Clustering algorithms

Konstantinos Koutroumbas

Unit 4

– CFO clustering algorithms (k-means)

koutroum@noa.gr

CFO clustering algorithms: A unified view

Data

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

Basic parameters - notation

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

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

CFO clustering algorithms: A unified view

Basic parameters – notation (cont.)

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

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

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

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

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

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

CFO clustering algorithms: A unified view

Aim:

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

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:

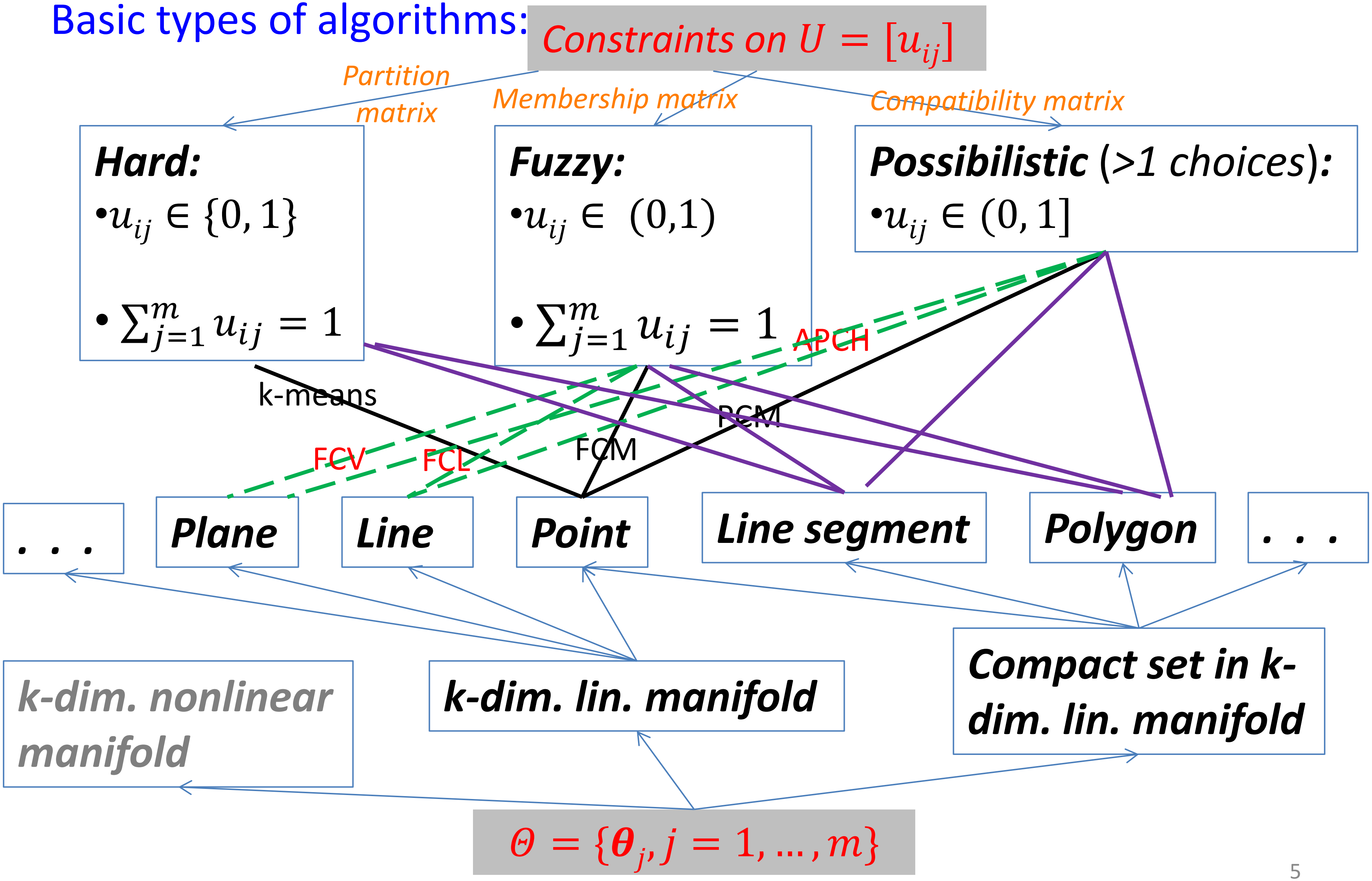
✓ For **fixed** $\boldsymbol{\theta}_j$'s, $J(\Theta, U)$ is a weighted sum of **fixed** distances $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

