# CLUSTERING ALGORITHMS VIA FUNCTION OPTIMIZATION

- In this context the clusters are assumed to be described by a parametric specific model whose parameters are unknown (all parameters are included in a vector denoted by $\underline{\theta}$).

  Examples:

  - Compact clusters. Each cluster $C_i$ is represented by a point $\underline{m}_i$ in the $l$-dimensional space. Thus $\underline{\theta}=[\underline{m}_1^{\ T}, \underline{m}_2^{\ T}, \ldots, \underline{m}_m^{\ T}]^T$.
  - Ring-shaped clusters. Each cluster $C_i$ is modeled by a hypersphere $C(\underline{c}_i, r_i)$, where $\underline{c}_i$ and $r_i$ are its center and its radius, respectively. Thus

$$\underline{\theta}=[\underline{c}_1^T, r_1, \underline{c}_2^T, r_2, \ldots, \underline{c}_m^T, r_m]^T.$$

- A cost $J(\underline{\theta})$ is defined as a function of the data vectors in $X$ and $\underline{\theta}$. Optimization of $J(\underline{\theta})$ with respect to $\underline{\theta}$ results in $\underline{\theta}$ that characterizes optimally the clusters underlying $X$.
- The number of clusters $m$ is a priori known in most of the cases.

- Cost optimization clustering algorithms considered in the sequel
    - Mixture decomposition schemes.
    - Fuzzy clustering algorithms.
    - Possibilistic clustering algorithms.

## ❖ Mixture Decomposition (MD) schemes

- Here, each vector belongs to a single cluster with a certain probability
- MD schemes rely on the Bayesian framework:

  A vector $\underline{x}_i$ is appointed to cluster $C_j$ if

  $P(C_j| \underline{x}_i)>P(C_k| \underline{x}_i)$, $\quad k=1,\ldots,m,\ k\neq j$.

However:

- No cluster labeling information is available for the data vectors
- The a priori cluster probabilities $P(C_j)\equiv P_j$ are also unknown

- A solution: Adoption of the EM algorithm
  - E-step

$$Q(\underline{\Theta};\underline{\Theta}(t))=\sum_{i=1}^{N}\sum_{j=1}^{m}P(C_j\,|\,\underline{x}_i;\underline{\Theta}(t))\ln(p(\underline{x}_i\,|\,C_j;\underline{\theta})P_j)$$

where

$\underline{\theta}=[\underline{\theta}_1^T,\ldots,\underline{\theta}_m^T]^T$ ($\underline{\theta}_j$ the parameter vector corresponding to $C_j$)

$\underline{P}=[P_1,\ldots P_m]^T$ ($P_j$ the a priori probability for $C_j$)

$\underline{\Theta}=[\underline{\theta}^T, \underline{P}^T]^T$

- M-step

$$\underline{\Theta}(t+1)=\arg\max_{\underline{\Theta}} Q(\underline{\Theta};\underline{\Theta}(t))$$

More specifically, the M-step results in:

For $\underline{\theta}_j$'s:

$$\sum_{i=1}^{N}\sum_{j=1}^{m} P(C_j \mid \underline{x}_i;\underline{\Theta}(t)) \frac{\partial}{\partial \underline{\theta}_j} \ln p(\underline{x}_i \mid C_j;\underline{\theta}_j) = 0^{(*)}$$

(*) Provided that all pairs of $(\underline{\theta}_k,\underline{\theta}_j)$ are functionally independent.

For $P_j$'s:

$$P_j = \frac{1}{N}\sum_{i=1}^{N} P(C_j \mid \underline{x}_i;\underline{\Theta}(t))^{(**)}$$

(**) Taking into account the constraints $P_k \geq 0$, $k=1,\ldots,m$ and $P_1+P_2+\ldots P_m=1$.

Thus, the EM algorithm for this case may be stated as follows:

## ❖ Generalized Mixture Decomposition Algorithmic Scheme (GMDAS)

- Choose initial estimates, $\underline{\theta}=\underline{\theta}(0)$ and $\underline{P}=\underline{P}(0)$.
- $t=0$
- Repeat
  - Compute

$$P(C_j \mid \underline{x}_i;\underline{\Theta}(t)) = \frac{p(\underline{x}_i \mid C_j;\underline{\theta}_j(t))P_j(t)}{\sum_{k=1}^{m} p(\underline{x}_i \mid C_k;\underline{\theta}_k(t))P_k(t)} \quad , i=1,\ldots,N,\ j=1,\ldots,m \qquad (1)$$

  - Set $\underline{\theta}_j(t+1)$ equal to the solution of the equation

$$\sum_{i=1}^{N}\sum_{j=1}^{m} P(C_j \mid \underline{x}_i;\underline{\Theta}(t)) \frac{\partial}{\partial \underline{\theta}_j} \ln p(\underline{x}_i \mid C_j;\underline{\theta}_j) = 0 \qquad (2)$$

  with respect to $\underline{\theta}_j$, for $j=1,\ldots,m$.
  - Set

$$P_j(t+1) = \frac{1}{N}\sum_{i=1}^{N} P(C_j \mid x_i;\Theta(t)) \quad , j=1,\ldots,m \qquad (3)$$

  - $t=t+1$
- Until convergence, with respect to $\underline{\Theta}$, is achieved.

## Remarks:

- A termination condition for GMDAS is

$$||\underline{\Theta}(t+1)-\underline{\Theta}(t)||<\varepsilon$$

  where $||.||$ is an appropriate vector norm and $\varepsilon$ a small user-defined constant.
- The above scheme is guaranteed to converge to a global or a local maximum of the loglikelihood function.
- Once the algorithm has converged, $\underline{x}_i$'s are assigned to clusters according to the Bayes rule.

## ➢ Compact and Hyperellipsoidal Clusters

In this case :

- each cluster $C_j$ is modeled by a normal distribution $N(\underline{\mu}_j, \Sigma_j)$.
- $\underline{\theta}_j$ consists of the parameters of $\underline{\mu}_j$ and the (independent) parameters of $\Sigma_j$.

It is

$$\ln p(\underline{x} \mid C_j; \underline{\theta}_j) = \ln \frac{|\Sigma_j|^{-1/2}}{(2\pi)^{l/2}} - \frac{1}{2}(\underline{x} - \underline{\mu}_j)^T \Sigma_j^{-1} (\underline{x} - \underline{\mu}_j) \ , \quad j=1, \ldots, m$$

For this case:

- Eq. (1) in GMDAS is replaced by

$$P(C_j \mid \underline{x}; \underline{\Theta}(t)) = \frac{|\Sigma_j(t)|^{-1/2} \exp(-\frac{1}{2}(\underline{x} - \underline{\mu}_j(t))^T \Sigma_j^{-1}(t)(\underline{x} - \underline{\mu}_j(t))) P_j(t)}{\sum_{k=1}^{m} |\Sigma_k(t)|^{-1/2} \exp(-\frac{1}{2}(\underline{x} - \underline{\mu}_k(t))^T \Sigma_k^{-1}(t)(\underline{x} - \underline{\mu}_k(t))) P_k(t)}$$

- Eq. (2) in GMDAS is replaced by the equations

$$\underline{\mu}_j(t+1) = \frac{\sum_{k=1}^{N} P(C_j \mid \underline{x}_k; \underline{\Theta}(t)) \underline{x}_k}{\sum_{k=1}^{N} P(C_j \mid \underline{x}_k; \underline{\Theta}(t))} \qquad \Sigma_j(t+1) = \frac{\sum_{k=1}^{N} P(C_j \mid \underline{x}_k; \underline{\Theta}(t))(\underline{x}_k - \underline{\mu}_j(t))(\underline{x}_k - \underline{\mu}_j(t))^T}{\sum_{k=1}^{N} P(C_j \mid \underline{x}_k; \underline{\Theta}(t))}$$

- Remark:
  - The above scheme is computationally very demanding since it requires the inversion of the $m$ covariance matrices at each iteration step. Two ways to deal with this problem are:
  - The use of a single covariance matrix for all clusters.
  - The use of different diagonal covariance matrices.

- Example 1: (a) Consider three two-dimensional normal distributions with mean values:

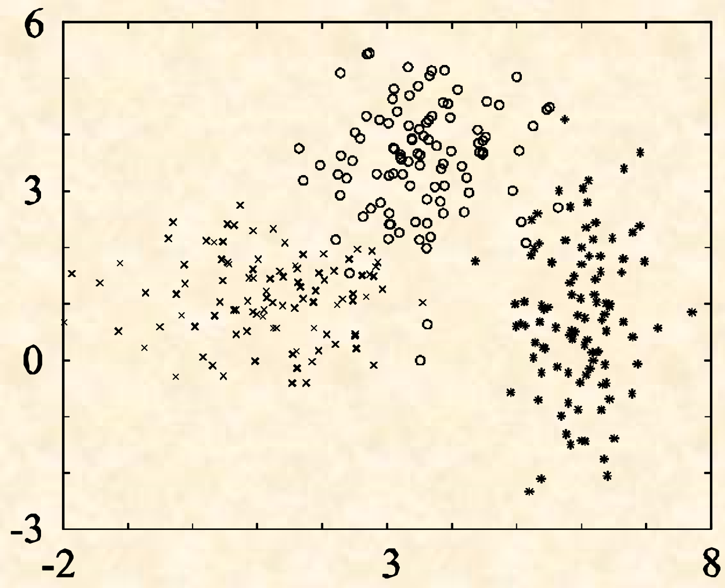
$$\underline{\mu}_1=[1, 1]^T,\ \underline{\mu}_2=[3.5, 3.5]^T,\ \underline{\mu}_3=[6, 1]^T$$

and covariance matrices

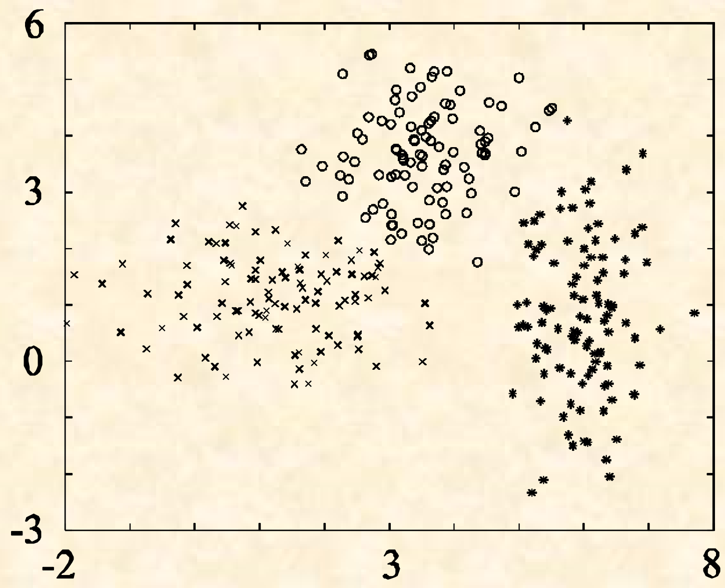
$$\Sigma_1 = \begin{bmatrix} 1 & -0.3 \\ -0.3 & 1 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}, \quad \Sigma_3 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix},$$

respectively.

