Clustering algorithms

Konstantinos Koutroumbas

Unit 3

- Missing data
- Sequential clustering algorithms
- Hard CFO clustering algorithms

koutroum@noa.gr

Proximity measures between vectors – Missing data

Missing data

- For some vectors of the data set X, some features values are unknown/missing.

- This issue arises very often in practice.

- It may be caused by a measurement device failure, inability to take measure due to specific physical conditions etc.

- Ways to *deal with this situation*:

  ✓ **Discard** all vectors with missing values (not recommended for small data sets).

  ✓ **Find** the *mean value* m_k of the available k-th feature values over that data set and **substitute** the missing k-th feature values with m_k.

Proximity measures between vectors – Missing data

Missing data

- Ways to *deal with this situation* (cont.):
  - ✓ **Define** $b_k = 0$, if **both** the k-*th* features x_k, y_k are **available** and 1 **otherwise**. Then

$$\wp(\boldsymbol{x}, \boldsymbol{y}) = \frac{l}{l - \sum_{k=1}^{l} b_k} \sum_{all\ k:\ b_k = 0} \phi(x_k, y_k)$$

where $\phi(x_k, y_k)$ denotes the proximity measure between two scalars x_k, y_k.
NOTE: The proximity is **based** only on the features for which both x_k, y_k are available.

  - ✓ For the k-th feature, $k = 1,2, \dots, l$, **find** the average proximity $\phi_{avg}(k)$ among all available values along the feature vectors in X. Then

$$\wp(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{l} \psi(x_k, y_k),$$

where $\psi(x_k, y_k) = \begin{cases} \phi(x_k, y_k), & if\ both\ x_k, y_k\ are\ available \\ \phi_{avg}(k), & otherwise \end{cases}$

Proximity measures between vectors – Missing data

Missing data

Exercise 4: Consider the data set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5\}$, with $\boldsymbol{x_1} = [0,0]^T, \boldsymbol{x_2} = [1,*]^T, \boldsymbol{x_3} = [0,*]^T, \boldsymbol{x_4} = [2,2]^T, \boldsymbol{x_5} = [3,1]^T$ ("*" stands for missing values).

(a) Compute the l_1 distances between all pairs of vectors, using all the four techniques for dealing with missing data.

(b) In which of these techniques, the computed distances are dependent on the specific data set?

Clustering algorithms

Number of possible clusterings

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

Question: In how many ways the N points of X can be assigned into m groups?

Answer: $$S(N,m) = \frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-1}\binom{m}{i}i^N$$

Examples:

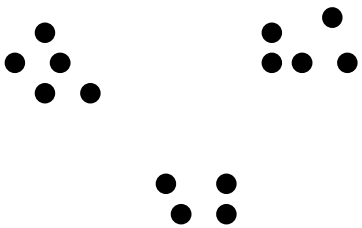
- $S(15,3) = 2{,}375{,}101$
- $S(20,4) = 45{,}232{,}115{,}901$
- $S(25,8) = 690{,}223{,}721{,}118{,}368{,}580$
- $S(100,5) \approx 10^{68}$!!

NOTE: The above calculations are for *fixed* m. If this varies, then we have to enumerate all clusterings, for all possible values of m!!

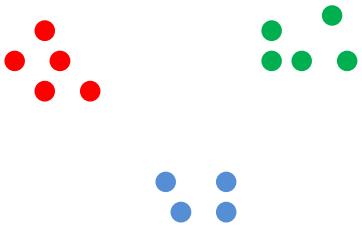
$\Rightarrow$

Evaluating all possible clusterings is impractical even for moderate values of N.

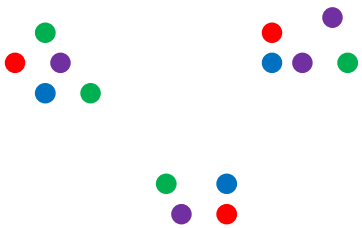
Clustering algorithms



Data set



A “sensible” clustering



A “less sensible” clustering

- Clustering algorithms may be **viewed** as schemes that *provide us with sensible clusterings by considering only a small fraction of all possible partitions of* X.
- This *fraction* **depends on** the adopted criteria.
- Thus a clustering algorithm tries to **identify** clusters formed by the data vectors, **in accordance to the adopted criteria**.

Clustering algorithms

Major categories of clustering algorithms

A vast amount of algorithms **exists** based on very diverse criteria

$\Rightarrow$ Strict categorization is extremely difficult (rather impossible).

A rough categorization:

- Sequential: A single clustering is produced. One or few sequential passes on the data.

- Hierarchical: A sequence of (nested) clusterings is produced.
  - Agglomerative
    - Matrix theory
    - Graph theory
  - Divisive
  - Combinations of the above (e.g., the Chameleon algorithm.)

Clustering algorithms

Major categories of clustering algorithms

A rough categorization:

Cost function optimization.

- For most of the cases a *single* clustering is obtained.
- They can be further **categorized** through the notion of "belongness".

Hard clustering (each point **belongs** exclusively to a single cluster):

- Basic hard clustering algorithms (e.g., k-means)
- k-medoids algorithms
- Mixture decomposition
- Branch and bound
- Simulated annealing
- Deterministic annealing
- Boundary detection
- Mode seeking
- Genetic clustering algorithms

Probabilistic clustering (a hard clustering case where probabilistic framework is utilized)

Fuzzy clustering (each point **belongs** to more than one clusters simultaneously).

Possibilistic clustering (it is based on the notion of the *"degree of compatibility"* of a point with a cluster).

Clustering algorithms

Major categories of clustering algorithms

A rough categorization:

Other.

- Algorithms based on graph theory (e.g., Spectral clustering, Minimum Spanning Tree, regions of influence, directed trees).
- Density-based algorithms.
- Competitive learning algorithms (basic competitive learning scheme, Kohonen self organizing maps).
- Subspace clustering algorithms.
- Ensemble of clusterings
- Kernel-based methods.
- …

Sequential clustering algorithms

The common traits shared by the sequential clustering algorithms are:

- One or very few passes on the data are required.
- The number of clusters m is not known a-priori, except (possibly) an upper bound of it, q.
- The clusters are **defined** with the aid of
  - ✓ An appropriately defined distance $d(\boldsymbol{x}, C)$ of a point from a cluster.
  - ✓ A threshold Θ associated with the distance.

Sequential clustering algorithms

Basic Sequential Clustering Algorithmic Scheme (BSAS)

- $m = 1$ \{number of clusters\}\
- $C_m = \{\boldsymbol{x}_1\}$
- For $i = 2$ to N
  - **Find** C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
  - If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m + 1$
    - $C_m = \{\boldsymbol{x}_i\}$
  - Else
    - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
    - Where necessary, update representatives (*)
  - End {if}
- End {for}

$X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$

NOTE: Different choices of cluster representatives and/or different choices of the distance between a point and a set **give rise** to different algorithms, falling under the umbrella of the **Basic Sequential Algorithm Scheme**.

(*) When the mean vector $\boldsymbol{m}_C$ is used as representative of the cluster C with n_C elements, the updating in the light of a new vector $\boldsymbol{x}$ becomes

$$\boldsymbol{m}_C{}^{new} = (n_C \, \boldsymbol{m}_C{}^{old} + \boldsymbol{x}) / (n_C + 1)$$

Sequential clustering algorithms

Basic Sequential Clustering Algorithmic Scheme (BSAS)

Remarks:

- The order of presentation of the data in the algorithm plays important role in the clustering results. Different order of presentation may lead to totally different clustering results, in terms of the number of clusters as well as the clusters themselves.

- The clustering results depend on the choice of the value of Θ.

- In BSAS the decision for a vector $\boldsymbol{x}$ is reached prior to the final cluster formation.