⇒ **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

Basic types of algorithms:



CFO clustering algorithms: A unified view

"Array of CFO algorithms"

algorithm

$C(U)$

$\boldsymbol{\theta}_j$

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

$C(U)$

$\boldsymbol{\theta}_j$

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

Hard CFO scheme

Fuzzy CFO scheme

Possib. CFO scheme

CFO scheme

CFO clustering algorithms: A unified view

"Array of CFO algorithms"

$C(U)$

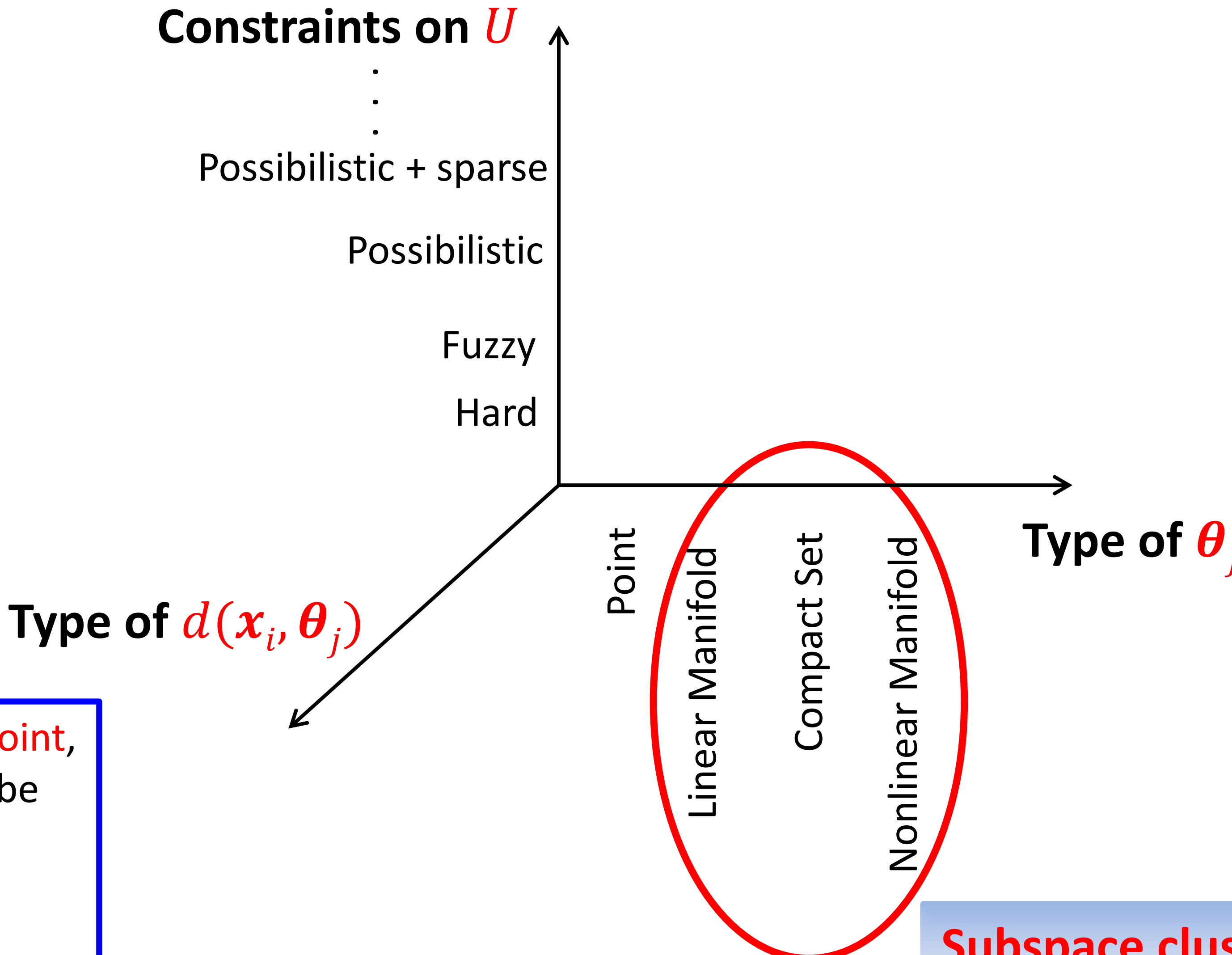
$\boldsymbol{\theta}_j$

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

CFO scheme

CFO clustering algorithms: A unified view

CFO clustering algorithms: A loose presentation



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”

$C(U)$

$\boldsymbol{\theta}_j$

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

Hard CFO scheme

CFO scheme

Cost function optimization (CFO) algorithms

Hard clustering algorithms:

Let $X = \{x_1, x_2, \ldots, x_N\}$ 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 $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_m\}$

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

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

Define the cost function

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

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

CFO hard clustering algorithms

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

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** $u_{ij} = \begin{cases} 1, & if\ d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = min_{q=1,\dots,m} d(\boldsymbol{x}_i, \boldsymbol{\theta}_q) \\ 0, & otherwise \end{cases}$

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