A group of $100$ vectors stem from each distribution. These form the data set $X$.



(a) The data set



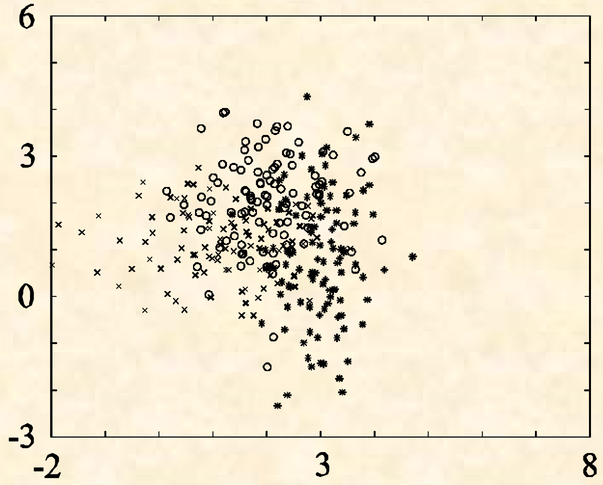
(b) Results of GMDAS

Confusion matrix:

| | *Cluster 1* | *Cluster 2* | *Cluster 3* |
|---|---|---|---|
| *1st distribution* | 99 | 0 | 1 |
| *2nd distribution* | 0 | 100 | 0 |
| *3rd distribution* | 3 | 4 | 93 |

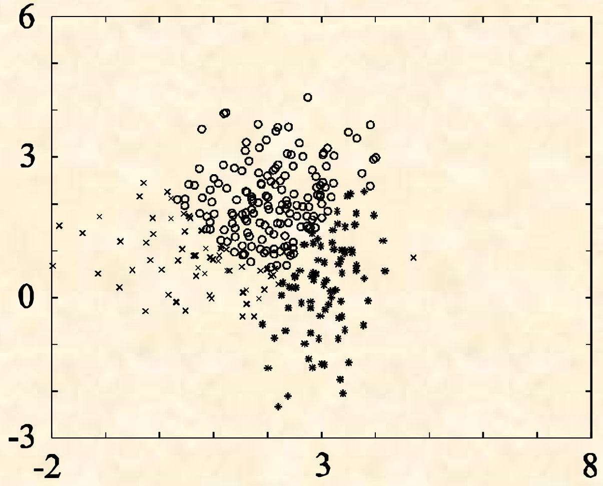
The algorithm reveals accurately the underlying structure.

(b) The same as (a) but now $\underline{\mu}_1=[1, 1]^T$, $\underline{\mu}_2=[2, 2]^T$, $\underline{\mu}_3=[3, 1]^T$ (The clusters are closer).



(a)

The data set



(b)

Results of GMDAS

Confusion matrix:

| | *Cluster 1* | *Cluster 2* | *Cluster 3* |
|---|---|---|---|
| *1st distribution* | 85 | 4 | 11 |
| *2nd distribution* | 35 | 56 | 9 |
| *3rd distribution* | 26 | 0 | 74 |

The algorithm reveals the underlying structure less accurately.

## ❖ Fuzzy clustering algorithms

- Each vector belongs simultaneously to more than one clusters.

- A fuzzy $m$-clustering of $X$, is defined by a set of functions

$$u_j: X \rightarrow A \equiv [0, 1], \;\; j=1, \ldots, m.$$

If $A=\{0,1\}$, a hard $m$-clustering of $X$ is produced.

- $u_j(\underline{x}_i)$ denotes the degree of membership of $\underline{x}_i$ in cluster $C_j$. It is

$$u_1(\underline{x}_i) + u_2(\underline{x}_i) + \ldots + u_m(\underline{x}_i) = 1$$

- The number of clusters $m$ is assumed to be known *a priori*.

## ❖ Fuzzy clustering algorithms (cont)

➢ Cost function definition

Let

- $\underline{\theta}_j$ be the representative vector of $C_j$.
- $\underline{\theta} \equiv [\underline{\theta}_1^T, \dots, \underline{\theta}_m^T]^T$.
- $U \equiv [u_{ij}] = [u_j(\underline{x}_i)]$
- $d(\underline{x}_i, \underline{\theta}_j)$ be the dissimilarity between $\underline{x}_i$ and $\underline{\theta}_j$
- $q(>1)$ a parameter called fuzzifier.

## ❖ Fuzzy clustering algorithms (cont)

➢ Most fuzzy clustering schemes result from the minimization of :

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

Subject to the constraints: $\sum_{j=1}^{m} u_{ij} = 1, \quad i = 1, \ldots, N$

where

$$u_{ij} \in [0,1], \quad i=1,\ldots,N, \quad j=1,\ldots,m$$

and

$$0 < \sum_{i=1}^{N} u_{ij} < N, \quad j = 1,2,\ldots,m$$

- Remarks:
  - The degree of membership of $\underline{x}_i$ in $C_j$ cluster is related to the grade of membership of $\underline{x}_i$ in rest $m$-1 clusters.
  - If $q$=1, no fuzzy clustering is better than the best hard clustering in terms of $J_q(\underline{\theta},U)$.
  - If $q$>1, there are fuzzy clusterings with lower values of $J_q(\underline{\theta},U)$ than the best hard clustering.

## ❖ Fuzzy clustering algorithms (cont)

➢ Minimizing $J_q(\underline{\theta},U)$:

Minimization $J_q(\underline{\theta},U)$ with respect to $U$, subject to the constraints leads to the following Lagrangian function,

$$J_{Lan}(\underline{\theta},U)=\sum_{i=1}^{N}\sum_{j=1}^{m}u_{ij}^{q}d(\underline{x}_i,\underline{\theta}_j)-\sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{m}u_{ij}-1\right)$$

Minimizing $J_{Lan}(\underline{\theta},U)$ with respect to $u_{rs}$, we obtain

$$u_{rs}=\frac{1}{\sum_{j=1}^{m}\left(\frac{d(\underline{x}_r,\underline{\theta}_s)}{d(\underline{x}_r,\underline{\theta}_j)}\right)^{\frac{1}{q-1}}},\qquad r=1,\ldots,N,\qquad s=1,\ldots,m.$$

Setting the gradient of $J(\underline{\theta},U)$, with respect to $\underline{\theta}$, equal to zero we obtain,

$$\frac{\partial J(\underline{\theta},U)}{\partial\underline{\theta}_j}=\sum_{i=1}^{N}u_{ij}^{q}\frac{\partial d(\underline{x}_i,\underline{\theta}_j)}{\partial\underline{\theta}_j}=0,\qquad j=1,\ldots,m.$$

The last two equations are coupled. Thus, no closed form solutions are expected. Therefore, minimization is carried out iteratively.

## Generalized Fuzzy Algorithmic Scheme (GFAS)

```
• Choose $\underline{\theta}_j(0)$ as initial estimate for $\underline{\theta}_j$, $j=1,\dots,m$.
• $t=0$
• Repeat
    – For $i=1$ to $N$
        o For $j=1$ to $m$
            $u_{ij}(t) = 1 \Big/ \sum_{j=1}^{m} \left( \frac{d(\underline{x}_i, \underline{\theta}_j(t))}{d(\underline{x}_i, \underline{\theta}_k(t))} \right)^{\frac{1}{q-1}}$    (A)
        o End {For-$j$}
    – End {For-$i$}
    – $t=t+1$
    – For $j=1$ to $m$
        o *Parameter updating:* Solve
            $\sum_{i=1}^{N} u_{ij}^{q}(t-1) \frac{\partial d(\underline{x}_i, \underline{\theta}_j)}{\partial \underline{\theta}_j} = 0.$    (B)
          with respect to $\underline{\theta}_j$ and set $\underline{\theta}_j(t)$ equal to this solution.
    – End {For-$j$}
• Until a termination criterion is met
```

## Remarks:

- A candidate termination condition is

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

  where $||.||$ is any vector norm and $\varepsilon$ a user-defined constant.
- GFAS may also be initialized from $U(0)$ instead of $\underline{\theta}_j(0)$, $j=1,\ldots,m$ and start iterations with computing $\underline{\theta}_j$ first.
- If a point $\underline{x}_i$ coincides with one or more representatives, then it is shared arbitrarily among the clusters whose representatives coincide with $\underline{x}_i$, subject to the constraint that the summation of the degree of membership over all clusters sums to $1$.

## ❖ Fuzzy Clustering – Point Representatives

- Point representatives are used in the case of compact clusters
- Each $\underline{\theta}_j$ consists of $l$ parameters
- Every dissimilarity measure $d(\underline{x}_i,\underline{\theta}_j)$ between two points can be used
- Common choices for $d(\underline{x}_i,\underline{\theta}_j)$ are
  - $d(\underline{x}_i,\underline{\theta}_j) = (\underline{x}_i - \underline{\theta}_j)^T A(\underline{x}_i - \underline{\theta}_j)$,

  where $A$ is symmetric and positive definite matrix.

  In this case:

$$\partial d(\underline{x}_i,\underline{\theta}_j) / \partial \underline{\theta}_j = 2A(\underline{\theta}_j - \underline{x}_i).$$

Thus the updating equation (B) in GFAS becomes

$$\underline{\theta}_j(t) = \sum_{i=1}^{N} u_{ij}^q(t-1)\underline{x}_i \Big/ \sum_{i=1}^{N} u_{ij}^q(t-1)$$

- GFAS with the above distance is also known as Fuzzy c-Means (FCM) or Fuzzy k-Means algorithm.
- FCM converges to a stationary point of the cost function or it has at least one subsequence that converges to a stationary point. This point may be a local (or global) minimum or a saddle point.