- BSAS perform a single pass on the data. Its complexity is $O(N)$ (when point representatives are used).

- If clusters are represented by point representatives, compact clusters are favored.

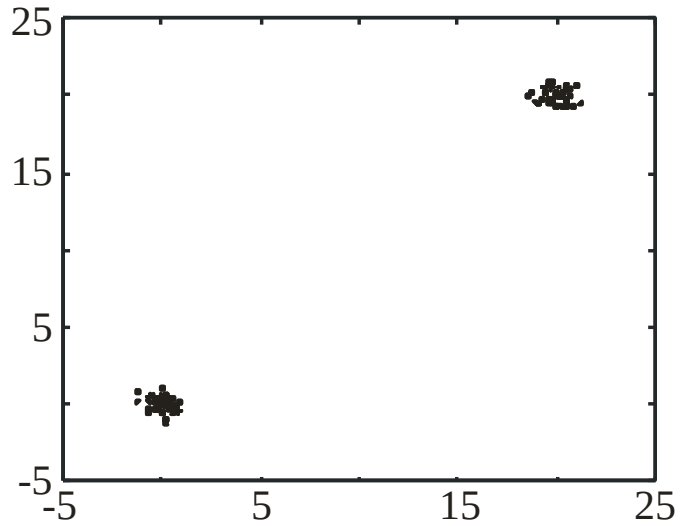
Sequential clustering algorithms

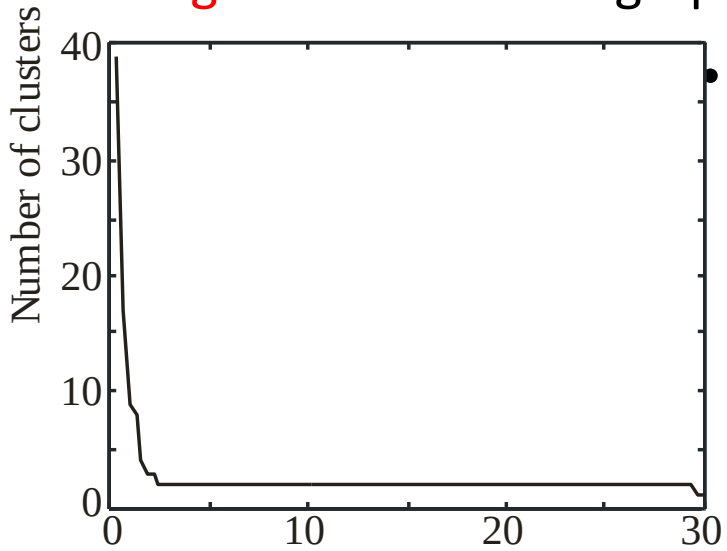
Basic Sequential Clustering Algorithmic Scheme (BSAS)

Estimating the number of clusters in the data set:

Let $BSAS(\Theta)$ denote the $BSAS$ algorithm when the dissimilarity threshold is Θ.

- For $\Theta = a$ to b step c
  - **Run** s times $BSAS(\Theta)$, each time presenting the data in a different order.
  - **Estimate** the number of clusters m_Θ, as the most frequent number resulting from the s runs of $BSAS(\Theta)$.
- Next Θ
- **Plot** m_Θ versus Θ and identify the number of clusters m as the one corresponding to the widest flat region in the above graph.





- **Consider** as final clustering, the clustering that results for the Θ in the middle of the widest flat region.

Sequential clustering algorithms

MBSAS, a Modification of BSAS

- In BSAS a decision for a data vector $\boldsymbol{x}$ is reached prior to the final cluster formation, which is determined after all vectors have been presented to the algorithm.

- MBSAS address this issue, at the cost of performing two passes on the data.

- MBSAS consists of:
  - A cluster determination phase (first pass on the data), which is the same as BSAS with the **exception** that no vector is assigned to an already formed cluster. At the end of this phase, each cluster **consists** of a single element.
  - A pattern classification phase (second pass on the data), where each one of the unassigned vectors is **assigned** to its closest cluster.

Exercise: Write the pseudocode for MBSAS (in the spirit of the BSAS pseudocode).

Remarks:

- In MBSAS, a decision for a vector $\boldsymbol{x}$ during the pattern classification phase is reached taking into account all clusters.
- MBSAS is sensitive to the order of presentation of the vectors.
- MBSAS requires two passes on the data. Its complexity is $O(N)$.

Sequential clustering algorithms

Refinement stages

The problem of closeness of clusters: "*In all the above algorithms it may happen that two formed clusters lie very close to each other*".
(they may be parts of the same **physical** cluster)

A simple merging procedure

(*A*) **Find** C_i, C_j $(i < j)$ such that $d(C_i, C_j) = min_{k,r=1,\cdots,m,k\neq r}\, d(C_k, C_r)$
If $d(C_i, C_j) \leq M_1$ then \{ M_1 is a user-defined threshold \}
- –**Merge** C_i, C_j to C_i and eliminate C_j .
- –If necessary, update the cluster representative of C_i .
- –Rename the clusters $C_{j+1}, \ldots, C_m$ to $C_j, \ldots, C_{m-1}$, respectively.
- $-m = m - 1$
- –Go to (*A*)

Else
- –Stop

End {if}

Sequential clustering algorithms

Refinement stages

The problem of sensitivity to the order of data presentation:
"*A vector $\boldsymbol{x}$ may have been assigned to a cluster C_i at the current stage but another cluster C_j may be formed at a later stage that lies closer to $\boldsymbol{x}$*"

A simple reassignment procedure

- **For** $i = 1$ to N
  - **Find** C_j such that $d(\boldsymbol{x}_i, C_j) = min_{k=1,\cdots,m}\, d(\boldsymbol{x}_i, C_k)$
  - **Set** $b(i) = j$ \{ $b(i)$ is the index of the cluster that lies closest to $\boldsymbol{x}_i$ \}
- **End** {for}

- **For** $j = 1$ to m
  - **Set** $C_j = \{\boldsymbol{x}_i \in X:\ b(i) = j\}$
  - If necessary, update representatives
- End {for}

Sequential clustering algorithms

A two-threshold sequential scheme (TTSAS)

- The formation of the clusters, as well as the assignment of vectors to clusters, is carried out concurrently (like BSAS and unlike MBSAS)
- Two thresholds Θ_1 and Θ_2 ($\Theta_1 < \Theta_2$) are **employed**.
- The general idea is the following:

If the distance $d(\boldsymbol{x}, C)$ of $\boldsymbol{x}$ from its closest cluster, C, is greater than Θ_2 then:
  –A new cluster represented by $\boldsymbol{x}$ is created.
Else if $d(\boldsymbol{x}, C) < \Theta_1$ then
  –$\boldsymbol{x}$ is **assigned** to C.
Else
  –The decision is **postponed** to a later stage.
End {if}

- *The unassigned vectors are presented iteratively to the algorithm until all of them are classified.*

Remarks:

- In practice, a few passes (≥ 2) of the data set are required.
- TTSAS is less sensitive to the order of data presentation, compared to BSAS.

Sequential clustering algorithms

The maxmin algorithm
(only for **point representatives**)

W may be **initialized** by **(a)** the two most distant points or **(b)** the mean of the data set.

Let W be the set of all points that have been chosen to define clusters up to the current iteration step. The definition of clusters is carried out as follows:

- For each $\boldsymbol{x} \in X - W$ **determine** $d_{\boldsymbol{x}} = min_{\boldsymbol{z} \in W}\, d(\boldsymbol{x}, \boldsymbol{z})$
- **Determine** $\boldsymbol{y}$: $d_{\boldsymbol{y}} = max_{x \in X - W} d_{\boldsymbol{x}}$
- **If** $d_{\boldsymbol{y}}$ is greater than a prespecified threshold (Θ) then
  – $\boldsymbol{y}$ **defines** a new cluster
- **else**
  –the cluster determination phase of the algorithm terminates.
- End {if}

After the definition of the clusters, each unassigned vector is **assigned** to its closest cluster.