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

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

CFO hard clustering algorithms

<u>*Generalized Hard Algorithmic Scheme (GHAS)*</u>

- **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 the partition*

  o For $j = 1$ to m

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

  o End {For-j}

– End {For-i}

$-t = t + 1$

– For $j = 1$ to m *% Parameter updating*

  o Set

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

– End {For-j}

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

CFO hard clustering algorithms

<u>Generalized Hard Algorithmic Scheme (GHAS)</u>

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 $J(U,\Theta)$.

CFO hard clustering algorithms

<u>*Generalized Hard Algorithmic Scheme (GHAS)*</u>

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 $J(U,\Theta)$ becomes now

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

- 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 $J(U,\Theta)$.

CFO hard clustering algorithms

<u>Generalized Hard Algorithmic Scheme (GHAS)</u>

The Isodata or k-Means or c-Means algorithm

- Choose arbitrary initial estimates $\boldsymbol{\theta}_j(0)$ for the $\boldsymbol{\theta}_j$' s, j=$1,\ldots,m$.
- $t = 0$
- **Repeat**

  – For $i = 1$ to N *% Determination of the partition*

  o For j=1 to m

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

  o End {For-j}

  – End {For-i}

  – $t = t + 1$

  – For $j = 1$ to m *% Parameter updating*

  o Set

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

  – End {For-j}

- **Until** no change in $\boldsymbol{\theta}_j$' s occurs between two successive iterations

CFO hard clustering algorithms

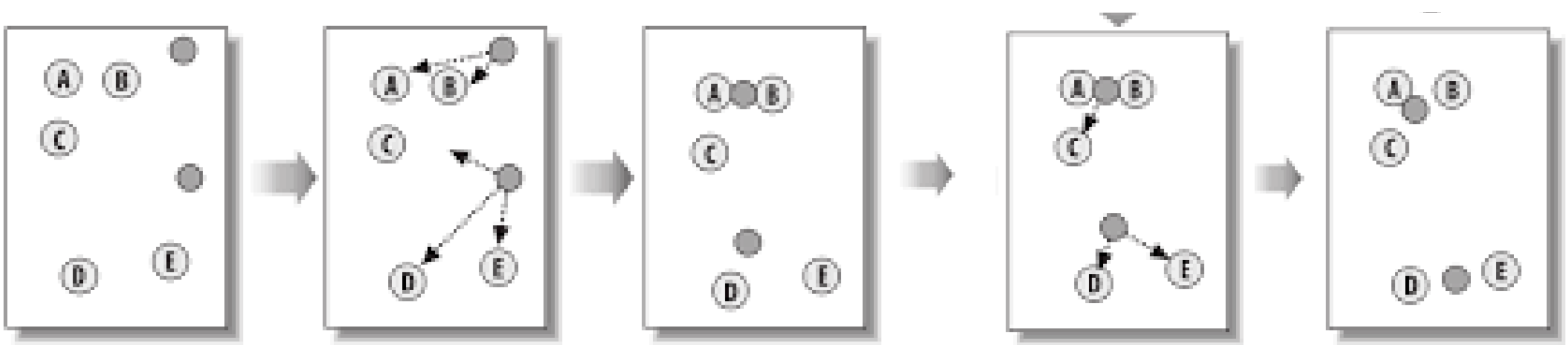
The k-means case.