## ❖ Fuzzy clustering – Point representatives (cont.)

- The *Minkowski* distance

$$d(\underline{x}_i,\underline{\theta}_j)=\left(\sum_{k=1}^{l}|x_{ik}-\theta_{jk}|^p\right)^{1/p}$$

where $p$ is a positive integer and $x_{ik}$, $\theta_{jk}$ are the $k$-th coordinates of $\underline{x}_i$ and $\underline{\theta}_j$.

For even and finite $p$, the differentiability of $d(\underline{x}_i,\underline{\theta}_j)$ is guaranteed. In this case the updating equation (B) of GFAS gives

$$\sum_{i=1}^{N}u_{ij}^{q}(t-1)\frac{(\theta_{jr}-x_{ir})^{p-1}}{\left(\sum_{k=1}^{l}|x_{ik}-\theta_{jk}|^{p}\right)^{1-\frac{1}{p}}}=0,\qquad r=1,\ldots,l$$

a system of $l$ nonlinear equations with $l$ unknowns.

GFAS algorithms with the Minkowski distance are also known as *pFCM* algorithms.

## ❖ Fuzzy Clustering – Point representatives (cont.)

➢ Example 2(a):

- Consider the setup of example 1(a).
- Consider GFAS with distances

  (i) $d(\underline{x}_i,\underline{\theta}_j) = (\underline{x}_i - \underline{\theta}_j)^T A(\underline{x}_i - \underline{\theta}_j)$, with $A$ being the identity matrix

  (ii) $d(\underline{x}_i,\underline{\theta}_j) = (\underline{x}_i - \underline{\theta}_j)^T A(\underline{x}_i - \underline{\theta}_j)$, with $A = \begin{bmatrix} 2 & 1.5 \\ 1.5 & 2 \end{bmatrix}$

  (iii) The Minkowski distance with $p=4$.

➢ Example 2(b):

- Consider the setup of example 1(b).
- Consider GFAS with the distances considered in example 2(a).

The corresponding confusion matrices for example 2(a) and 2(b) are (Here a vector is assigned to the cluster for which $u_{ij}$ has the maximum value.)

For the example 2(a)

$$A_i = \begin{bmatrix} 98 & 2 & 0 \\ 14 & 84 & 2 \\ 11 & 0 & 89 \end{bmatrix} \quad A_{ii} = \begin{bmatrix} 63 & 11 & 26 \\ 5 & 95 & 0 \\ 39 & 23 & 38 \end{bmatrix} \quad A_{iii} = \begin{bmatrix} 96 & 0 & 4 \\ 11 & 89 & 0 \\ 13 & 2 & 85 \end{bmatrix}$$

For the example 2(b)

$$A'_i = \begin{bmatrix} 51 & 46 & 3 \\ 14 & 47 & 39 \\ 43 & 0 & 57 \end{bmatrix} \quad A'_{ii} = \begin{bmatrix} 79 & 21 & 0 \\ 19 & 58 & 23 \\ 28 & 41 & 31 \end{bmatrix} \quad A'_{iii} = \begin{bmatrix} 51 & 3 & 46 \\ 37 & 62 & 1 \\ 11 & 36 & 53 \end{bmatrix}$$

➢ Remarks:

- In $A_i$ and $A_{iii}$ (example 2(a)) almost all vectors from the same distribution are assigned to the same cluster.
- The closer the clusters are, the worse the performance of all the algorithms.
- The choice of matrix $A$ in $d(\underline{x}_i,\underline{\theta}_j) = (\underline{x}_i - \underline{\theta}_j)^T A(\underline{x}_i - \underline{\theta}_j)$ plays an important role to the performance of the algorithm.

## ❖ Fuzzy Clustering – Quadric surfaces as representatives

Here the representatives are quadric surfaces (hyperellipsoids, hyperparaboloids, etc.)

- General form of an equation describing a quadric surface $Q$:

1. $\underline{x}^T A \underline{x} + \underline{b}^T \underline{x} + c = 0$,

   where $A$ is an $l$x$l$ symmetric matrix, $\underline{b}$ is an $l$x1 vector, $c$ is a scalar and $\underline{x}=[x_1, \ldots, x_l]^T$.

   For various choices of these quantities we obtain hyperellipses, hyperparabolas and so on.

2. $\underline{q}^T \underline{p}=0$,

   where

   $\underline{q}=[x_1^2, x_2^2, \ldots, x_l^2, x_1 x_2, \ldots, x_{l-1} x_l, x_1, x_2, \ldots, x_l, 1]^T$

   and

   $\underline{p}=[p_1, p_2, \ldots, p_l, p_{l+1}, \ldots, p_r, p_{r+1}, \ldots, p_s]^T$

   with $r=l(l+1)/2$ and $s=r+l+1$.

NOTE: The above representations of $Q$ are equivalent.

## ❖ Fuzzy Clustering – Quadric surfaces as representatives (cont)

First concern: "*Definition of the distance of a point* $\underline{x}$ *to a quadric surface* $Q$"

➢ Types of distances

- (Squared) Algebraic distance:

$$d_a^2(\underline{x},Q)=(\underline{x}^T A\underline{x} + \underline{b}^T\underline{x} + c)^2 \equiv \underline{p}^T M\underline{p},$$

  where $M=\underline{q}\underline{q}^T$.

- Perpendicular distance:

$$d_p^2(\underline{x},Q)=min_z\|\underline{x}-\underline{z}\|^2$$

  subject to the constraint

$$\underline{z}^T A\underline{z} + \underline{b}^T\underline{z} + c=0$$

In words, $d_p^2(\underline{x},Q)$ is the distance between $\underline{x}$ and the closest to $\underline{x}$ point that lies in $Q$.

## ❖ Fuzzy Clustering – Quadric surfaces as representatives (cont)

- Radial distance (only when $Q$ is a hyperellipsoidal):
  For $Q$ hyperellipsoidal, the representative equation can become

$$(\underline{x}-\underline{c})^T A(\underline{x}-\underline{c})=1$$

  where $\underline{c}$ is the center of the ellipse and $A$ a positive definite symmetric matrix defining major axis, minor axis and orientation.
  Then the following is true

$$d_r^2(\underline{x},Q)=\|\underline{x}-\underline{z}\|^2,$$

  subject to the constraints

$$(\underline{z}-\underline{c})^T A(\underline{z}-\underline{c})=1$$

  and

$$(\underline{z}-\underline{c})=a(\underline{x}-\underline{c})$$

  In words,
  – the intersection point $\underline{z}$ between the line segment $\underline{x}$-$\underline{c}$ and $Q$ is determined
  – the squared Euclidean distance between $\underline{x}$ and $\underline{z}$ is computed.

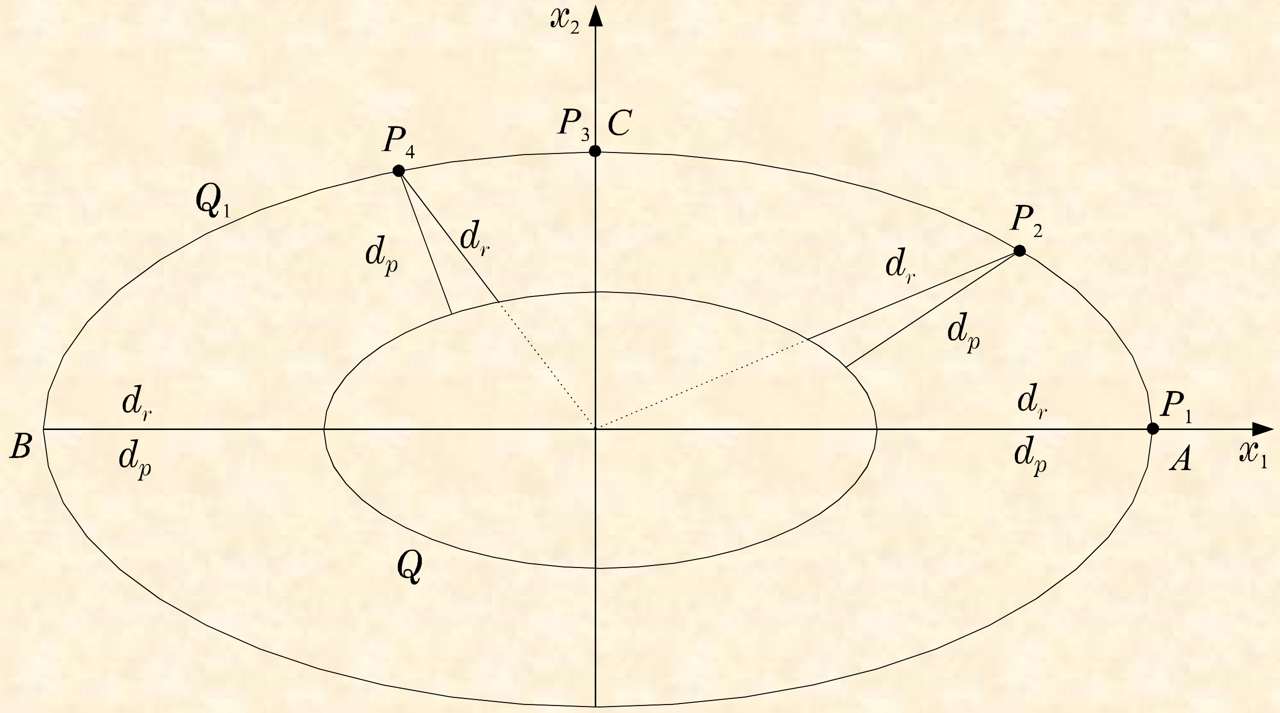
## ❖ Fuzzy Clustering – Quadric surfaces as representatives (cont)

- (Squared) Normalized radial distance (only when $Q$ is a hyperellipsoidal):

