

$d(\{x_1, x_2\}, \{x_3\}) = \max(d(\{x_1\},\{x_3\}), d(\{x_2\},\{x_3\})$
$= \max(2,3) = 3$

$d(\{x_1, x_2\}, \{x_4\}) =$
$\max(26,25) = 26$

$d(\{x_1, x_2\}, \{x_5\}) =$
$\max(37,36) = 37$

P_1:

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1, x_2\}$	0	3	26	37
$\{x_3\}$	3	0	16	25
$\{x_4\}$	26	16	0	**1.5**
$\{x_5\}$	37	25	**1.5**	0

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4\}$	$\{x_5\}$
$\{x_1, x_2\}$	0	3	26	37
$\{x_3\}$	3	0	16	25
$\{x_4\}$	26	16	0	**1.5**
$\{x_5\}$	37	25	**1.5**	0

$d(\{x_1, x_2\}, \{x_4, x_5\}) =$
$\max(26,37) = 37$

$d(\{x_3\}, \{x_4, x_5\}) =$
$\max(16,25) = 25$

Agglomerative matrix theory based Clustering Algorithms

Example 3 (in detail): (b) The complete-link case

$(C_q = C_i \cup C_j,\ d(C_q, C_s) = \max(d(C_i, C_s), d(C_j, C_s))$

P_2:

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2\}$	0	**3**	37
$\{x_3\}$	**3**	0	25
$\{x_4, x_5\}$	37	25	0

	$\{x_1, x_2\}$	$\{x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2\}$	0	**3**	37
$\{x_3\}$	**3**	0	25
$\{x_4, x_5\}$	37	25	0

$d(\{x_1, x_2, x_3\}, \{x_4, x_5\}) =$
$= \max(37{,}25) = 37$

P_3:

	$\{x_1, x_2, x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2, x_3\}$	0	**37**
$\{x_4, x_5\}$	**37**	0

	$\{x_1, x_2, x_3\}$	$\{x_4, x_5\}$
$\{x_1, x_2, x_3\}$	0	**37**
$\{x_4, x_5\}$	**37**	0

P_4:

	$\{x_1, x_2, x_3, x_4, x_5\}$
$\{x_1, x_2, x_3, x_4, x_5\}$	0

$\mathfrak{R}_0 = \{\{\underline{x}_1\}, \{\underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4\}, \{\underline{x}_5\}\}, (\mathbf{0})$

$\mathfrak{R}_1 = \{\{\underline{x}_1, \underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4\}, \{\underline{x}_5\}\}, (\mathbf{1})$

$\mathfrak{R}_2 = \{\{\underline{x}_1, \underline{x}_2\}, \{\underline{x}_3\}, \{\underline{x}_4, \underline{x}_5\}\}, (\mathbf{1.5})$

$\mathfrak{R}_3 = \{\{\underline{x}_1, \underline{x}_2, \underline{x}_3\}, \{\underline{x}_4, \underline{x}_5\}\}, (\mathbf{3})$

$\mathfrak{R}_4 = \{\{\underline{x}_1, \underline{x}_2, \underline{x}_3, \underline{x}_4, \underline{x}_5\}\}, (\mathbf{37})$

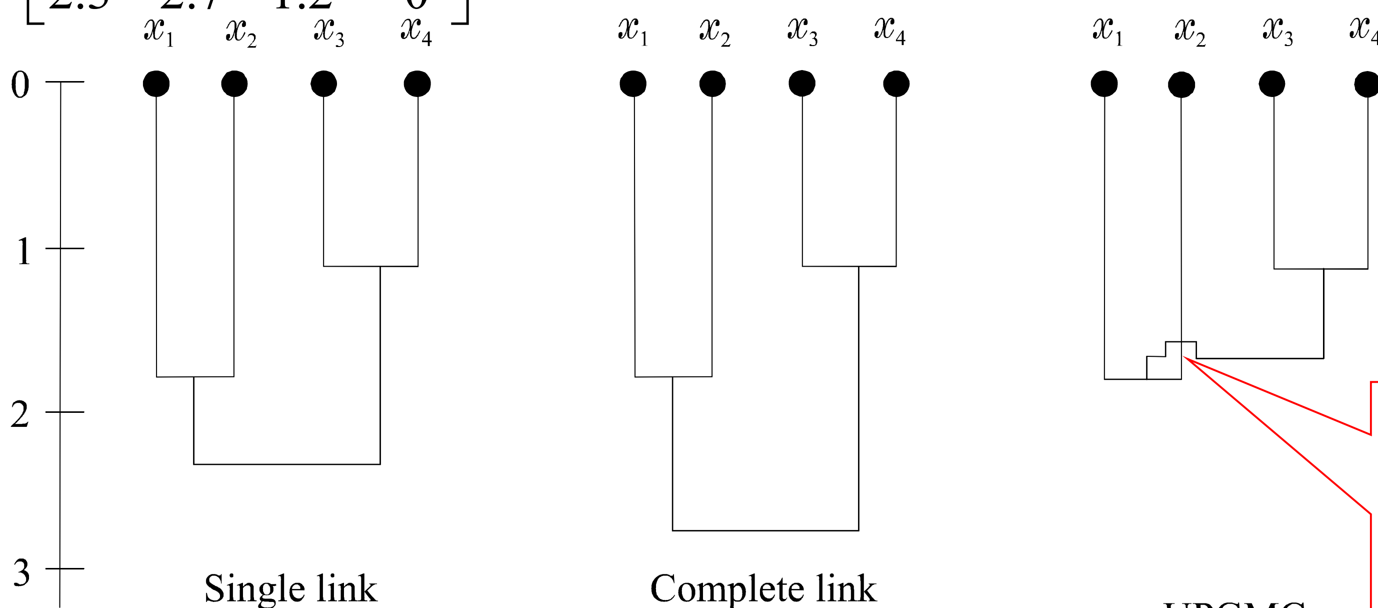
Agglomerative matrix theory based Clustering Algorithms

