Clustering algorithms

Konstantinos Koutroumbas

Unit 7

– Possibilistic CFO clustering algorithms
– Discussion on CFO clust. Algorithms
– Introduction to hierarchical clustering algorithms

koutroum@noa.gr

Possibilistic CFO clustering algorithms

Possibilistic clustering algorithms:

Let $X = \{x_1, x_2, \dots, x_N\}$ be a set of data points.

For each vector $\boldsymbol{x}_i$ its degree of compatibility with all clusters, $u_{ij}, j = 1, \dots, m$, is considered.

The constraints on u_{ij}'s are

- $u_{ij} \in [0,1], \; i = 1, \dots, N, j = 1, \dots, m$
- $0 < \sum_{i=1}^{N} u_{ij} < N, \; j = 1, \dots, m$

Each cluster is **represented** by a representative $\boldsymbol{\theta}_j$ (point repr., hyperplane…).

Let $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_m\}$

Define the cost function

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

When $J_q(U, \Theta)$ is **minimized**?

When all u_{ij}'s are (very close to) zero.

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

A possible scenario:

Minimize the cost function

$$J_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}{}^q\, d\left(\boldsymbol{x}_i,\boldsymbol{\theta}_j\right) + \sum_{j=1}^{m}\eta_j\sum_{i=1}^{N}(1-u_{ij})^q$$

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed $\boldsymbol{\theta}_j$'s:** Equating the partial derivative of $J_q(U,\Theta)$ wrt u_{ij} to 0 we obtain

$$\frac{\partial J_q(U,\Theta)}{\partial u_{ij}} = 0 \Leftrightarrow u_{ij} = \frac{1}{1+\left(\frac{d(\boldsymbol{x}_i,\boldsymbol{\theta}_j)}{\eta_j}\right)^{\frac{1}{q-1}}}$$

Notes: (a) u_{ij} **depends exclusively** on $\boldsymbol{\theta}_j$.

(b) It is $u_{ij} \in [0,1]$

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

A possible scenario:

Minimize the cost function

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

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed u_{ij}'s:** Solve the following m independent minimization problems

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^{q} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

Possibilistic CFO clustering algorithms

Generalized Possibilistic Algorithmic Scheme (GPAS1)

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

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

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

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

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

Possibilistic CFO clustering algorithms

Remarks:

- A candidate termination condition is

$$||\boldsymbol{\theta}(t)\text{-}\boldsymbol{\theta}(t\text{-}1)||<\varepsilon,$$

  where ||.|| is any vector norm and ε a user-defined constant.

- GFAS may also be initialized from $U(0)$ instead of $\boldsymbol{\theta}_j(0)$, $j=1,\ldots,m$ and start iterations with computing $\boldsymbol{\theta}_j$ first.

- Based on GPAS, a possibilistic algorithm can be derived, for each fuzzy clustering algorithm derived previously.

- High values of q:
  - In possibilistic clustering **cause** almost equal contributions of all vectors to all clusters
  - In fuzzy clustering cause increased sharing of the vectors among all clusters.

Possibilistic CFO clustering algorithms

Three observations

- Decomposition of $J(\Theta,U)$:

Since for each vector $\boldsymbol{x}_i$, u_{ij}'s, $j = 1, \ldots, m$ are independent from each other, $J(\Theta, U)$ can be written as

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

$$= \sum_{j=1}^{m} \left[\sum_{i=1}^{N} u_{ij}{}^{q} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q \right] \equiv \sum_{j=1}^{m} J_j$$

where

$$J_j = \sum_{i=1}^{N} u_{ij}{}^{q} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q$$

Each J_j is **associated** with a different cluster and *minimization of $J(\Theta, U)$ with respect to u_{ij}'s can be carried out separately for each J_j.*

Possibilistic CFO clustering algorithms

Three observations

- About η_j's:

–They **determine** the relative significance of the two terms in $J(\Theta,U)$.

–They are **related** to the "variance" of the points of C_j's, $j=1,\ldots,m$, around their centers.

–Two scenarios for the estimation of η_j's, for the point representatives case, are the following:

o **Run** the related FCM algorithm and after its convergence estimate η_j's as

$$\eta_j = \frac{\sum_{i=1}^{N} {u_{ij}}^q d(\boldsymbol{x}_i,\boldsymbol{\theta}_j)}{\sum_{i=1}^{N} {u_{ij}}^q} \quad \text{or} \quad \eta_j = \frac{\sum_{u_{ij}>a} d(\boldsymbol{x}_i,\boldsymbol{\theta}_j)}{\sum_{u_{ij}>a} 1}$$

○ **Set** $\eta_j = \eta = \frac{\beta}{q\sqrt{m}}$, where $\beta = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}\|^2$ and $\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$

Possibilistic CFO clustering algorithms

Three observations

- The mode-seeking property

Unlike Hard and fuzzy clustering algorithms which are partition algorithms (they terminate with the predetermined number of clusters no matter how many physical clusters are naturally formed in X), GPAS is a mode-seeking algorithm (it searches for dense regions of vectors in X).

Advantage: The number of clusters need not be a priori known.
If the number of clusters in GPAS, m, is greater than the true number of clusters k in X, some representatives will coincide with others. If $m<k$, ***some*** (and not all) of the clusters will be identified.