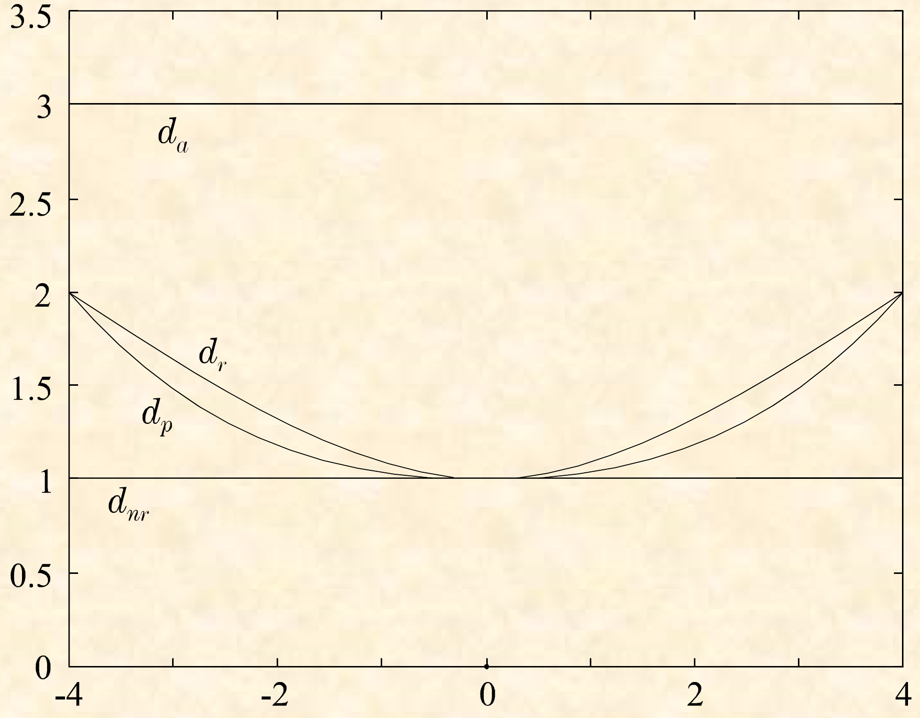
$$d_{nr}^2(\underline{x},Q)=(((\underline{x}-\underline{c})^T A(\underline{x}-\underline{c}))^{1/2}-1)^2$$

➢ Example 3:

- Consider two ellipses $Q$ and $Q_1$, centered at $\underline{c}=[0, 0]^T$, with $A=diag(0.25, 1)$ and $A_1=diag(1/16, ¼)$, respectively.
- Let $P(x_1,x_2)$ be a point in $Q_1$ moving from $A(4,0)$ to $B(-4,0)$, with $x_2>0$.



## ❖ Fuzzy Clustering – Quadric surfaces as representatives (cont)



- $d_a$ and $d_{nr}$ do not vary as $P$ moves.
- $d_r$ can be used as an approximation of $d_p$, when $Q$ is a hyperellipsoid.

## ❖ Fuzzy Clustering – Quadric surfaces as representatives (cont)

### *Fuzzy Shell Clustering Algorithms*

➢ The Adaptive Fuzzy C-Shells (AFCS) algorithm.

- It recovers hyperellipsoidal clusters.
- It is the result of the minimization of the cost function

$$J_{nr}(\underline{\theta},U)=\sum_{i=1}^{N}\sum_{j=1}^{m}u_{ij}^{q}d_{nr}^{2}(\underline{x}_i,Q_j)$$

with respect to $u_{ij}$'s, $\underline{c}_j$'s, $A_j$'s, $j=1,\ldots,m$.

- AFCS stems from GFAS, with the "parameter updating" being as follows:
  - Parameter updating:
    - o Solve with respect to $\underline{c}_j$ and $A_j$ the following equations

$$\sum_{i=1}^{N}u_{ij}^{q}(t-1)\frac{d_{nr}(\underline{x}_i,\underline{\theta}_j)}{\phi(\underline{x}_i,\underline{\theta}_j)}(\underline{x}_i-\underline{c}_j)=0,$$

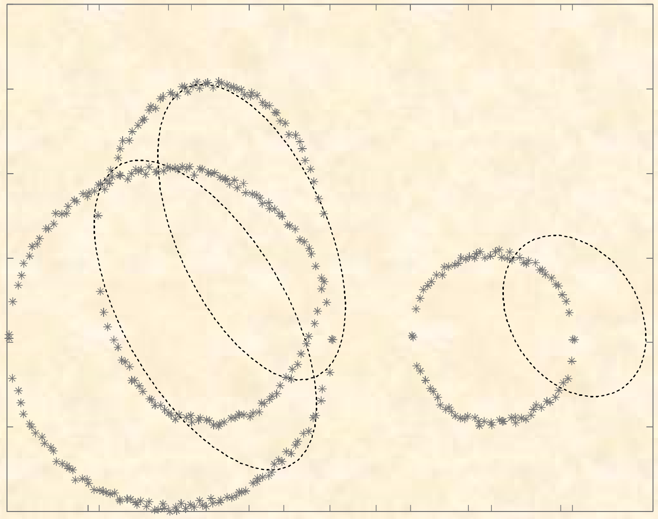
$$\sum_{i=1}^{N}u_{ij}^{q}(t-1)\frac{d_{nr}(\underline{x}_i,\underline{\theta}_j)}{\phi(\underline{x}_i,\underline{\theta}_j)}(\underline{x}_i-\underline{c}_j)(\underline{x}_i-\underline{c}_j)^{T}=O$$

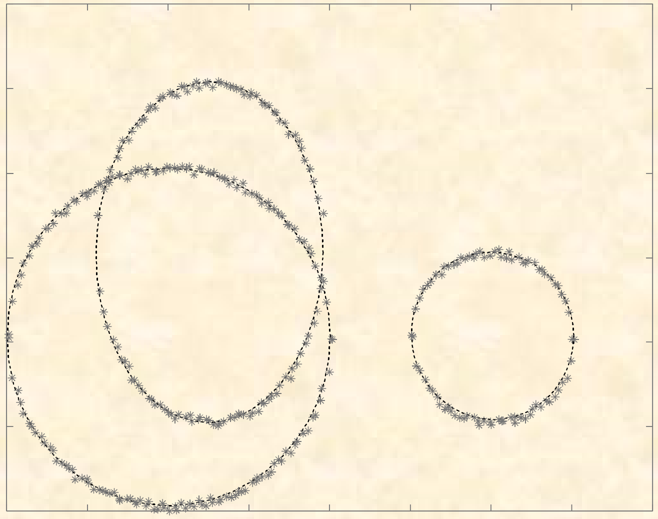
where:

$$\phi^2(\underline{x}_i,\underline{\theta}_j)=(\underline{x}_i-\underline{c}_j)^T A_j(\underline{x}_i-\underline{c}_j),$$
$$d_{nr}^2(\underline{x}_i,\underline{\theta}_j)=(\phi(\underline{x}_i,\underline{\theta}_j)-1)^2$$

o Set $\underline{c}_j(t)$ and $A_j(t)$, $j=1,\ldots,m,$ equal to the resulting solution

- Example 4: Thick dots represent the points of the data set. Thin dots represent (a) the initial estimates and (b) the final estimates of the ellipses


(a)


(b)

## ❖ Fuzzy Clustering – Quadric surfaces as representatives (cont)

➢ The Fuzzy C Ellipsoidal Shells (FCES) Algorithm

- It recovers hyperellipsoidal clusters.
- It is the result of the minimization of the cost function

$$J_r(\underline{\theta}, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}^q d_r^2(\underline{x}_i, Q_j)$$

- FCES stems from GFAS. Setting the derivative of $J_r(\underline{\theta},U)$ with respect to $\underline{c}_j$'s and $A_j$'s equal to zero, the "parameter updating" part of the algorithm follows.

➢ The Fuzzy C Quadric Shells (FCQS) Algorithm

- It recovers general hyperquadric shapes.
- It is the result of the minimization of the cost function

$$J_a(\underline{\theta}, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}^q d_a^2(\underline{x}_i, Q_j) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}^q \underline{p}_j^T M_i \underline{p}_j, \quad (\underline{p}_j \equiv \underline{\theta}_j)$$

subject to constraints such as:

## ❖ Fuzzy Clustering – Quadric surfaces as representatives (cont)

(i) $\|p_j\|^2=1$, (ii) $\sum_{k=1}^{r+l} p_{jk}^2 = 1$

(iii) $p_{j1}=1$, (iv) $p_{js}{}^2=1$, (v) $\| \sum_{k=1}^{l} p_{jk}^2 + 0.5 \sum_{k=l+1}^{r} p_{jk}^2 \|^2 = 1$

- FCQS stems from GFAS. Setting the derivative of $J_r(\underline{\theta},U)$ with respect to $\underline{p}_j$'s equal to zero and taking into account the constraints, the "parameter updating" part of the algorithm follows.

➢ The Modified Fuzzy C Quadric Shells (MFCQS) Algorithm

- It recovers hyperquadric shapes.
- It results from the GFAS scheme where
  - The grade of membership of a vector $\underline{x}_i$ in a cluster is determined using the perpendicular distance.
  - The updating of the parameters of the representatives is carried out using the parameter updating part of FCQS (where the algebraic distance is used).

## ❖ Fuzzy Clustering – Hyperplanes as representatives

Algorithms that recover hyperplanar clusters.

➢ Fuzzy c-varieties (FCV) algorithm

- It is based on the minimization of the distances of the vectors in $X$ from hyperplanes.
- *Disadvantage:* It tends to recover very long clusters and, thus, collinear distinct clusters may be merged to a single one.