Choose arbitrary initial estimates $\boldsymbol{\theta}_j(0)$ for the $\boldsymbol{\theta}_j$' s, $j = 1, \dots, m$.

Repeat

- For $i = 1$ to N *Partition determination*
  - o Determine the closest representative, say $\boldsymbol{\theta}_j$, for $\boldsymbol{x}_i$
  - o Set $u_{ij} = 1$ and $u_{iq} = 0$, $q = 1, \dots, m$, $q \neq j$.
- End {For}
- For $j = 1$ to m *Parameter updating*
  - o Determine $\boldsymbol{\theta}_j$ as the mean of the vectors $\boldsymbol{x}_i \in X$ with $u_{ij} = 1$.
- End {For}

Until no change in $\boldsymbol{\theta}_j$' s occurs between two successive iterations



CFO hard clustering algorithms

Remarks

- It is a batch, single clustering algorithm
- It is a hard clustering algorithm that uses point representatives $\boldsymbol{\theta}_j$ for the clusters C_j.
- It results from the optimization of the following cost function

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

where $U = [u_{ij}]$ and $\Theta = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m\}$

- It is of iterative nature.
- Initially it places the representatives $\boldsymbol{\theta}_j$ at random positions in space.
- It gradually moves the representatives towards the centers of the true clusters.
- In practice, its time complexity is $\underline{O(q \cdot m \cdot N)}$ ($\underline{q}$ is the number of iterations).
- It requires the number of clusters m to be known a priori.

CFO hard clustering algorithms

<u>*Generalized Hard Algorithmic Scheme (GHAS)*</u>

The Isodata or k-Means or c-Means algorithm

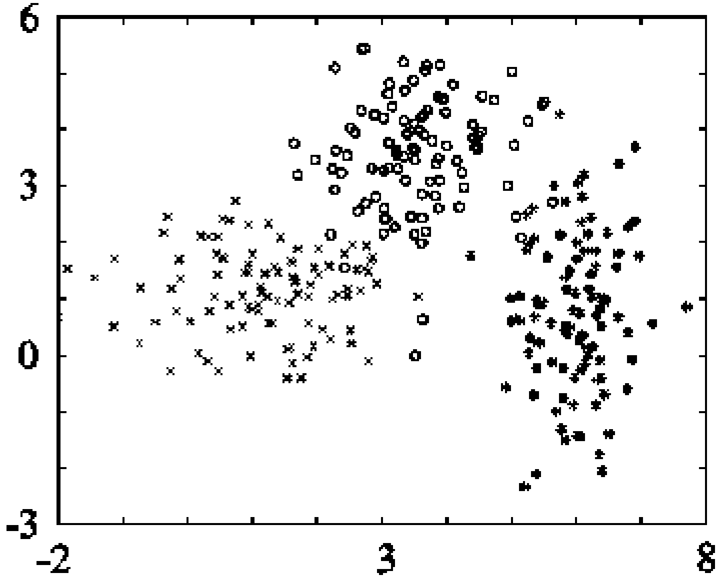
Example 1: (a) Consider three two-dimensional normal distributions with mean values:

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

and respective covariance matrices

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

Generate a group of 100 vectors from each distribution. These form the data set X.