Remarks:

- The maxmin algorithm is more computationally demanding than MBSAS.
- Its result is independent of the order of data presentation to the algorithm.
- It is expected to produce better clustering results than MBSAS.
- Its performance may be **degraded** in the presence of noise.

Sequential clustering algorithms

Example in MATLAB 1:

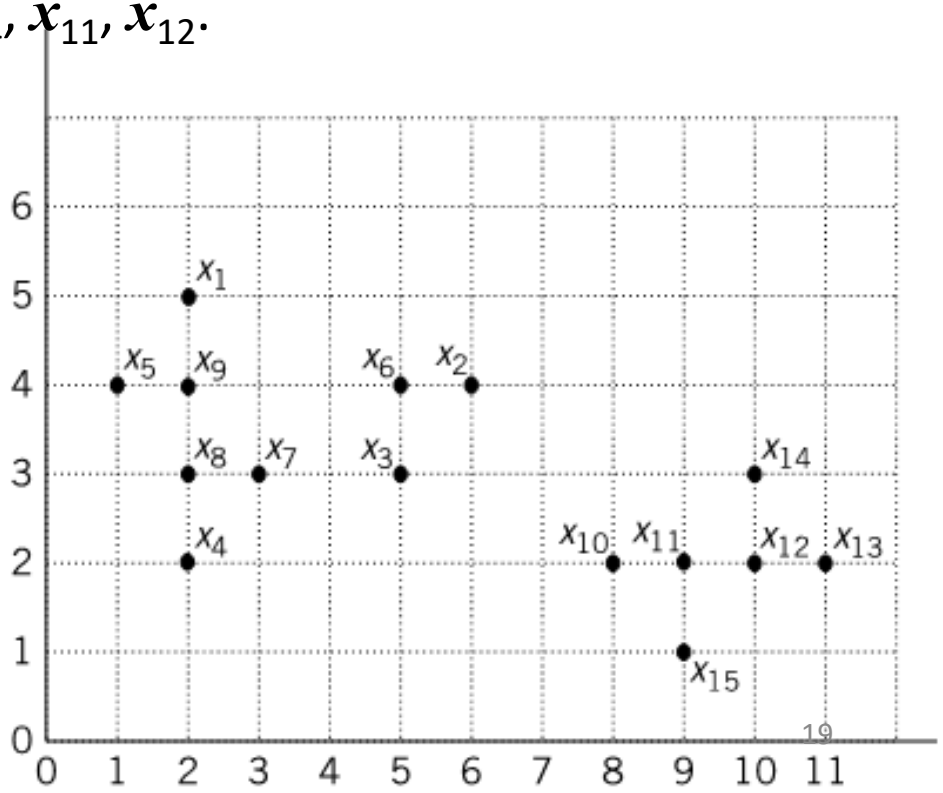
Consider the data vectors depicted in the figure below and perform a "visual" clustering on it.

1. Apply the BSAS algorithm on X, presenting its elements in the order $\boldsymbol{x}_8, \boldsymbol{x}_6, \boldsymbol{x}_{11}, \boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_7, \boldsymbol{x}_{10}, \boldsymbol{x}_9, \boldsymbol{x}_{12}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{15}$, for $\Theta = 2.5$ and $q = 15$.

2. Repeat step 1, now with the order of presentation to the algorithm as $\boldsymbol{x}_7, \boldsymbol{x}_3, \boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_9, \boldsymbol{x}_6, \boldsymbol{x}_8, \boldsymbol{x}_4, \boldsymbol{x}_2, \boldsymbol{x}_{10}, \boldsymbol{x}_{15}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{11}, \boldsymbol{x}_{12}$.

3. Repeat step 1, now with $\Theta = 1.4$.

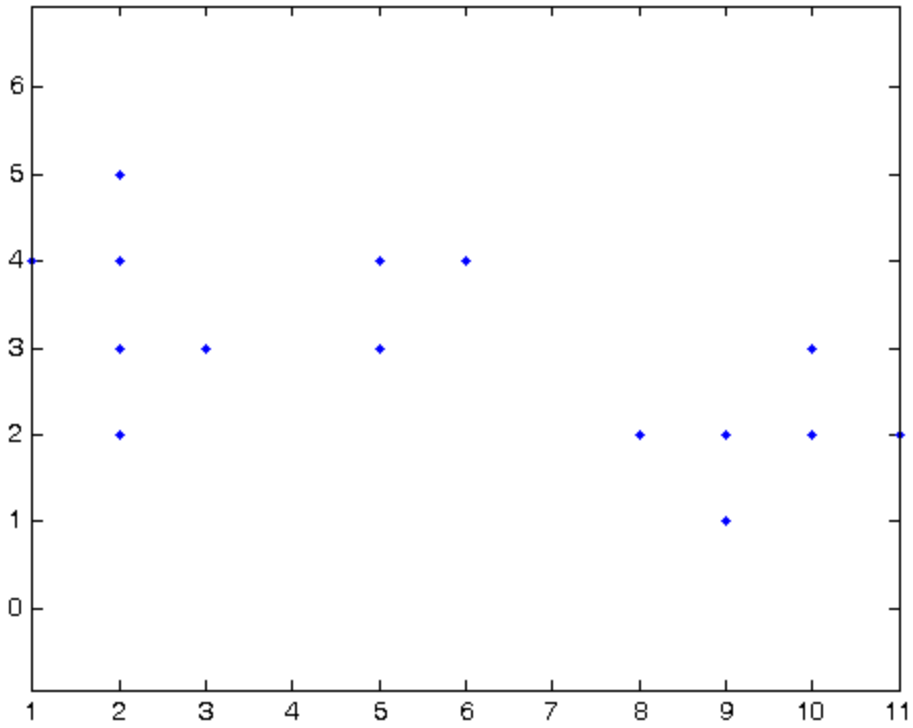
4. Repeat step 1, now with $q = 2$.



Sequential clustering algorithms

Example in MATLAB 1: Results

Consider the data vectors depicted in the figure below and perform a "visual" clustering on it.

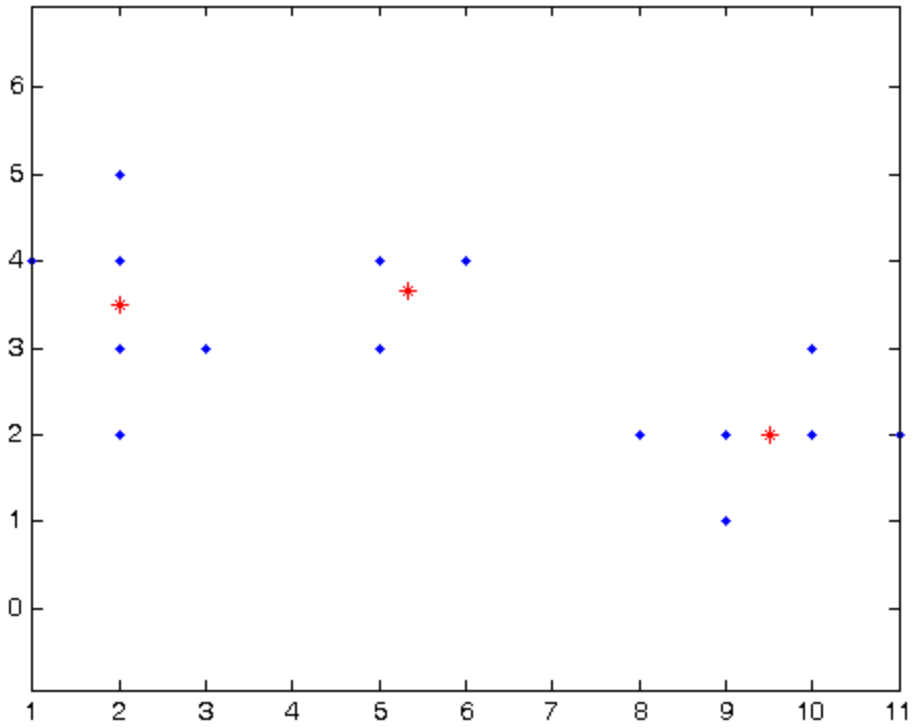


The data points

Sequential clustering algorithms

Example in MATLAB 1: Results

Consider the data vectors depicted in the figure below and perform a "visual" clustering on it.