➢ Gustafson-Kessel (GK) algorithm

- Each planar cluster is represented by a center $\underline{c}_j$ and a covariance matrix $\Sigma_j$, i.e., $\underline{\theta}_j=(\underline{c}_j, \Sigma_j)$.
- The distance between a point $\underline{x}$ and the $j$-th cluster is defined as

$$d_{GK}^2(\underline{x},\underline{\theta}_j)=|\Sigma_j|^{1/l}(\underline{x}-\underline{c}_j)^T\Sigma_j^{-1}(\underline{x}-\underline{c}_j)$$

- The GK algorithm is derived via the minimization of the cost function

$$J_{GK}(\underline{\theta},U)=\sum_{i=1}^{N}\sum_{j=1}^{m}u_{ij}^q d_{GK}^2(\underline{x}_i,\underline{\theta}_j)$$

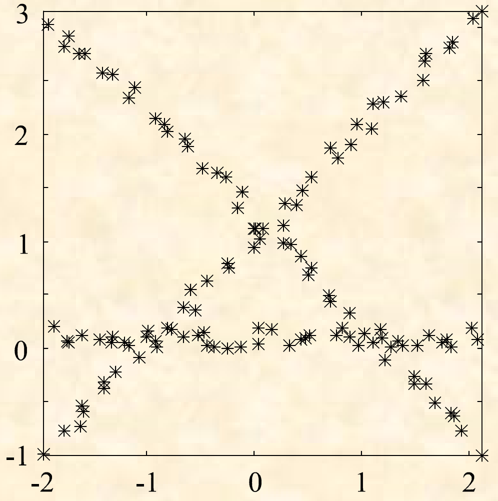
## ❖ Fuzzy Clustering – Hyperplanes as representatives (cont)

- The GK algorithm stems from GFAS. Setting the derivative of $J_{GK}(\underline{\theta},U)$ with respect to $\underline{c}_j$'s and $A_j$'s equal to zero, the "parameter updating" part of the algorithm becomes:

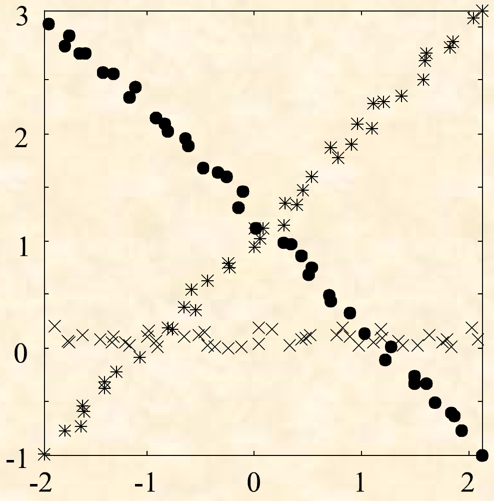
  – $$\underline{c}_j(t)=\frac{\sum_{i=1}^{N}u_{ij}^q(t-1)\underline{x}_i}{\sum_{i=1}^{N}u_{ij}^q(t-1)}$$

  – $$\Sigma_j(t)=\frac{\sum_{i=1}^{N}u_{ij}^q(t-1)(\underline{x}_i-\underline{c}_j(t))(\underline{x}_i-\underline{c}_j(t))^T}{\sum_{i=1}^{N}u_{ij}^q(t-1)}$$

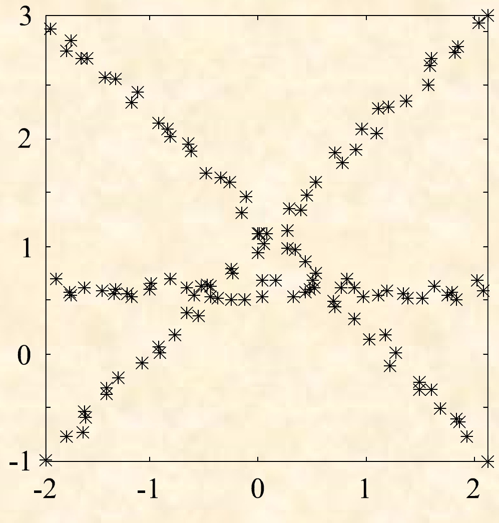
➢ Example 5:

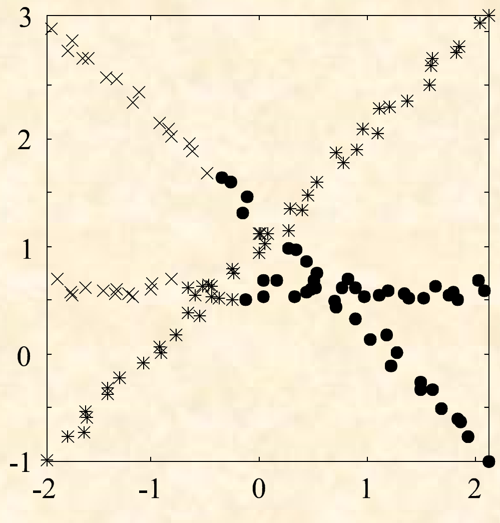


(a)



(b)


(a)


(b)

- In the first case, the clusters are well separated and the GK-algorithm recovers them correctly.
- In the second case, the clusters are not well separated and the GK-algorithm fails to recover them correctly.

## ❖ Possibilistic Clustering

- Unlike fuzzy clustering, the constraints on $u_{ij}$'s are
  - $u_{ij} \in [0, 1]$
  - $max_{j=1,\dots,m}\, u_{ij} > 0, \quad i=1,\dots,N$
  - $0 < \sum_{i=1}^{N} u_{ij} \leq N, \quad i = 1,\dots, N$

- Possibilistic clustering algorithms result from the optimization of cost functions like

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

where $\eta_j$ are suitably chosen positive constants (see below). The 2nd term is inserted in order to avoid the trivial zero solution for the $u_{ij}$' s.

(other choices for the second term of the cost function are also possible (see below)).

- Setting $\partial J(\underline{\theta},U)/\ \partial u_{ij}=0$ we obtain: $$u_{ij} = 1 \Bigg/ \left(1+\left(\frac{d(\underline{x}_i,\underline{\theta}_j)}{\eta_j}\right)^{\frac{1}{q-1}}\right)$$

## ❖ Possibilistic clustering (cont)

➢ *Generalized Possibilistic Algorithmic Scheme (GPAS)*

```
• Fix $\eta_j$, $j=1,\dots,m$.
• Choose $\underline{\theta}_j(0)$ as the initial estimates of $\underline{\theta}_j$, $j=1,\dots,m$.
• $t=0$
• Repeat
    – For $i=1$ to $N$
        o For $j=1$ to $m$
            $u_{ij}(t) = 1 \Big/ \left( 1 + \left( \frac{d(\underline{x}_i, \underline{\theta}_j(t))}{\eta_j} \right)^{\frac{1}{q-1}} \right)$
        o End {For-$j$}
    – End {For-$i$}
    – $t=t+1$
```

## ❖ Possibilistic clustering (cont)

➢ *Generalized Possibilistic Algorithmic Scheme (GPAS) (cont)*

- – For $j$=1 to $m$
  - o *Parameter updating:* Solve

$$\sum_{i=1}^{N} u_{ij}^{q}(t-1)\frac{\partial d(\underline{x}_i,\underline{\theta}_j)}{\partial \underline{\theta}_j}=0$$

with respect to $\underline{\theta}_j$ and set $\underline{\theta}_j(t)$ equal to the computed solution

- – End {For $j$}

- Until a termination criterion is met

➢ Remarks:

- $||\underline{\theta}(t)-\underline{\theta}(t-1)||<\varepsilon$ may be employed as a termination condition.
- Based on GPAS, a possibilistic algorithm can be derived, for each fuzzy clustering algorithm derived previously.

## ❖ Possibilistic clustering (cont)

➢ *Two observations*

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

  $$J(\underline{\theta},U)=\sum_{j=1}^{m}J_j$$

  where

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

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

- About $\eta_j$'s:
  - They determine the relative significance of the two terms in $J(\underline{\theta},U)$.
  - They are related to the size and the "shape" of the $C_j$'s, $j=1,\ldots,m$.
  - They may be determined as follows:
    - o Run the GFAS algorithm and after its convergence estimate $\eta_j$'s as

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

    - o Run the GPAS algorithm

## Possibilistic clustering (cont)

- Remark:

  High values of $q$:

  - In possibilistic clustering imply almost equal contributions of all vectors to all clusters
  - In fuzzy clustering imply increased sharing of the vectors among all clusters.

- *The mode-seeking property*
  - Unlike GMDAS and GFAS which are partition algorithms (they terminate with the predetermined number of clusters no matter how many 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 captured.

## ❖ Hard Clustering Algorithms

Each vector belongs exclusively to a single cluster. This implies that:

- $u_{ij} \in \{0, 1\}, \quad j=1,\dots,m$
- $\sum_{j=1}^{m} u_{ij} = 1$

That is, it can be seen as an extreme special case of the fuzzy algorithmic schemes.

However, now, the cost function

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

is not differentiable with respect to $\underline{\theta}_j$.

Despite that, the two-step optimization procedure (with respect to $u_{ij}$'s and with respect to $\underline{\theta}_j$'s) adopted in GFAS is applied also here, taking into account that, for fixed $\underline{\theta}_j$'s, the $u_{ij}$'s that minimize $J(\underline{\theta},U)$ are chosen as

$$u_{ij} = \begin{cases} 1, & \mathit{if}\ d(\underline{x}_i,\underline{\theta}_j) = \min_{k=1,\dots,m} d(\underline{x}_i,\underline{\theta}_k) \\ 0, & \mathit{otherwise} \end{cases}, \quad i = 1,\dots, N$$

## ❖ Hard Clustering Algorithms (cont)

➢ *Generalized Hard Algorithmic Scheme (GHAS)*

```
• Choose $\underline{\theta}_j(0)$ as initial estimates for $\underline{\theta}_j$, $j=1,\dots,m$.
• $t=0$
• Repeat
    – For $i=1$ to $N$
        o For $j=1$ to $m$
            Determination of the partition:
            $u_{ij}(t) = \begin{cases} 1, & \text{if } d(\underline{x}_i,\underline{\theta}_j(t)) = \min_{k=1,\dots,m} d(\underline{x}_i,\underline{\theta}_k(t)) \\ 0, & \text{otherwise} \end{cases}, \quad i=1,\dots,N$
        o End {For-$j$}
    – End {For-$i$}
    – $t=t+1$
```

## ❖ Hard Clustering Algorithms (cont)

- ***Generalized Hard Algorithmic Scheme (GHAS) (cont.)***
  - For $j=1$ to $m$
    - o *Parameter updating:* Solve

$$\sum_{i=1}^{N} u_{ij}(t-1)\frac{\partial d(\underline{x}_i,\underline{\theta}_j)}{\partial \underline{\theta}_j}=0$$

    - o with respect to $\underline{\theta}_j$ and set $\underline{\theta}_j(t)$ equal to the computed solution
  - End {For-$j$}
- Until a termination criterion is met

- Remarks:
  - In the update of each $\underline{\theta}_j$, only the vectors $\underline{x}_i$ for which $u_{ij}(t-1)=1$ are used.
  - GHAS may terminate when either
    - $||\underline{\theta}(t)-\underline{\theta}(t-1)||<\varepsilon$ or
    - $U$ remains unchanged for two successive iterations.