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

Another possible scenario:

Minimize the cost function

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

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed $\boldsymbol{\theta}_j$'s:** Equating the partial derivative of $J(U,\Theta)$ wrt u_{ij} to 0 we obtain

$$\frac{\partial J_q(U,\Theta)}{\partial u_{ij}} = 0 \Leftrightarrow u_{ij} = exp\left(-\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\eta_j}\right)$$

Notes: (a) u_{ij} **depends exclusively** on $\boldsymbol{\theta}_j$.

(b) It is $u_{ij} \in [0,1]$

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

A possible scenario:

Minimize the cost function

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

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed u_{ij}'s:** Solve the following m independent minimization problems

$$\boldsymbol{\theta}_j=argmin_{\boldsymbol{\theta}_j}\sum_{i=1}^{N}u_{ij}d(\boldsymbol{x}_i,\boldsymbol{\theta}_j)$$

Possibilistic CFO clustering algorithms

Generalized Possibilistic Algorithmic Scheme (GPAS2)

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

$$u_{ij}(t) = exp\left(-\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{\eta_j}\right)$$

    - End {For-j}
  - End {For-i}

  - t=t+1

  - For j=1 to m *% Parameter updating*
    - Set

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

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

CFO clustering algorithms: A unified view

Basic parameters – notation (cont.)

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

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

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

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

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

CFO clustering algorithms: A unified view

Aim:

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

How this is achieved:

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

$$J(\Theta,U)=\sum_{i=1}^{N}\sum_{j=1}^{m}u_{ij}^{q}d(x_i,\vartheta_j) \quad (q\geq 1)$$

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

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

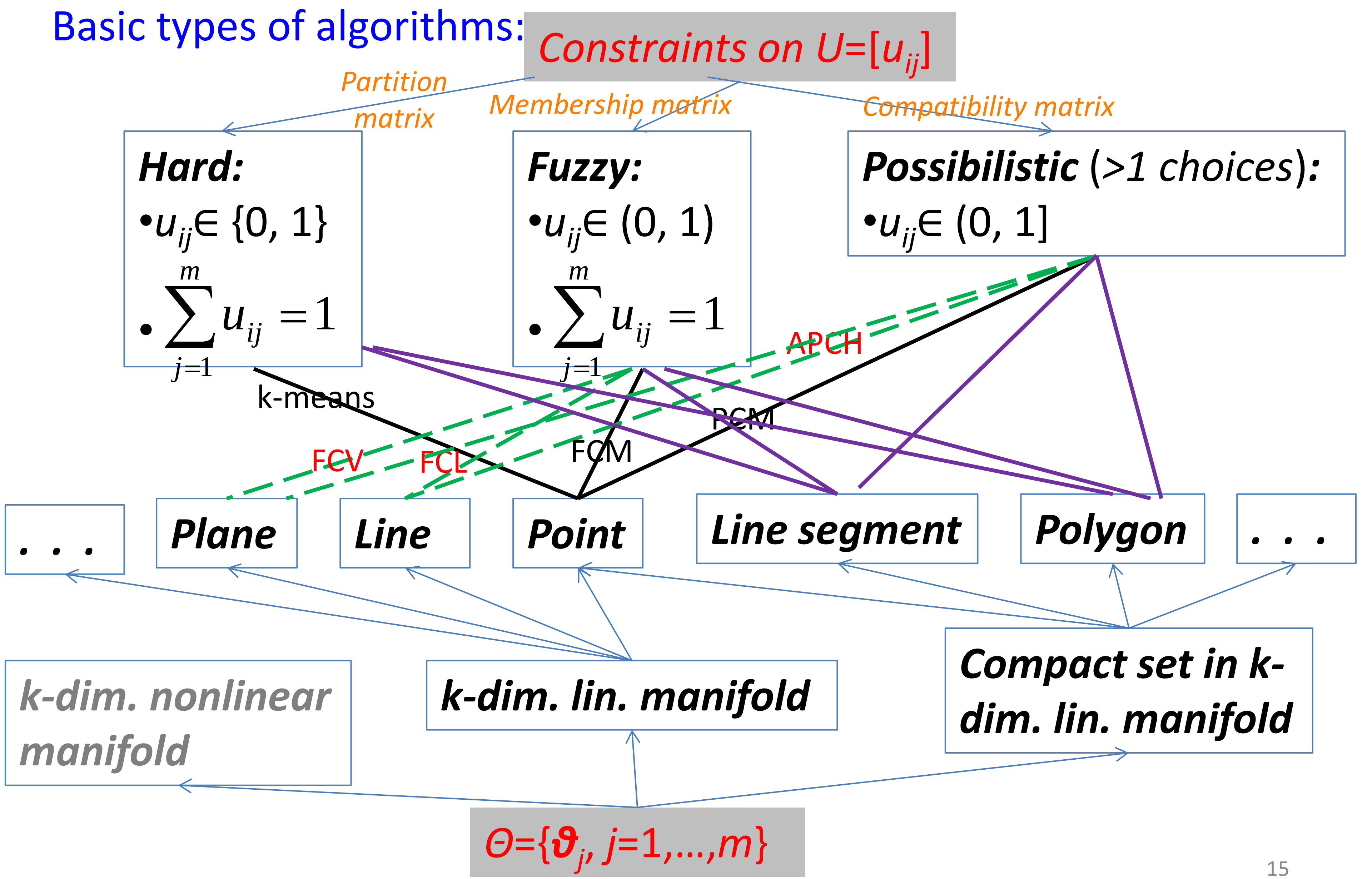
Intuition:

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

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

- ✓ For fixed u_{ij}'s, **minimization** of $J(\Theta,U)$ wrt $\boldsymbol{\vartheta}_j$'s leads $\boldsymbol{\vartheta}_j$'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

C(U)

ϑ_j

	Hard Constr.	Fuzzy Constr.	Possib. 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:

- ✓ *Initialize* $\Theta=\Theta(0)$
- ✓ **Repeat**
  - $t=0$
  - $U(t) = argmin_U \ J(\Theta(t),U)$, 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

"Array of CFO algorithms"

C(U)

ϑ_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”

C(U)

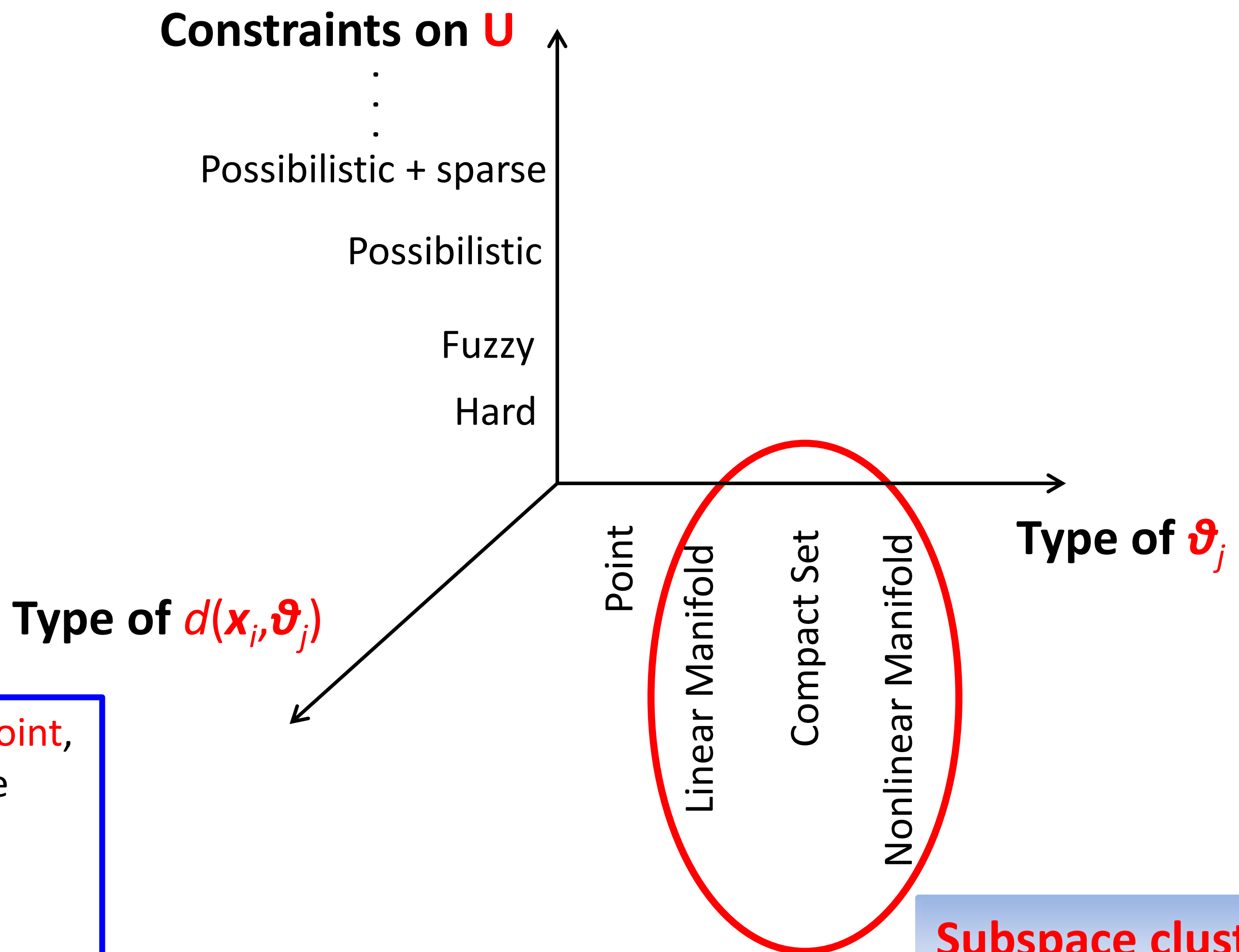
ϑ_j

	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



E.g.:If $\boldsymbol{\vartheta}_j$ is a point, $d(\mathbf{x}_i,\boldsymbol{\vartheta}_j)$ 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)** $u_{ij} \in \{0,1\}$, $i = 1,\dots,N, j = 1,\dots,m$, and **(b)** $\sum_{j=1}^{m} u_{ij} = 1, i = 1,\dots,N$.