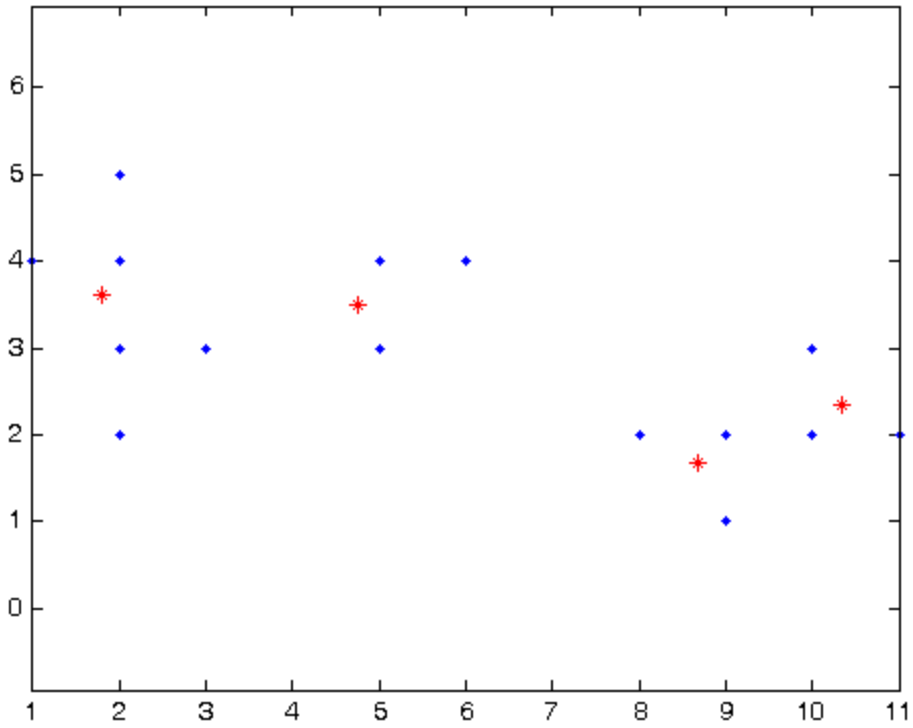


1. $\boldsymbol{x}_8, \boldsymbol{x}_6, \boldsymbol{x}_{11}, \boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_7, \boldsymbol{x}_{10}, \boldsymbol{x}_9, \boldsymbol{x}_{12}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{15}$, $\Theta = 2.5$, $q = 15$.

Sequential clustering algorithms

Example in MATLAB 1: Results

Consider the data vectors depicted in the figure below and perform a "visual" clustering on it.

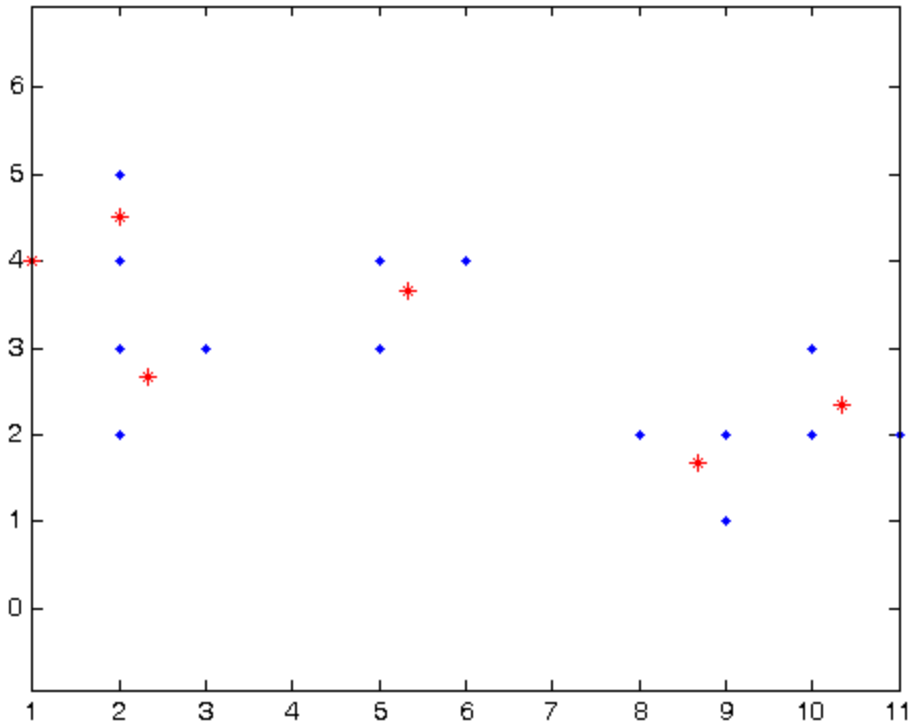


2. $\boldsymbol{x}_7, \boldsymbol{x}_3, \boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_9, \boldsymbol{x}_6, \boldsymbol{x}_8, \boldsymbol{x}_4, \boldsymbol{x}_2, \boldsymbol{x}_{10}, \boldsymbol{x}_{15}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{11}, \boldsymbol{x}_{12}$, $\Theta = 2.5, q = 15.$

Sequential clustering algorithms

Example in MATLAB 1: Results

Consider the data vectors depicted in the figure below and perform a "visual" clustering on it.

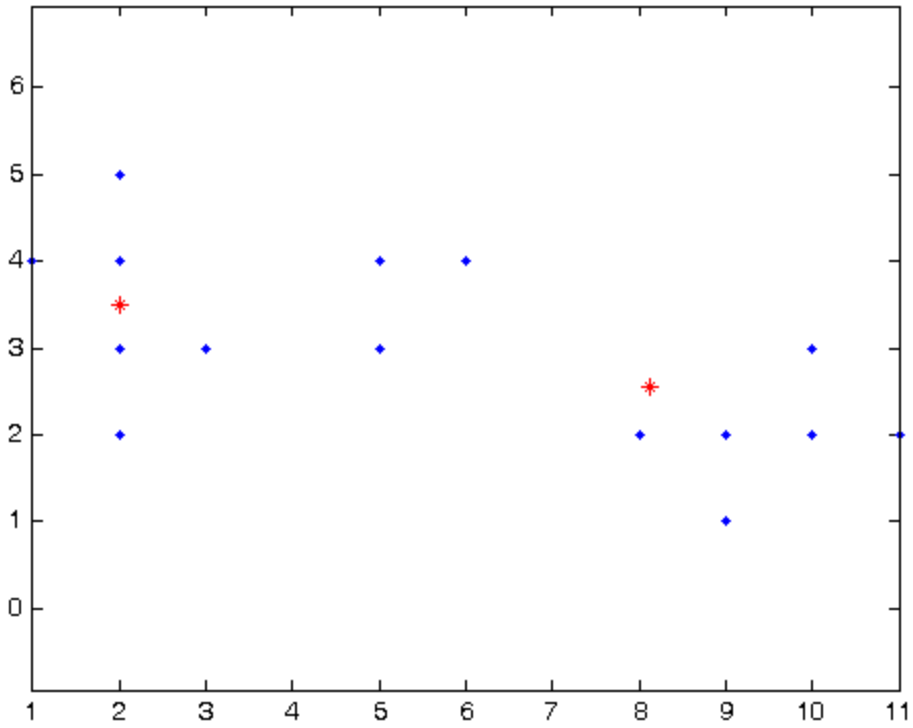


3. $x_8, x_6, x_{11}, x_1, x_5, x_2, x_3, x_4, x_7, x_{10}, x_9, x_{12}, x_{13}, x_{14}, x_{15}$, $\Theta = 1.4$, $q = 15$.