## ❖ Hard Clustering Algorithms (cont)

### ➢ More Remarks:

- For each hard clustering algorithm there exists a corresponding fuzzy clustering algorithm. The updating equations for the parameter vectors $\underline{\theta}_j$ in the hard clustering algorithms are obtained from their fuzzy counterparts for $q=1$.
- Hard clustering algorithms are not as robust as the fuzzy clustering algorithms when other than point representatives are used.
- The two-step optimization procedure in GHAS does not necessarily lead to a local minimum of $J(\underline{\theta},U)$.

## ❖ Hard Clustering Algorithms (cont)

### ➢ The Isodata or k-Means or c-Means algorithm

General comments

- It is a special case of GHAS where
  - Point representatives are used.
  - The squared Euclidean distance is employed.
- The cost function $J(\underline{\theta},U)$ becomes now

تابع هزینه

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

- Applying GHAS in this case, it turns out that it converges to a minimum of the cost function.
- Isodata recovers clusters that are as compact as possible.
- For other choices of the distance (including the Euclidean), the algorithm converges but not necessarily to a minimum of $J(\underline{\theta},U)$.

## ❖ Hard Clustering Algorithms (cont)

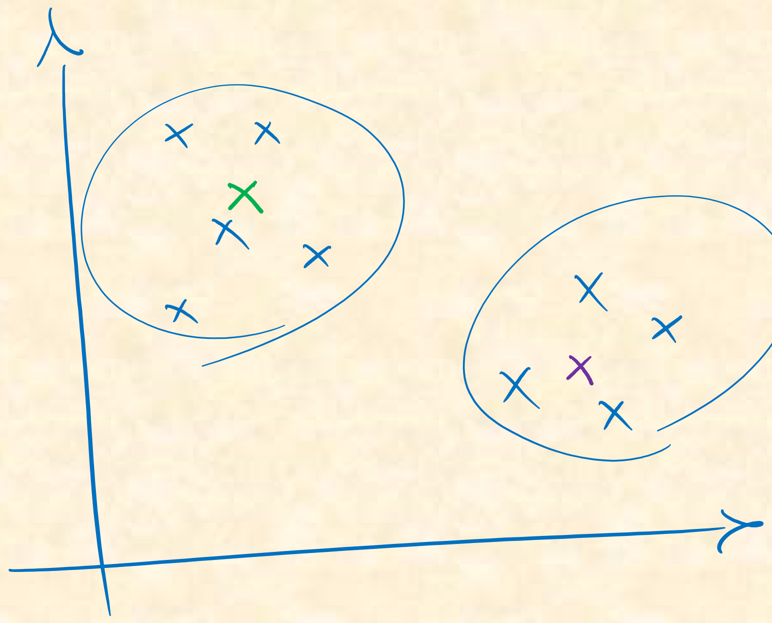
### ➢ The Isodata or k-Means or c-Means algorithm

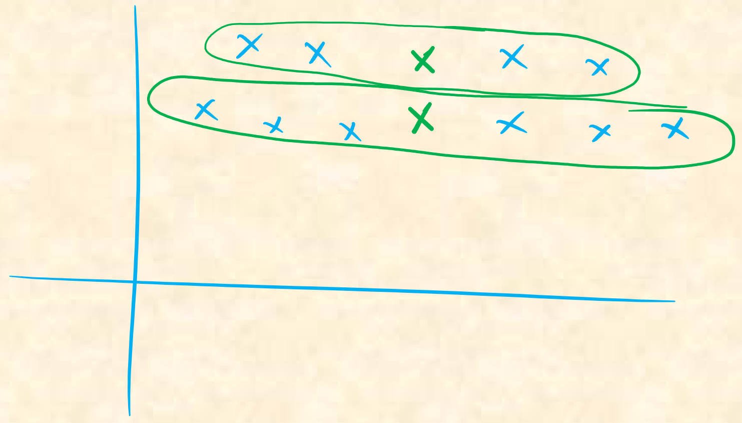
```
• Choose arbitrary initial estimates $\underline{\theta}_j(0)$ for the $\underline{\theta}_j$' s, $j=1,\ldots,m$.
  مراکز خوشه‌ها
• Repeat
  – For $i=1$ to $N$
      o Determine the closest representative, say $\underline{\theta}_j$, for $\underline{x}_i$
      o Set $b(i)=j$.
  – End {For}
  – For $j=1$ to $m$
      o *Parameter updating:* Determine $\underline{\theta}_j$ as the mean of the
        vectors $\underline{x}_i \in X$ with $b(i)=j$.
  – End {For}
• Until no change in $\underline{\theta}_j$' s occurs between two successive iterations
```

➢ Example 6(a): The k-means algorithm with $m=3$ identifies successfully the clusters in the data set of example 1(a). The confusion matrix is

$$A = \begin{bmatrix} 94 & 3 & 3 \\ 0 & 100 & 0 \\ 9 & 0 & 91 \end{bmatrix}$$

$m = 2$





## Example 14.12 (a)

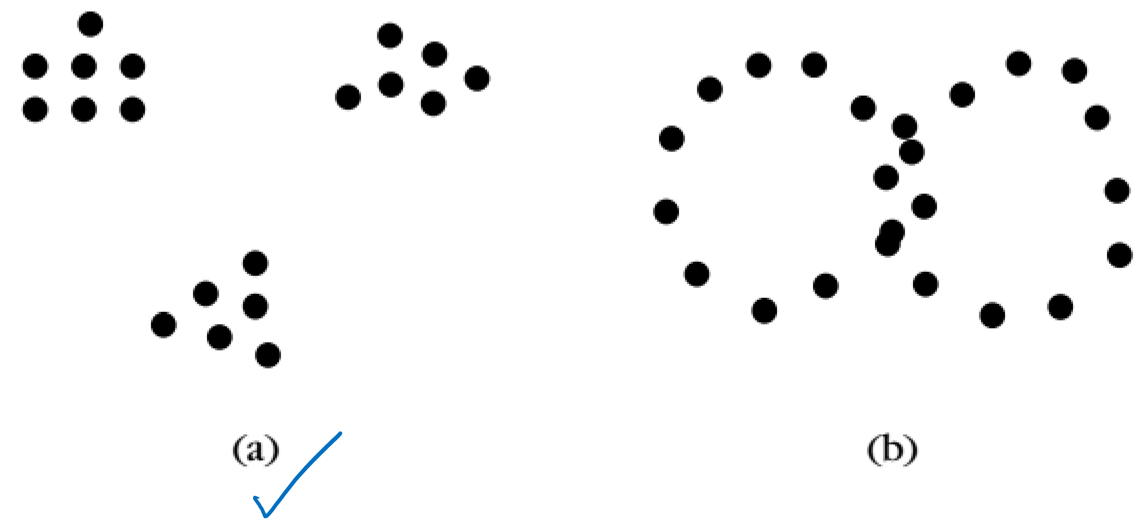


FIGURE 14.1

(a) Compact clusters. (b) Spherical clusters.

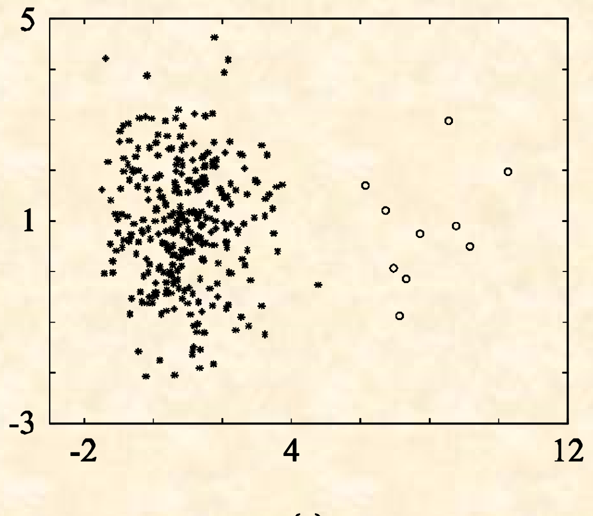
## ❖ Hard Clustering Algorithms – k-means (cont)

- Example 6(b): (i) Consider two 2-dimensional Gaussian distributions $N(\underline{\mu}_1,\Sigma_1)$, $N(\underline{\mu}_2,\Sigma_2)$, with $\underline{\mu}_1=[1, 1]^T$, $\underline{\mu}_2=[8, 1]^T$, $\Sigma_1=1.5I$ and $\Sigma_2=I$. (ii) Generate 300 points from the 1st distribution and 10 points from the 2nd distribution. (iii) Set $m=2$ and initialize randomly $\underline{\theta}_j$'s ($\underline{\theta}_j \equiv \underline{\mu}_j$).

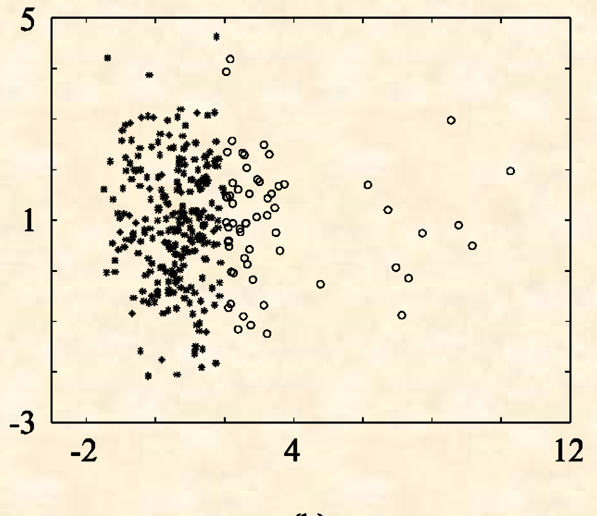
  After convergence the large group has been split into two clusters.

  Its right part has been assigned to the same cluster with the points of the small group (see figure below).