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:



Single link (a)

Complete link (b)

UPGMC algorithm (c)

$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}$ is formed at lower dissimilarity level than $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ (crossover)

Agglomerative matrix theory based Clustering Algorithms

Example (in detail): The WPGMC case

$(C_q = C_i \cup C_j,\ d_{qs} = \frac{1}{2}d_{is} + \frac{1}{2}d_{js} - \frac{1}{4}d_{ij})$

P_0:

	$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_4\}$
$\{x_1\}$	0	1.8	2.4	2.3
$\{x_2\}$	1.8	0	2.5	2.7
$\{x_3\}$	2.4	2.5	0	**1.2**
$\{x_4\}$	2.3	2.7	**1.2**	0

	$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_4\}$
$\{x_1\}$	0	1.8	2.4	2.3
$\{x_2\}$	1.8	0	2.5	2.7
$\{x_3\}$	2.4	2.5	0	**1.2**
$\{x_4\}$	2.3	2.7	**1.2**	0

$$d_{(3,4),1} = \frac{1}{2}d_{3,1} + \frac{1}{2}d_{4,1} - \frac{1}{4}d_{3,4} = \frac{1}{2}2.4 + \frac{1}{2}2.3 - \frac{1}{4}1.2 = 2.05$$

$$d_{(3,4),2} = \frac{1}{2}d_{3,2} + \frac{1}{2}d_{4,2} - \frac{1}{4}d_{3,4} = \frac{1}{2}2.5 + \frac{1}{2}2.7 - \frac{1}{4}1.2 = 2.3$$

P_1:

	$\{x_1\}$	$\{x_2\}$	$\{x_3, x_4\}$
$\{x_1\}$	0	**1.8**	2.05
$\{x_2\}$	**1.8**	0	2.3
$\{x_3, x_4\}$	2.05	2.3	0

	$\{x_1\}$	$\{x_2\}$	$\{x_3, x_4\}$
$\{x_1\}$	0	**1.8**	2.05
$\{x_2\}$	**1.8**	0	2.3
$\{x_3, x_4\}$	2.05	2.3	0

$\mathfrak{R}_0 = \{\{\underline{x}_1\},\{\underline{x}_2\},\{\underline{x}_3\},\{\underline{x}_4\}\}$, $(\mathbf{0})$

$\mathfrak{R}_1 = \{\{\underline{x}_1\},\{\underline{x}_2\},\{\underline{x}_3,\underline{x}_4\}\}$, $(\mathbf{1.2})$

$\mathfrak{R}_2 = \{\{\underline{x}_1,\underline{x}_2\},\{\underline{x}_3,\underline{x}_4\}\}$, $(\mathbf{1.8})$

$\mathfrak{R}_3 = \{\{\underline{x}_1,\underline{x}_2,\underline{x}_3,\underline{x}_4\}\}$, $(\mathbf{1.275}$!!$)$

$$d_{(1,2),(3,4)} = \frac{1}{2}d_{1,(3,4)} + \frac{1}{2}d_{2,(3,4)} - \frac{1}{4}d_{1,2} = \frac{1}{2}2.05 + \frac{1}{2}2.3 - \frac{1}{4}1.8 = 1.275$$

P_2:

	$\{x_1, x_2\}$	$\{x_3, x_4\}$
$\{x_1, x_2\}$	0	**1.275**
$\{x_3, x_4\}$	**1.275**	0

P_3:

	$\{x_1, x_2, x_3, x_4\}$
$\{x_1, x_2, x_3, x_4\}$	0

Agglomerative matrix theory based Clustering Algorithms

- **Monotonicity condition:**

If clusters C_i and C_j are selected to be merged in cluster C_q, at the tth 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 clustering 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 satisfy** it.

Agglomerative matrix theory based Clustering Algorithms

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.