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

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

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

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

  - 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

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

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

- **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

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

  - 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

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

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 $\boldsymbol{\mu}_j$, Σ_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**

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

with

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

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

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

Term **A**

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

Term **B**

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

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)

Assumption 1: $\Sigma_j = \Sigma = \textit{constant}, \; j = 1, \dots, m$**.** Then

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

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

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

Assumption 2: $P_j = \frac{1}{m}, \; j = 1, \dots, m$**.** Then

$Term\ \boldsymbol{C}$

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

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

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

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

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

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

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

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

CFO clustering algorithms: Final remarks (2)

The role of q in the fuzzy clustering

Consider the minimization problem for fuzzy clustering

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

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

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

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

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

<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: $d_{is_i} = min_{j=1,\dots,m} d_{ij}, i = 1, \dots, N$

Then,

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

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

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

Thus, in this cases, 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 (3)

The role of q in the possibilistic clustering

Consider the minimization problem for fuzzy 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 **(a)** $u_{ij} \in (0,1),\ i = 1, \dots, N, j = 1, \dots, m.$

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

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

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 $\boldsymbol{\theta}_j$, they all have the same weight in the determination of $\boldsymbol{\theta}_j$, while all the other points have no influence in the determination of $\boldsymbol{\theta}_j$.

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**
$\Re_0 = \{\{\boldsymbol{x}_1\}, \dots, \{\boldsymbol{x}_N\}\}$	$\Re_0 = \{\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}\}$
. . .	. . .
$\Re_{N-1} = \{\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}\}$	$\Re_{N-1} = \{\{\boldsymbol{x}_1\}, \dots, \{\boldsymbol{x}_N\}\}$
$\Re_0 \angle \dots \angle \Re_{N-1}$	$\Re_{N-1} \angle \dots \angle \Re_0$

Agglomerative Clustering Algorithms

Let $g(C_i, C_j)$ 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:

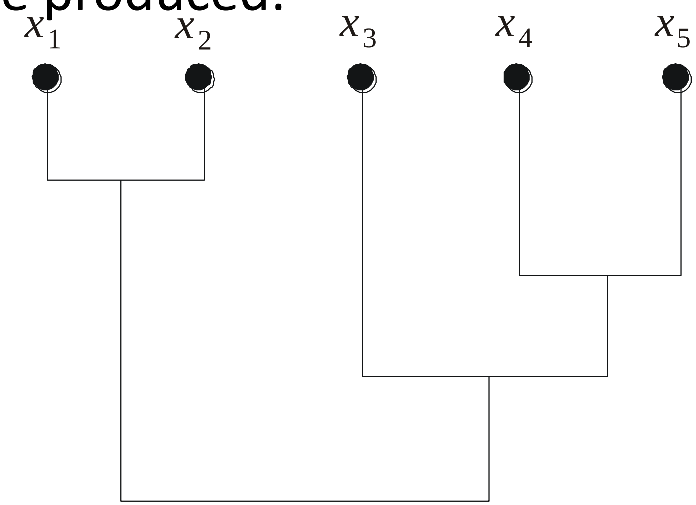
$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

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

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

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

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



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

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

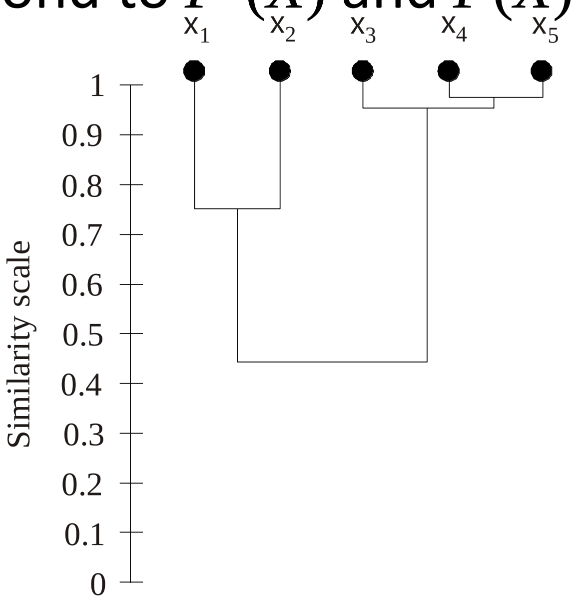
Agglomerative Clustering Algorithms

Definitions of some useful quantities:

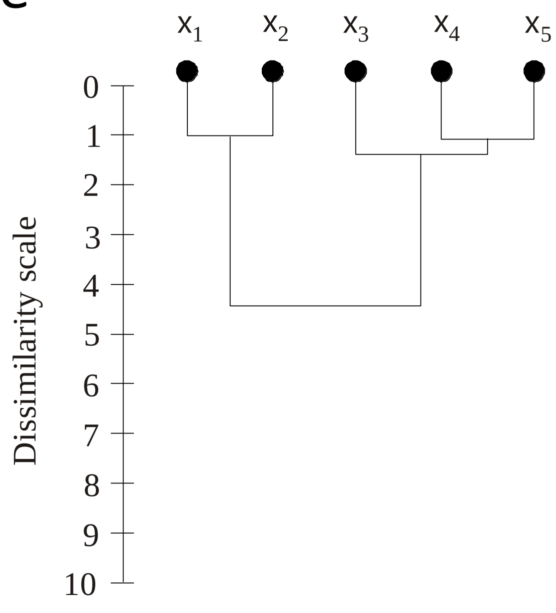
➢Proximity (dissimilarity or dissimilarity) dendrogram: A dendrogram that takes into account the level of proximity (dissimilarity or similarity) where two clusters are merged for the first time.