  This indicates that k-means cannot deal accurately with clusters having significantly different sizes.



(a)



(b)

## Example 14.12 (b)

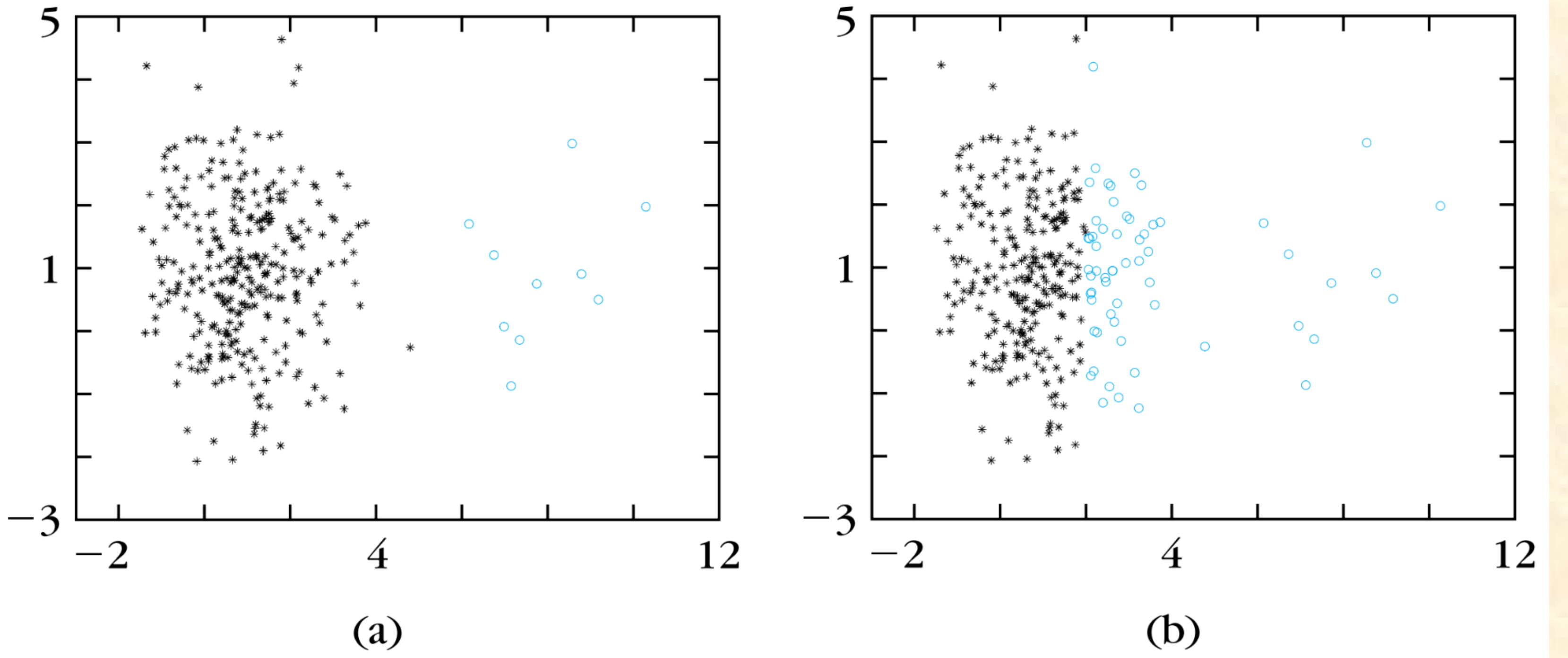


FIGURE 14.17

(a) The data set. (b) The results of the isodata algorithm.

## Hard Clustering Algorithms – k-means (cont)

- Remarks:
  - k-means recovers compact clusters.
  - Sequential versions of the k-means, where the updating of the representatives takes place immediately after the identification of the representative that lies closer to the current input vector $\underline{x}_i$, have also been proposed.
  - A variant of the k-means results if the number of vectors in each cluster is constrained *a priori*.
  - The computational complexity of the k-means is $O(Nmq)$, where $q$ is the number of iterations required for convergence. In practice, $m$ and $q$ are significantly less than $N$, thus, k-means becomes eligible for processing large data sets.

- Further remarks:

  Some drawbacks of the original k-means accompanied with the variants of the k-means that deal with them are discussed next.

## ❖ Hard Clustering Algorithms – k-means (cont)

➢ Drawback 1: *Different initial partitions may lead k-means to produces different final clusterings, each one corresponding to a different local minimum.*

Strategies for facing drawback 1:

- Single run methods
  - Use a sequential algorithm (discussed previously) to produce initial estimates for $\underline{\theta}_j$'s.
  - Partition randomly the data set into $m$ subsets and use their means as initial estimates for $\underline{\theta}_j$' s.
- Multiple run methods
  - Create different partitions of $X$, run k-means for each one of them and select the best result.
  - Compute the representatives iteratively, one at a time, by running k-means $mN$ times. It is claimed that convergence is independent of the initial estimates of $\underline{\theta}_j$' s.
- Utilization of tools from stochastic optimization techniques (simulated annealing, genetic algorithms etc).

## ❖ Hard Clustering Algorithms – k - means (cont)

➢ Drawback 2: *Knowledge of the number of clusters* $m$ *is required a priori.*

Strategies for facing drawback 2:

- Employ splitting, merging and discarding operations of the clusters resulting from k-means.
- Estimate $m$ as follows:
  - Run a sequential algorithm many times for different thresholds of dissimilarity $\Theta$.
  - Plot $\Theta$ versus the number of clusters and identify the largest plateau in the graph and set $m$ equal to the value that corresponds to this plateau.

## ❖ Hard Clustering Algorithms – k - means (cont)

➢ Drawback 3: *k-means is sensitive to outliers and noise.*

Strategies for facing drawback 3:

- Discard all "small" clusters (they are likely to be formed by outliers).
- Use a k-medoids algorithm (see below), where a cluster is represented by one of its points.

➢ Drawback 4: *k-means is not suitable for data with nominal (categorical) coordinates.*

Strategies for facing drawback 4:

- Use a k-medoids algorithm.

## ❖ Hard Clustering Algorithms

➢ *k-Medoids Algorithms*

- Each cluster is represented by a vector selected among the elements of $X$ (medoid).
- A cluster contains
  - Its medoid
  - All vectors in $X$ that
    - Are not used as medoids in other clusters
    - Lie closer to its medoid than the medoids representing other clusters.

Let $\Theta$ be the set of medoids of all clusters, $I_{\Theta}$ the set of indices of the points in $X$ that constitute $\Theta$ and $I_{X\text{-}\Theta}$ the set of indices of the points that are not medoids.

- Obtaining the set of medoids $\Theta$ that best represents the data set, $X$ is equivalent to minimizing the following cost function

## k-Medoids Algorithms (cont)

$$J(\Theta, U) = \sum_{i \in I_{X-\Theta}} \sum_{j \in I_\Theta} u_{ij} d(\underline{x}_i, \underline{x}_j)$$

with

$$u_{ij} = \begin{cases} 1, & if \;\; d(\underline{x}_i, \underline{x}_j) = \min_{q \in I_\Theta} d(\underline{x}_i, \underline{x}_q) \\ 0, & otherwise \end{cases}, \quad i = 1, \ldots, N$$

- Representing clusters with mean malues vs representing clusters with medoids

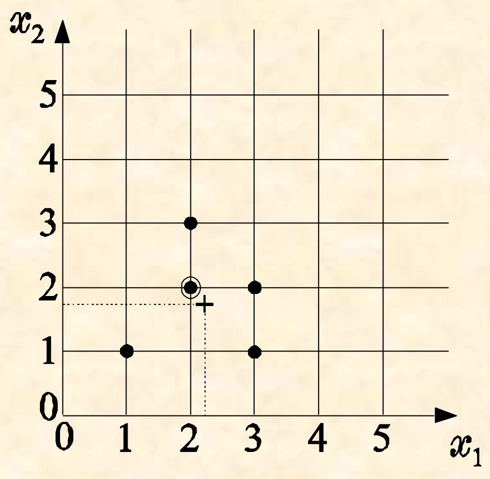
| Mean Values | Medoids |
|---|---|
| 1. Suited only for continuous domains | 1. Suited for either cont. or discrete domains |
| 2. Algorithms using means are sensitive to outliers | 2. Algorithms using medoids are less sensitive to outliers |
| 3. The mean possess a clear geometrical and statistical meaning | 3. The medoid has not a clear geometrical meaning |
| 4. Algorithms using means are not computationally demanding | 4. Algorithms using medoids are more computationally demanding |

## ➢ k-Medoids Algorithms (cont)

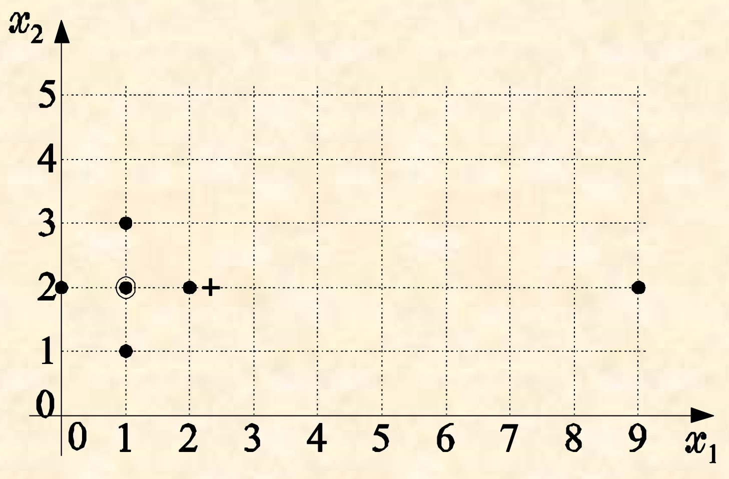
➢ Example 7: (It illustrates the first two points in the above comparison)

(a) The five-point two-dimensional set stems from the discrete domain $D=\{1,2,3,4,\ldots\}x\{1,2,3,4,\ldots\}$. Its medoid is the circled point and its mean is the "+" point, which does not belong to $D$.

(b) In the six-point two-dimensional set , the point $(9,2)$ can be considered as an outlier. While the outlier affects significantly the mean of the set, it does not affect its medoid.