Sequential clustering algorithms

Example in MATLAB 1: Results

Consider the data vectors depicted in the figure below and perform a "visual" clustering on it.



4. $\boldsymbol{x}_8, \boldsymbol{x}_6, \boldsymbol{x}_{11}, \boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_7, \boldsymbol{x}_{10}, \boldsymbol{x}_9, \boldsymbol{x}_{12}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{15}$, $\Theta = 2.5, q = 2.$

Sequential clustering algorithms

Example in MATLAB 2:

Generate and plot a data set X_1, that consists of $N = 400$ 2-dim. data vectors. These vectors form **four groups**, each one of which contains vectors that stem from Gaussian distributions with means $\boldsymbol{m}_1 = [0, 0]^T$, $\boldsymbol{m}_2 = [4, 0]^T$, $\boldsymbol{m}_3 = [0, 4]^T$, $\boldsymbol{m}_4 = [5, 4]^T$, respectively, and respective covariance matrices $S_1 = I$, $S_2 = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1.5 \end{bmatrix}$, $S_3 = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1.1 \end{bmatrix}$, $S_4 = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.5 \end{bmatrix}$. Then do the following:

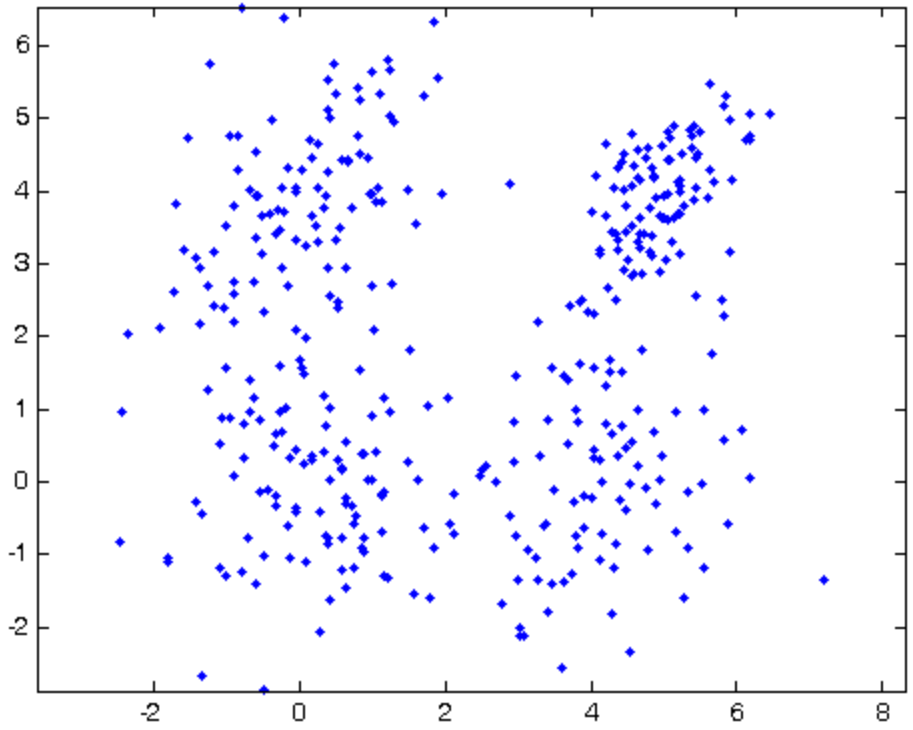
1. Determine the number of clusters formed in X_1 by doing the following:

   a. Determine the maximum, d_{max}, and the minimum, d_{min}, distances between any two points in the data set.

   b. Determine the values of Θ for which the BSAS will run. These may be defined as Θ_{min}, $\Theta_{min} + s$, $\Theta_{min} + 2s$,...,Θ_{max}, where $\Theta_{min} = 0.25\,\frac{d_{min}+d_{max}}{2}$, $\Theta_{max} = 1.75\,\frac{d_{min}+d_{max}}{2}$ and $s = \frac{\Theta_{min}+\Theta_{max}}{n_\Theta}$, n_Θ is the number of successive values of Θ that will be considered. Use $n_\Theta = 50$.

Sequential clustering algorithms

Example in MATLAB 2: Data set



Sequential clustering algorithms

Example in MATLAB 2 (cont.):

c. For each of the previously defined values of Θ, run the BSAS algorithm $n_{times} = 10$, so that the data vectors are presented with different ordering to BSAS in each run. From the n_{times} estimates of the number of clusters, select the most frequently met value, m_Θ, as the most accurate. Let $\boldsymbol{m}_{tot}$ be the n_Θ-dimensional vector, which contains the m_Θ values.

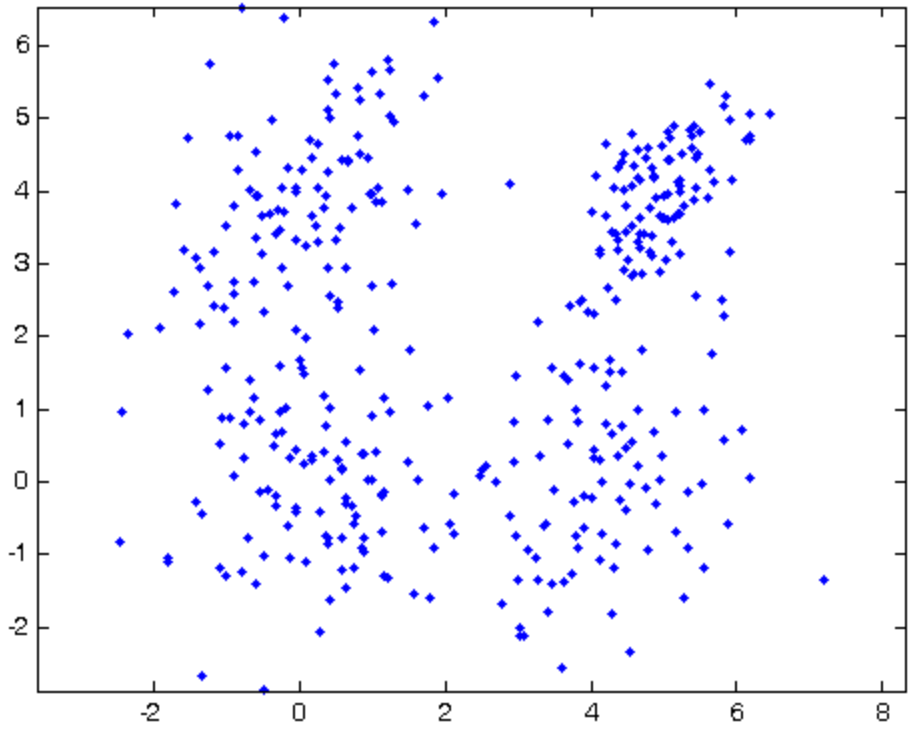
d. Plot m_Θ versus Θ. Determine the widest flat region, r, of Θ's (excluding the one that corresponds to the single-cluster case) and let n_r be the number of Θ's in $\{\Theta_{min}, \Theta_{min} + s, \ldots, \Theta_{max}\}$ that also lie in r. If n_r is "significant" (e.g., greater than 10% of n_Θ), the corresponding number of clusters, m_{best}, is selected as the best estimate and the mean of the values of Θ in r is chosen as the corresponding best value for Θ (Θ_{best}). Otherwise, the single-cluster clustering is adopted.