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)



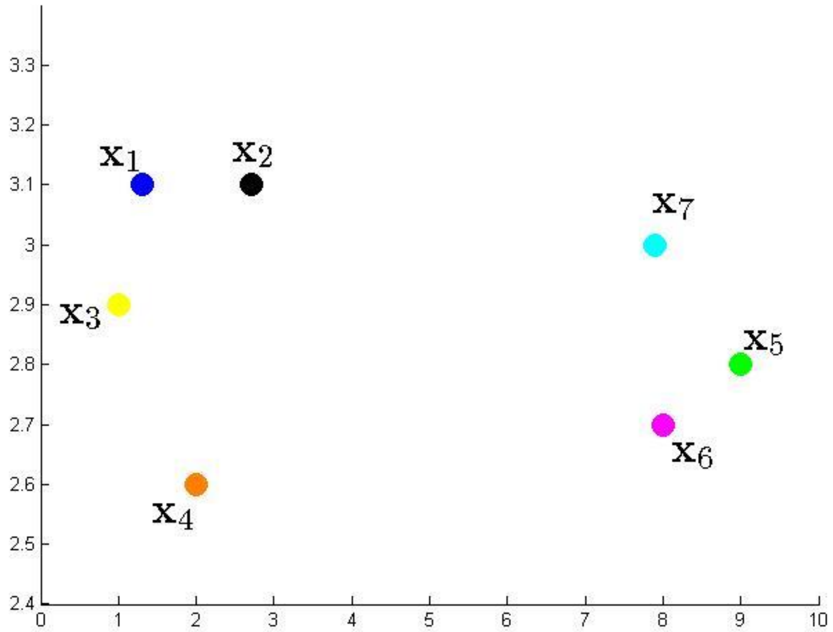
(b)

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:

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

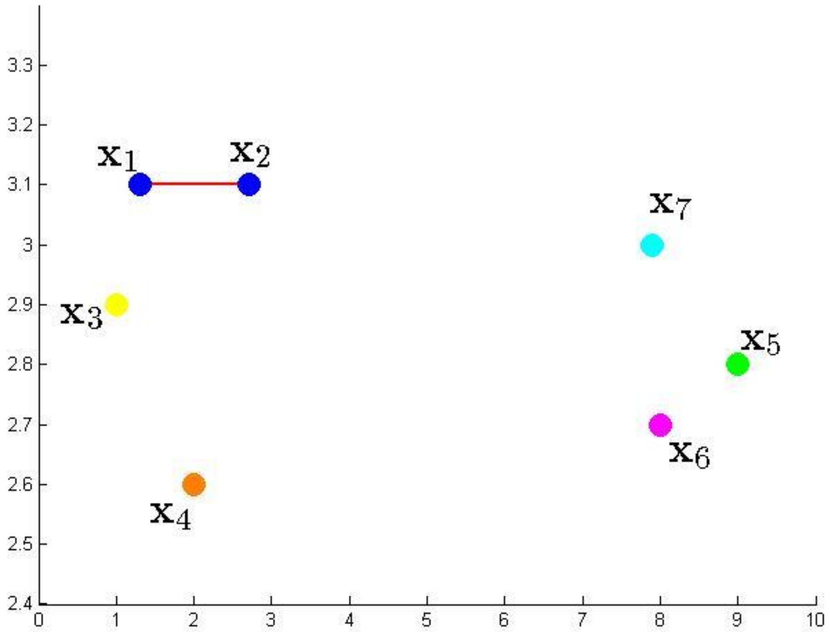


<u>Agglomerative philosophy:</u>

- 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:

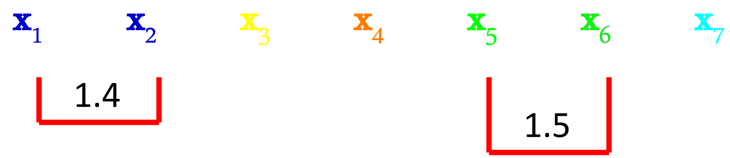


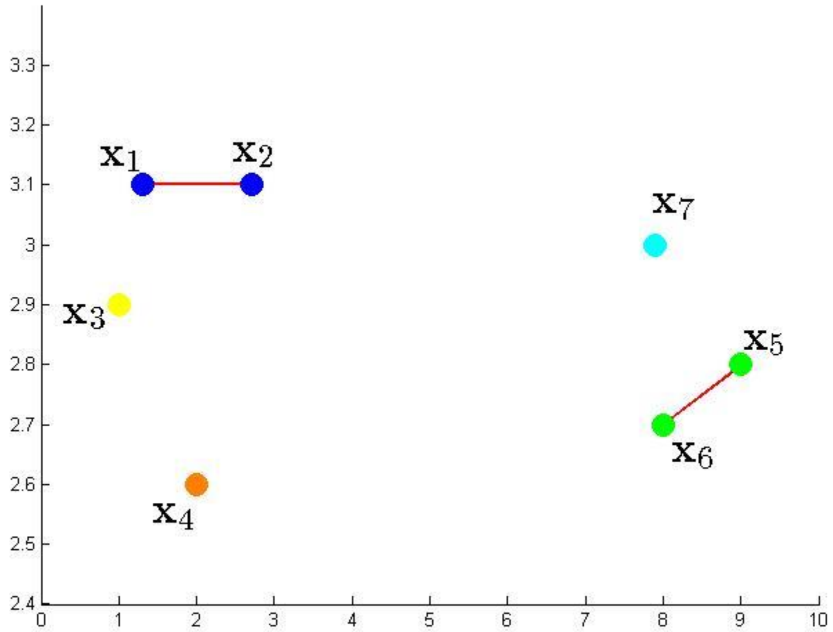
<u>Agglomerative philosophy:</u>

- 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:



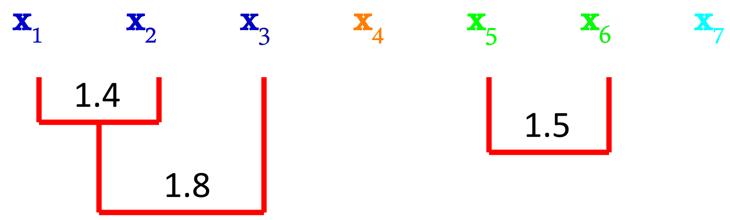


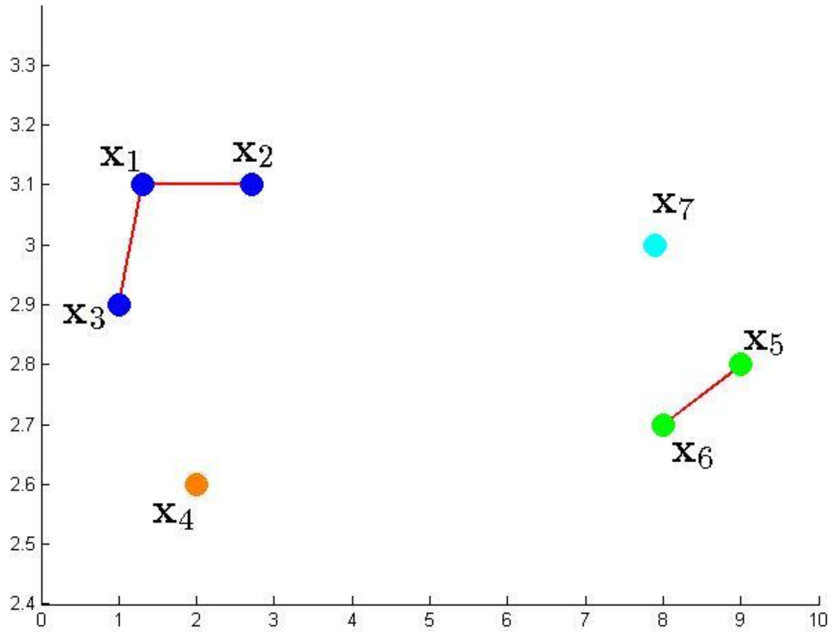
<u>Agglomerative philosophy:</u>

- 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:



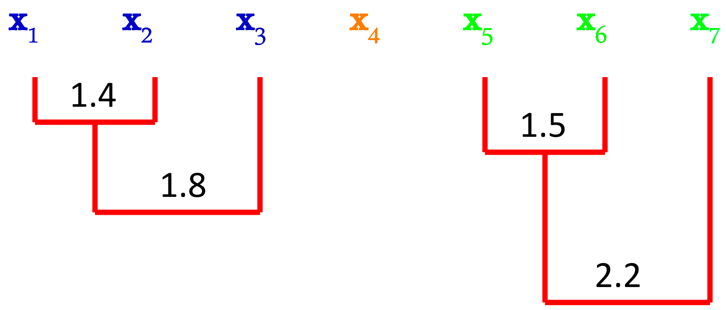


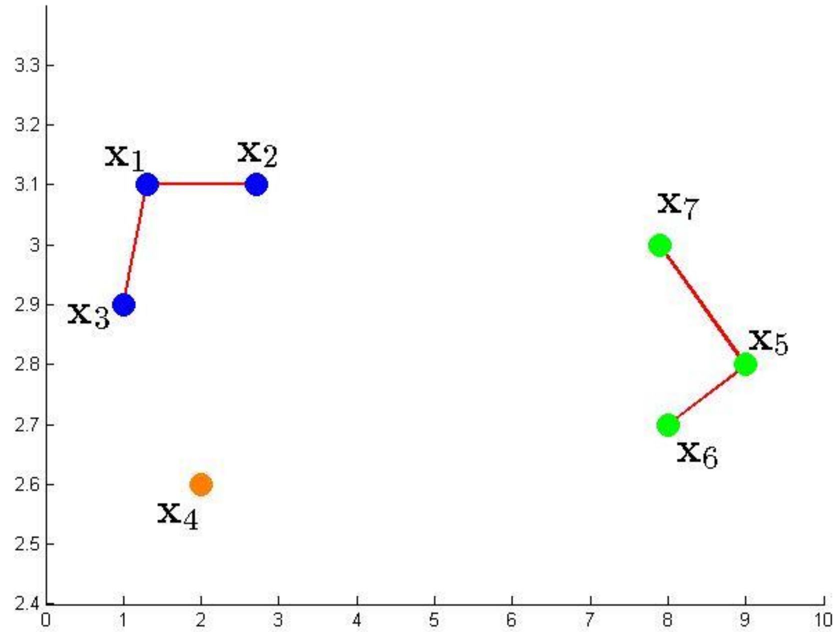
<u>Agglomerative philosophy:</u>

- 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:



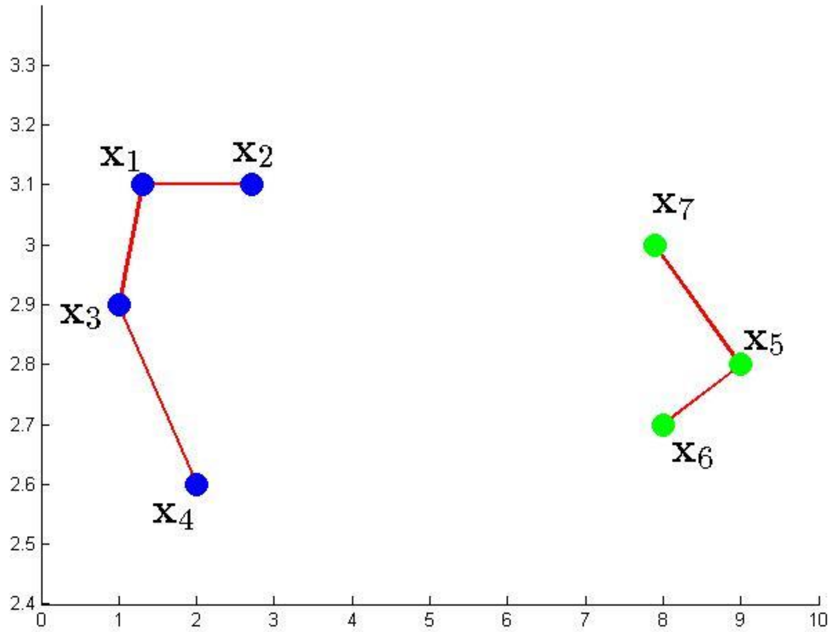


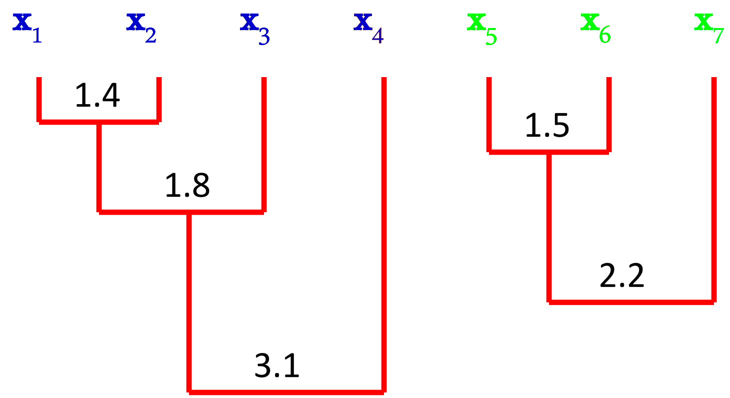
Agglomerative philosophy:

- 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:



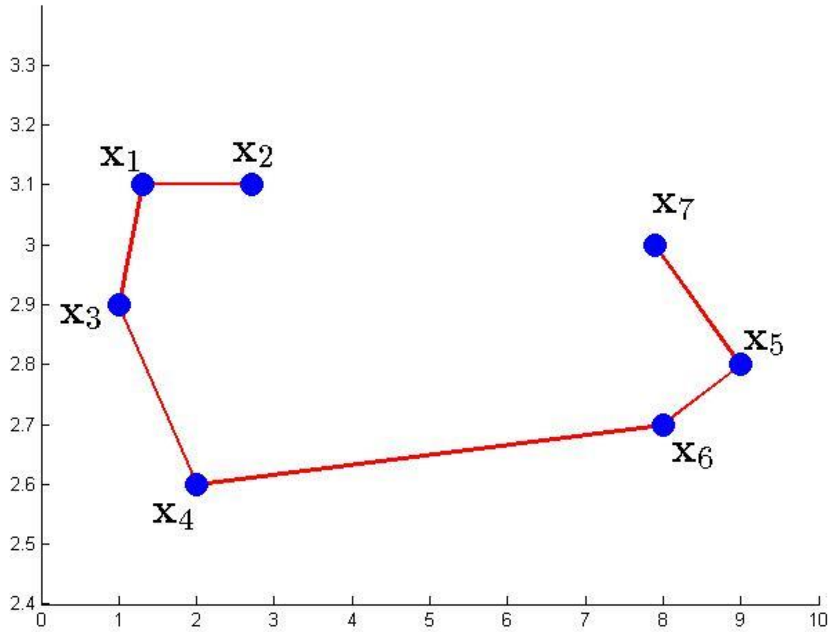


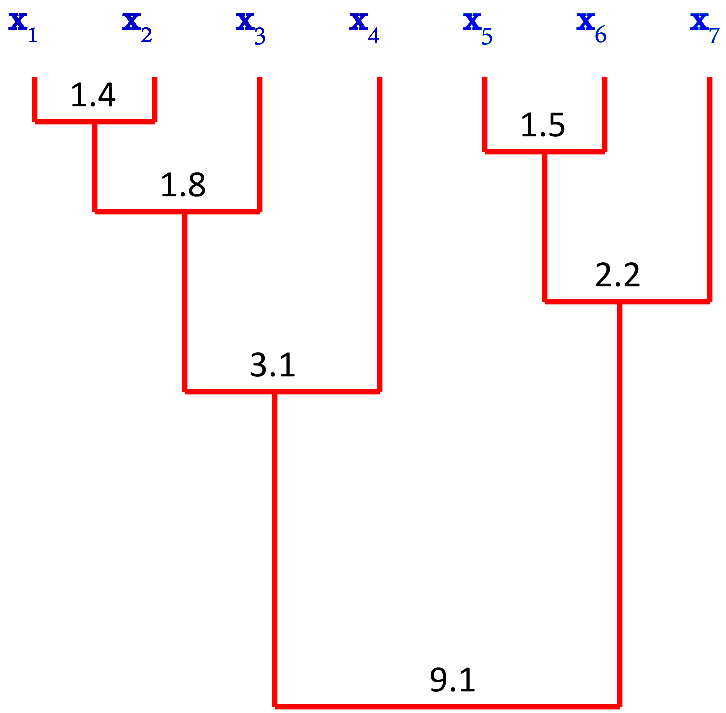
<u>Agglomerative philosophy:</u>

- 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:



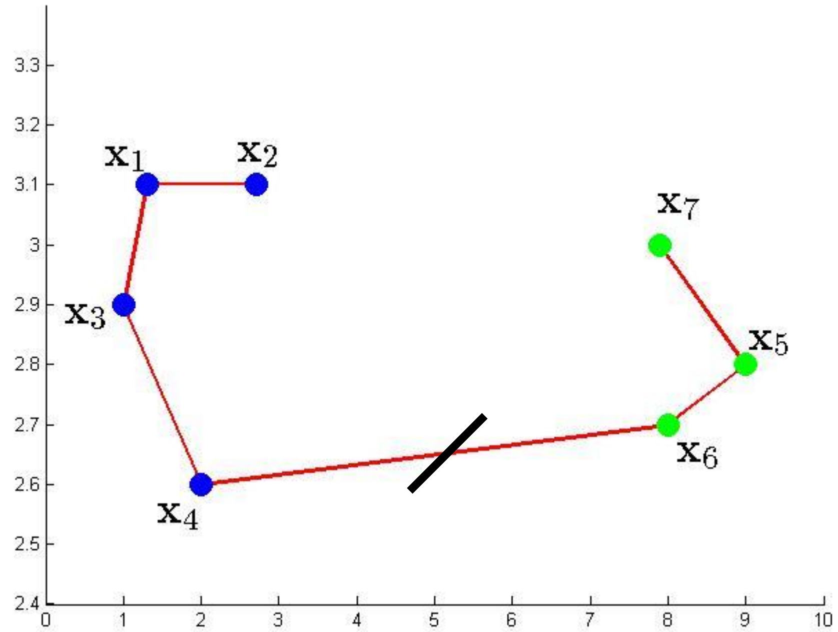


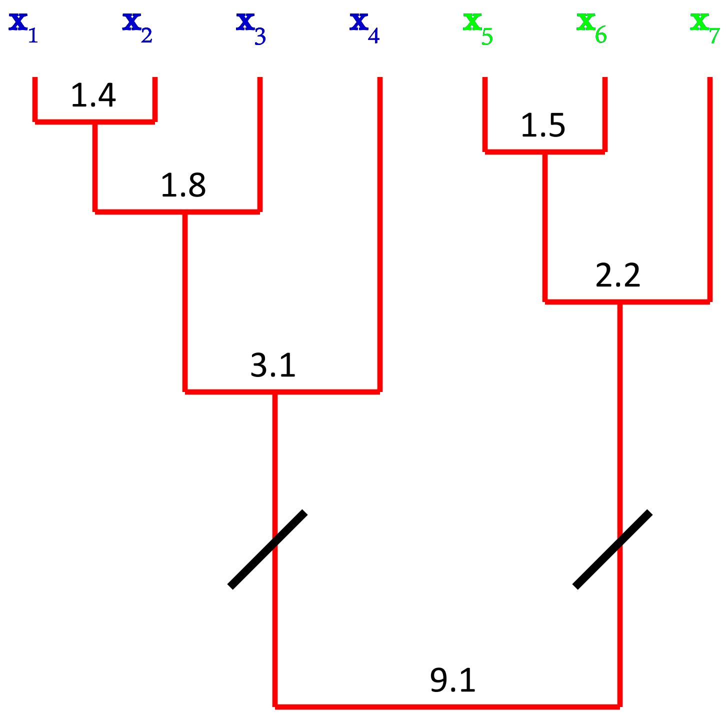
<u>Agglomerative philosophy:</u>

- 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.