(a)



(b)

## ❖ Hard Clustering Algorithms - k-Medoids Algorithms (cont)

Algorithms to be considered

- PAM (Partitioning Around Medoids)
- CLARA (Clustering LARge Applications)
- CLARANS (Clustering Large Applications based on RANdomized Search)

- *The PAM algorithm*
  - The number of clusters $m$ is required *a priori*.

  Definitions-preliminaries

  - Two sets of medoids $\Theta$ and $\Theta'$, each one consisting of $m$ elements, are called neighbors if they **share** $m$-1 elements.
  - A set $\Theta$ of medoids with $m$ elements can have $m(N-m)$ neighbors.
  - Let $\Theta_{ij}$ denote the neighbor of $\Theta$ that results if $\underline{x}_j$, $j \in I_{X-\Theta}$ replaces $\underline{x}_i$, $i \in I_{\Theta}$.
  - Let $\Delta J_{ij} = J(\Theta_{ij}, U_{ij}) - J(\Theta, U)$.

## ❖ Hard Clustering Algorithms - k-Medoids Algorithms (cont)

- ***The PAM algorithm***
  - *Determination of $\Theta$ that best represents the data*
    - Generate a set $\Theta$ of $m$ medoids, randomly selected out of $X$.
    - (A) Determine the neighbor $\Theta_{qr}$, $q \in I_\Theta$, $r \in I_{X-\Theta}$ among the $m(N-m)$ neighbors of $\Theta$ for which $\Delta J_{qr} = \min_{i \in I_\Theta, j \in I_{X-\Theta}} \Delta J_{ij}$.
    - If $\Delta J_{qr} < 0$ is negative then
      - o Replace $\Theta$ by $\Theta_{qr}$
      - o Go to (A)
    - End
  - *Assignment of points to clusters*
    - Assign each $\underline{x} \in I_{X-\Theta}$ to the cluster represented by the closest to $\underline{x}$ medoid.

Computation of $\Delta J_{ij}$. It is defined as: $\Delta J_{ij} = \sum_{h \in I_{X-\Theta}} C_{hij}$

where $C_{hij}$ is the difference in $J$, resulting from the (possible) assignment of the vector $\underline{x}_h \in X-\Theta$ from the cluster it currently belongs to another, as a consequence of the replacement of $\underline{x}_i \in \Theta$ by $\underline{x}_j \in X-\Theta$.

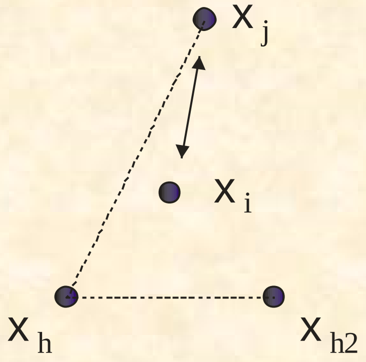
## ❖ Hard Clustering Algorithms - k-Medoids Algorithms (cont)

➢ *The PAM algorithm (cont)*

Computation of $\underline{C}_{hij}$:

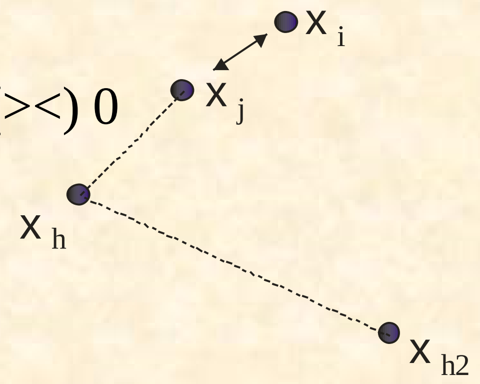
- $\underline{x}_h$ belongs to the cluster represented by $\underline{x}_i$ ($\underline{x}_{h2} \in \Theta$ denotes the second closest to $\underline{x}_h$ representative) and $d(\underline{x}_h, \underline{x}_j) \geq d(\underline{x}_h, \underline{x}_{h2})$. Then

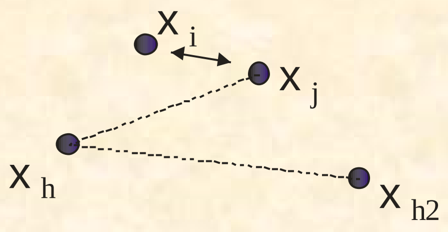
$$C_{hi\,j} = d(\underline{x}_h, \underline{x}_{h2}) - d(\underline{x}_h, \underline{x}_i) \geq 0$$



- $\underline{x}_h$ belongs to the cluster represented by $\underline{x}_i$ ($\underline{x}_{h2} \in \Theta$ denotes the second closest to $\underline{x}_h$ representative) and $d(\underline{x}_h, \underline{x}_j) \leq d(\underline{x}_h, \underline{x}_{h2})$. Then

$$C_{hij} = d(\underline{x}_h, \underline{x}_j) - d(\underline{x}_h, \underline{x}_i)\ (><)\ 0$$



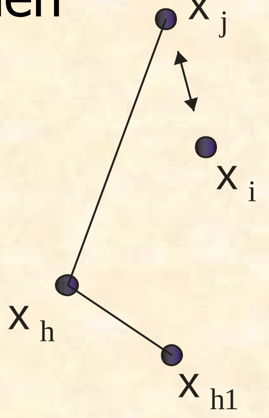


## ❖ Hard Clustering Algorithms - k-Medoids Algorithms (cont)

➢ ***The PAM algorithm (cont)***

Computation of $\underline{C}_{hij}$ (cont.):

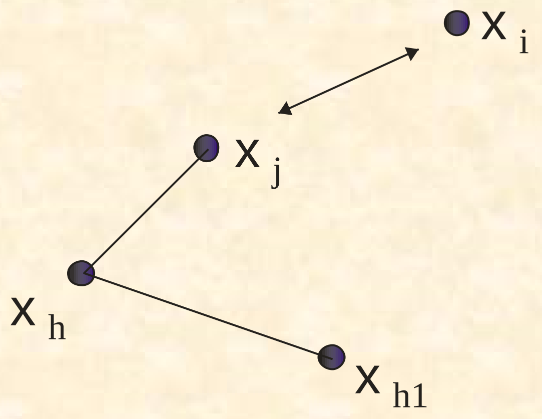
- $\underline{x}_h$ is not represented by $\underline{x}_i$ ($\underline{x}_{h1}$ denotes the closest to $\underline{x}_h$ medoid) and $d(\underline{x}_h, \underline{x}_{h1}) \leq d(\underline{x}_h, \underline{x}_j)$. Then



$C_{hij}=0$

- $\underline{x}_h$ is not represented by $\underline{x}_i$ ($\underline{x}_{h1}$ denotes the closest to $\underline{x}_h$ medoid) and $d(\underline{x}_h, \underline{x}_{h1}) > d(\underline{x}_h, \underline{x}_j)$. Then

$C_{hij} = d(\underline{x}_h, \underline{x}_j) - d(\underline{x}_h, \underline{x}_{h1}) < 0$



## ❖ Hard Clustering Algorithms - k-Medoids Algorithms (cont)

➢ *The PAM algorithm (cont)*

➢ Remarks:

- Experimental results show the PAM works satisfactorily with small data sets.
- Its computational complexity per iteration is $O(m(N\text{-}m)^2)$. Unsuitable for large data sets.

## ❖ Hard Clustering Algorithms - k-Medoids Algorithms (cont)

➢ *The CLARA algorithm*

- It is more suitable for large data sets.
- The strategy:
  - Draw randomly a sample $X'$ of size $N'$ from the entire data set.
  - Run the PAM algorithm to determine $\Theta'$ that best represents $X'$.
  - Use $\Theta'$ in the place of $\Theta$ to represent the entire data set $X$.
- The rationale:
  - Assuming that $X'$ has been selected in a way representative of the statistical distribution of the data points in $X$, $\Theta'$ will be a good approximation of $\Theta$, which would have been produced if PAM were run on $X$.
- The algorithm:
  - Draw $s$ sample subsets of size $N'$ from $X$, denoted by $X'_1,\ldots,X'_s$ (typically $s = 5$, $N' = 40+2m$).
  - Run PAM on each one of them and identify $\Theta'_1,\ldots,\Theta'_s$.
  - Choose the set $\Theta'_j$ that minimizes

$$J(\Theta',U) = \sum_{i \in I_{X-\Theta'}} \sum_{j \in I_{\Theta'}} u_{ij} d(\underline{x}_i, \underline{x}_j)$$

based on the entire data set $X$.

## ❖ Hard Clustering Algorithms - k-Medoids Algorithms (cont)

➢ *The CLARANS algorithm*

- It is more suitable for large data sets.
- It follows the philosophy of PAM with the difference that only a fraction $q(<m(N-m))$ of the neighbors of the current set of medoids is considered.
- It performs several runs ($s$) starting from different initial conditions for $\Theta$.
- The algorithm:

```
– For $i=1$ to $s$
    o Initialize randomly $\Theta$.
    o (A) Select randomly $q$ neighbors of $\Theta$.
    o For $j=1$ to $q$
        * If the present neighbor of $\Theta$ is better than $\Theta$ (in terms of $J(\Theta,U)$)
          then
            -- Set $\Theta$ equal to its neighbor
            -- Go to (A)
        * End If
    o End For
    o Set $\Theta^i=\Theta$
– End For
– Select the best $\Theta^i$ with respect to $J(\Theta,U)$.
– Based on this set of medoids assign each vector $\underline{x}\in X-\Theta$ to
  the cluster whose representative is closest to $\underline{x}$.
```

## Hard Clustering Algorithms - k-Medoids Algorithms (cont)

- ***The CLARANS algorithm (cont)***
- Remarks:
  - CLARANS depends on $q$ and $s$. Typically, $s=2$ and $q=\max(0.125m(N-m), 250)$
  - As $q$ approaches $m(N-m)$ CLARANS approaches PAM and the complexity increases.
  - CLARANS can also be described in terms of graph theory concepts.
  - CLARANS unravels better quality clusters than CLARA.
  - In some cases, CLARA is significantly faster than CLARANS.
  - CLARANS retains its quadratic computational nature and thus it is not appropriate for very large data sets.