2. Run the BSAS algorithm for $\Theta = \Theta_{best}$ and plot the data set using different colors and symbols for points from different clusters.

3. Apply the reassignment procedure on the clustering results obtained in the previous step and plot the new clustering.

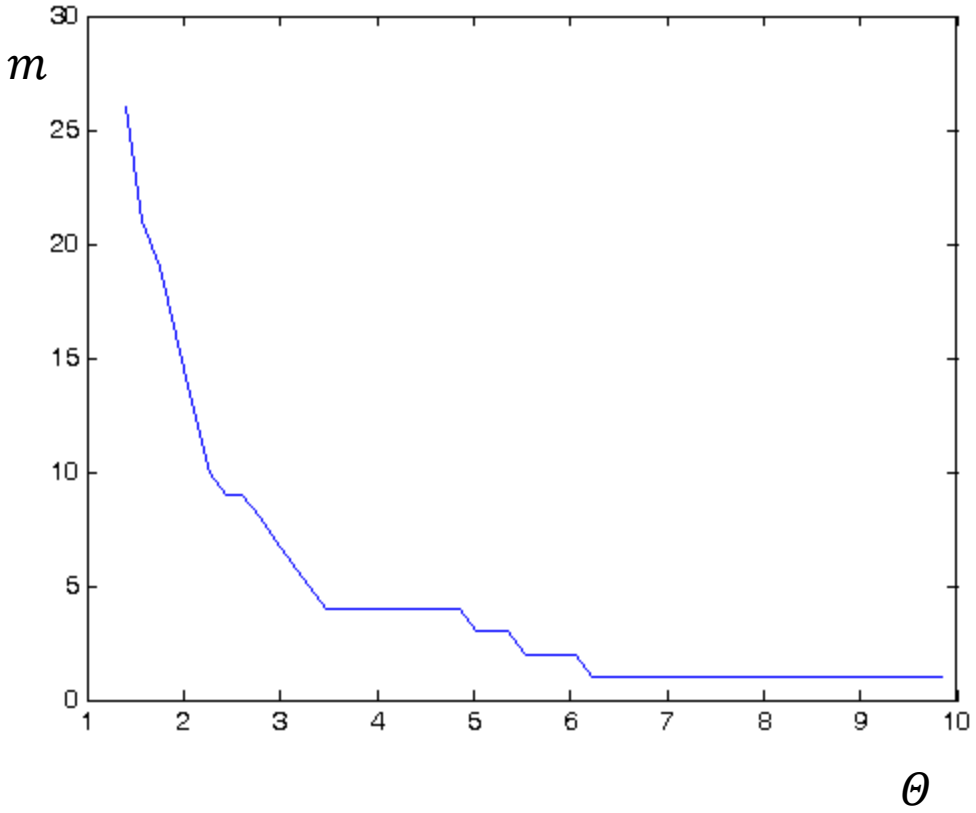
Sequential clustering algorithms

Example in MATLAB 2: Data set



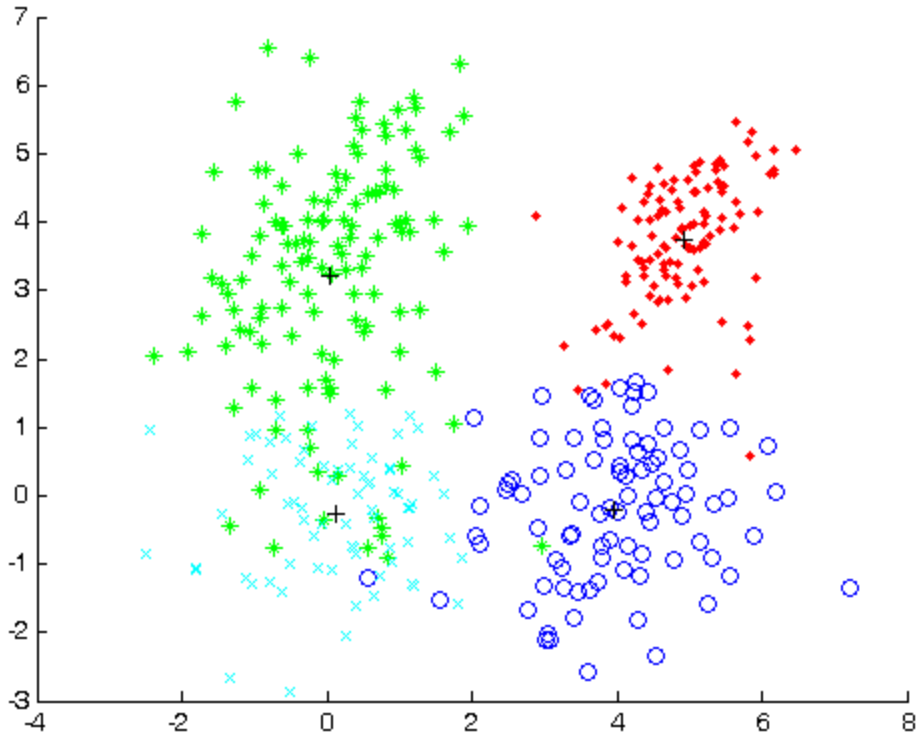
Sequential clustering algorithms

Example in MATLAB 2: Plot m versus Θ



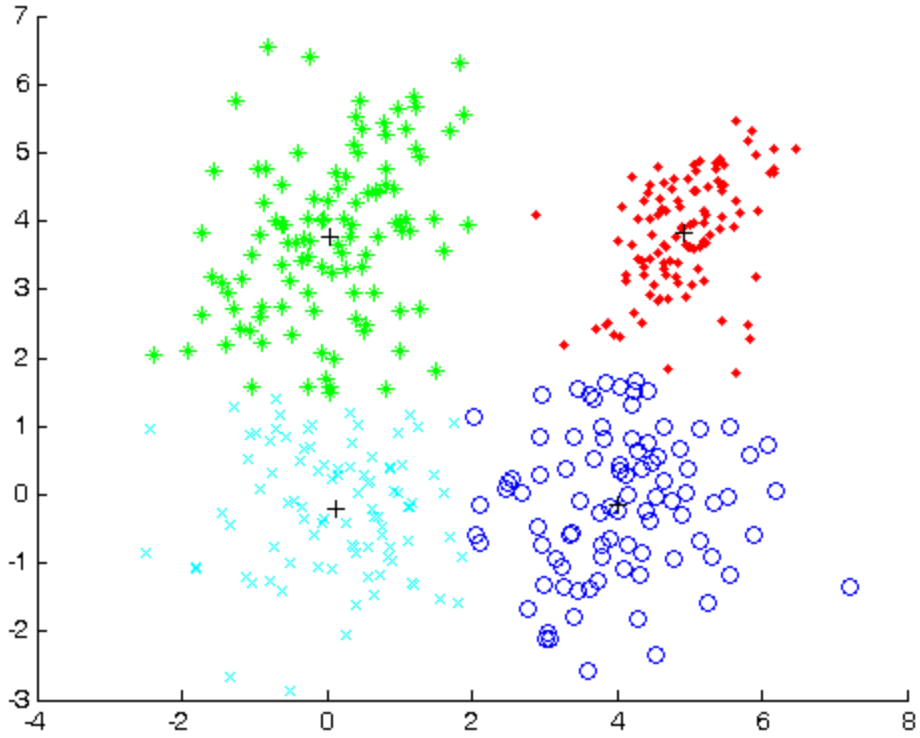
Sequential clustering algorithms

Example in MATLAB 2: BSAS clustering for the best value for Θ



Sequential clustering algorithms

Example in MATLAB 2: BSAS clustering after re-assignment



CFO clustering algorithms: A unified view

Data

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

Basic parameters - notation

✓ $\Theta = \{\boldsymbol{\theta}_j, j = 1, \ldots, m\}$ ($\boldsymbol{\theta}_j$ 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 $P(j|\boldsymbol{x}_i)$

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

- $u_{ij} \in [0,1]$ 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).

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

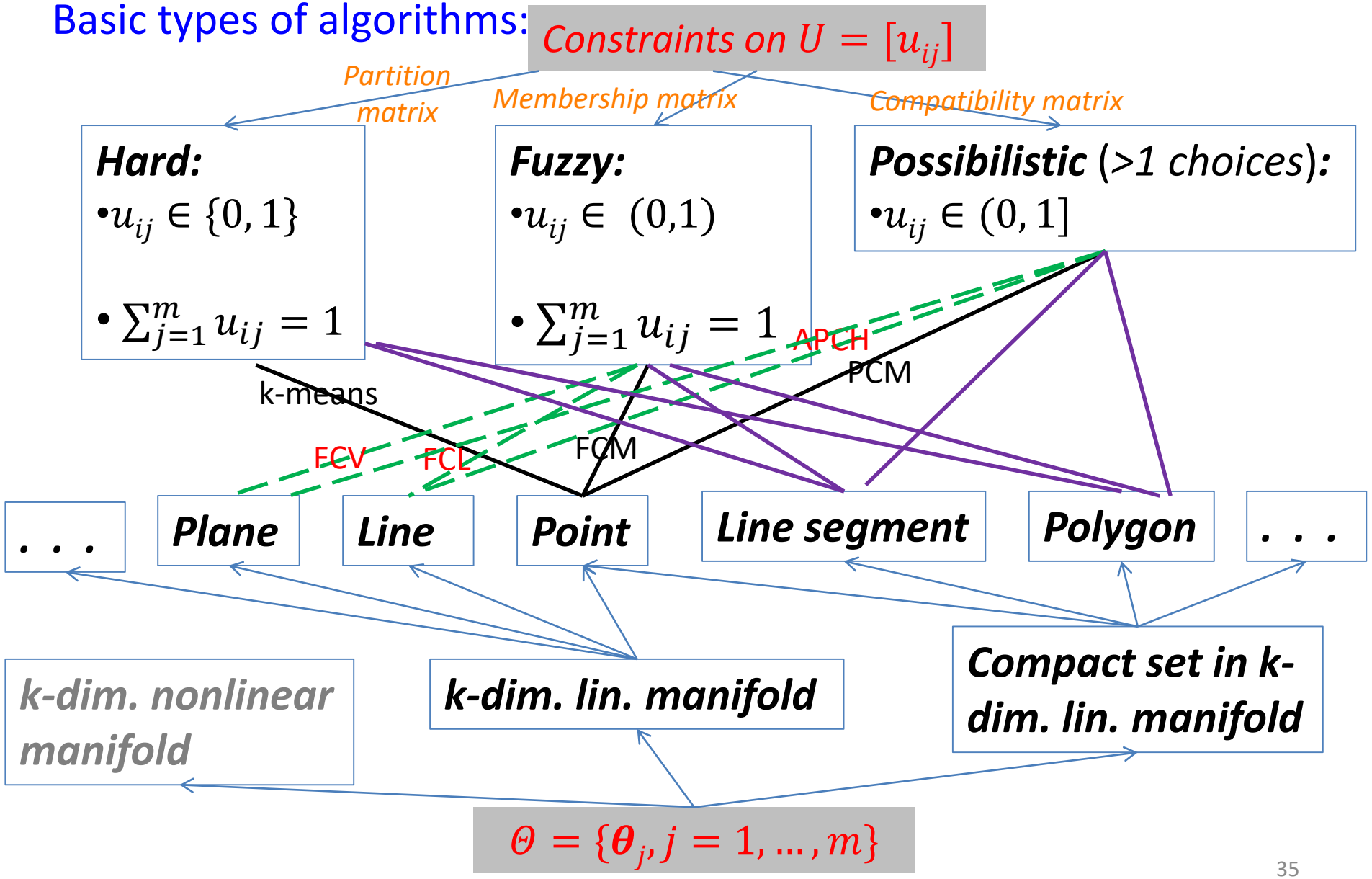
Intuition:

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

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

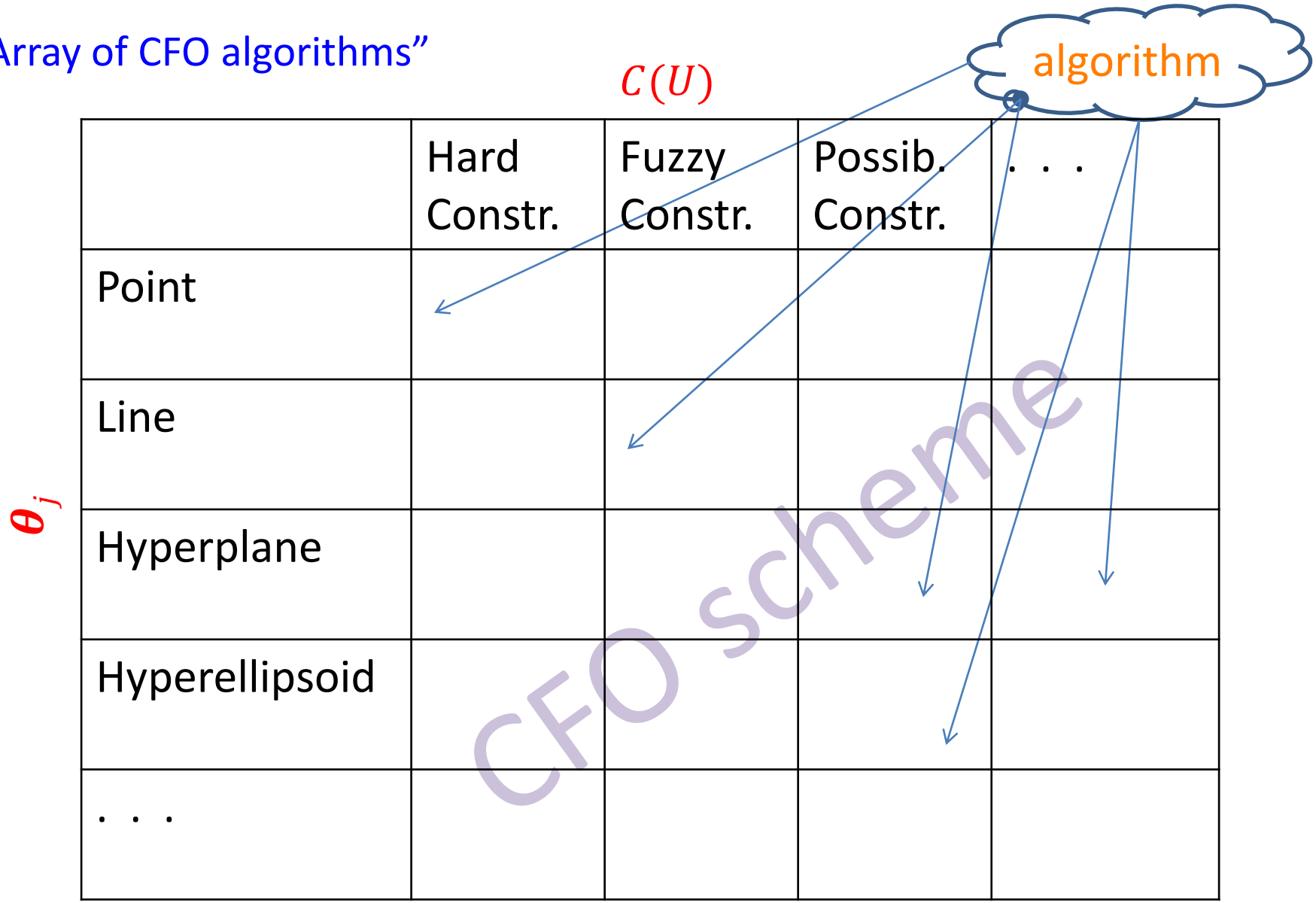
✓ For **fixed** u_{ij}**'s, minimization** of $J(\Theta, U)$ wrt $\boldsymbol{\theta}_j$'s leads $\boldsymbol{\theta}_j$'s closer to their most relative data points.

CFO clustering algorithms: A unified view



CFO clustering algorithms: A unified view

"Array of CFO algorithms"



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

CFO clustering algorithms: A unified view

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

- ✓ *Initialize* $\Theta = \Theta(0)$
- ✓ $t = 0$
- ✓ **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

"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"

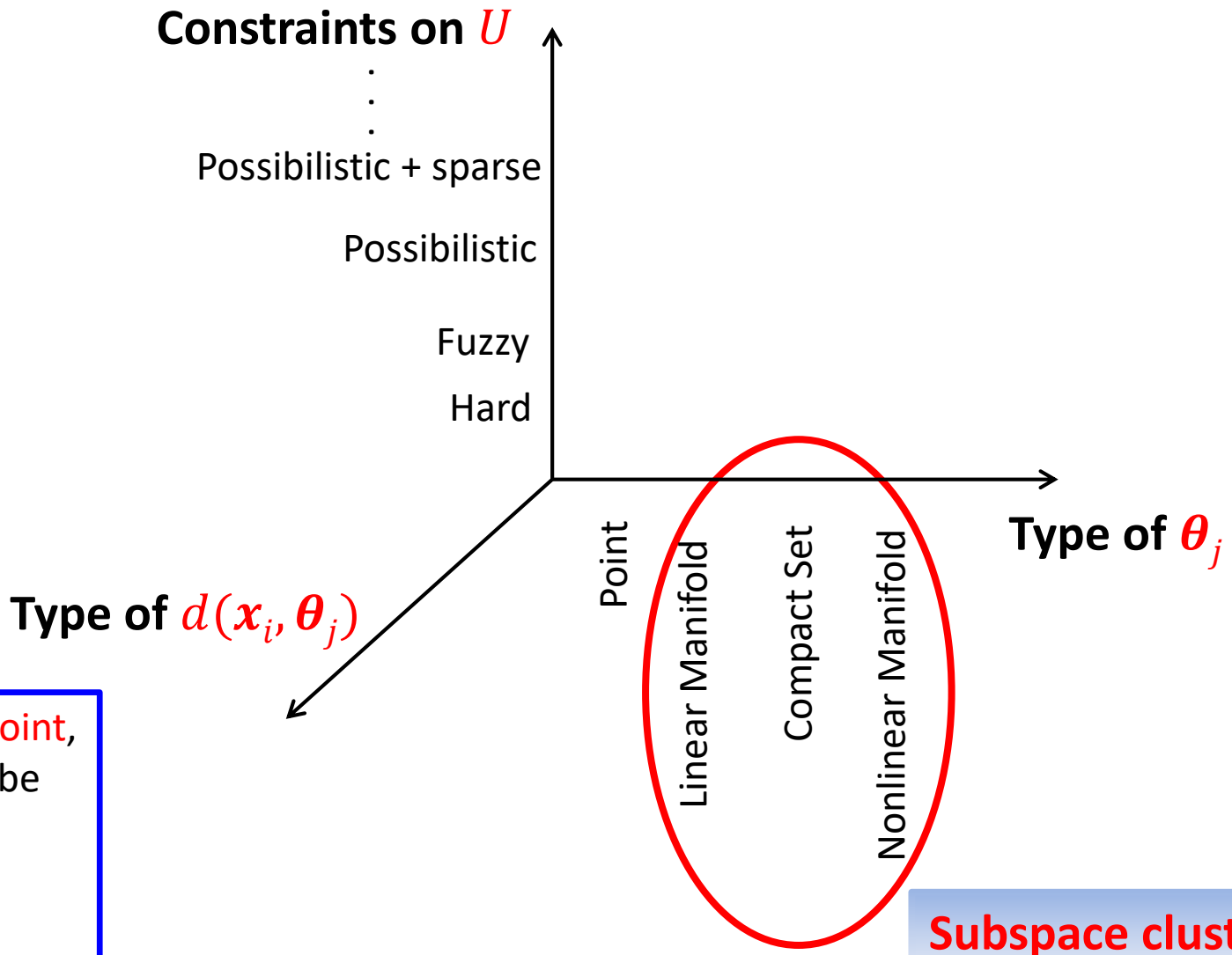
$C(U)$

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

Subspace clustering

CFO clustering algorithms: A unified view

"Array of CFO algorithms"

$C(U)$

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

Hard CFO scheme

CFO scheme

Cost function optimization (CFO) algorithms

Hard clustering algorithms:

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

Each vector belongs exclusively to a single cluster.

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 $u_{ij} = \begin{cases} 1, & if\ \boldsymbol{x}_i \in C_j \\ 0, & otherwise \end{cases}$ and $U = [u_{ij}]_{Nxm}$

It is $\sum_{j=1}^{m} u_{ij} = 1\,, i = 1, \dots, N$

Define the cost function

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

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

CFO hard clustering algorithms

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

For **fixed $\boldsymbol{\theta}_j$'s:** When, *for each $\boldsymbol{x}_i$*, **only** *its distance from its closest representative is taken into account*.

This suggests to **define** $u_{ij}=\begin{cases}1, & if\ d(\boldsymbol{x_i},\boldsymbol{\theta}_j)=min_{q=1,\ldots,m}d(\boldsymbol{x_i},\boldsymbol{\theta}_q)\\ 0, & otherwise\end{cases}$

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

$$min_{\boldsymbol{\theta}_j}\sum_{\boldsymbol{x}_i\in C_j}d(\boldsymbol{x}_i,\boldsymbol{\theta}_j)\equiv min_{\boldsymbol{\theta}_j}\sum_{i=1}^{N}u_{ij}\,d\left(\boldsymbol{x}_i,\boldsymbol{\theta}_j\right)$$

Thus, the Generalized Hard Algorithmic Scheme (GHAS) is given below

CFO hard clustering algorithms

Generalized Hard Algorithmic Scheme (GHAS)

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j, j = 1, \ldots, m$.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of the partition*
    - For $j = 1$ to m

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

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

  - $t = t + 1$

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

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

  - End {For-j}

- **Until** a termination criterion is met.

CFO hard clustering algorithms

Generalized Hard Algorithmic Scheme (GHAS)

Remarks:

- In the update of each $\boldsymbol{\theta}_j$, only the vectors $\boldsymbol{x}_i$ for which $u_{ij}(t-1)=1$ are used.

- GHAS may **terminate** when either
  - $||\Theta(t)-\Theta(t-1)|| < \varepsilon$ or
  - U remains **unchanged** for two successive iterations.

- The two-step optimization procedure in GHAS does not necessarily lead to a local minimum of $J(U,\Theta)$.