Confusion matrix for the results of k-means.

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

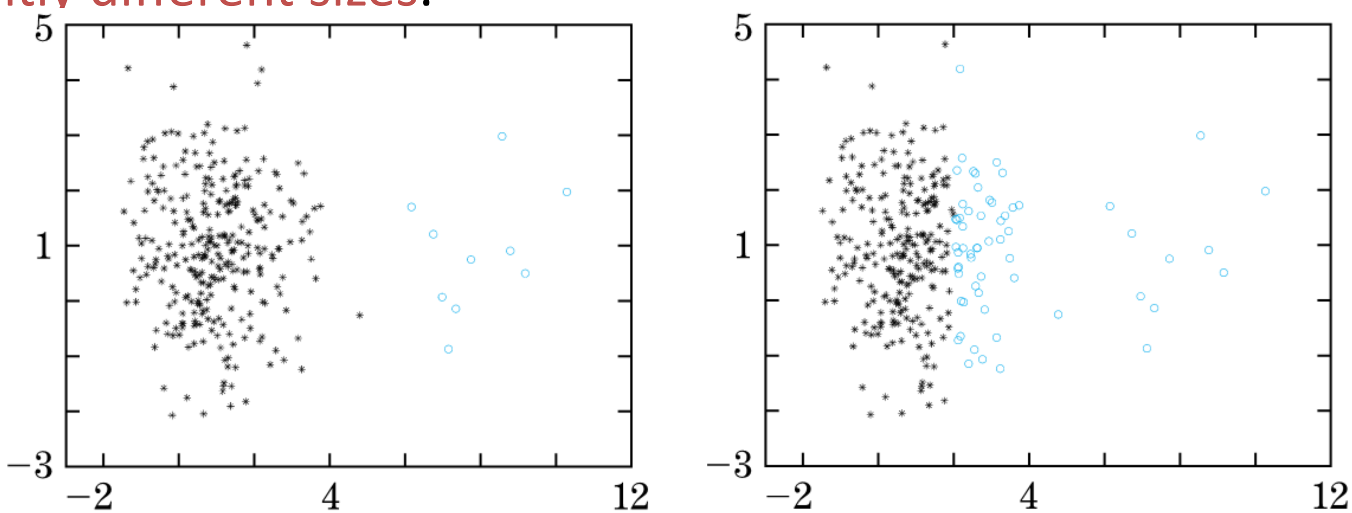
CFO hard clustering algorithms

<u>*Generalized Hard Algorithmic Scheme (GHAS)*</u>

The Isodata or k-Means or c-Means algorithm

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

- After convergence <u>*the large group has been split into two clusters*</u>.
- 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.



CFO hard clustering algorithms

<u>*Generalized Hard Algorithmic Scheme (GHAS)*</u>

The Isodata or k-Means or c-Means algorithm

Remarks:

- k-means recovers compact clusters.
- 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.
- Sequential (online) 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 $\boldsymbol{x}_i$, have also been proposed.
- A variant of the k-means results if the number of vectors in each cluster is constrained *a priori*.

Further remarks:

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

CFO hard clustering algorithms

Generalized Hard Algorithmic Scheme (GHAS)

The Isodata or k-Means or c-Means algorithm

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 $\boldsymbol{\theta}_j$'s.
  - Partition randomly the data set into m subsets and use their means as initial estimates for $\boldsymbol{\theta}_j$' s.
- *Multiple run methods*
  - Create different partitions of X, run k-means for each one of them and select the best result.
- *Utilization of tools from stochastic optimization techniques* (simulated annealing, genetic algorithms etc).

CFO hard clustering algorithms

<u>Generalized Hard Algorithmic Scheme (GHAS)</u>

The Isodata or k-Means or c-Means algorithm

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

Strategies for facing drawback 2:

- Employ splitting, merging and/or discarding operations of the clusters resulting from k-means.

- Estimate m as follows:
  - Run a **sequential** algorithm many times for different thresholds of dissimilarity Θ.
  - Plot Θ 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.

CFO hard clustering algorithms

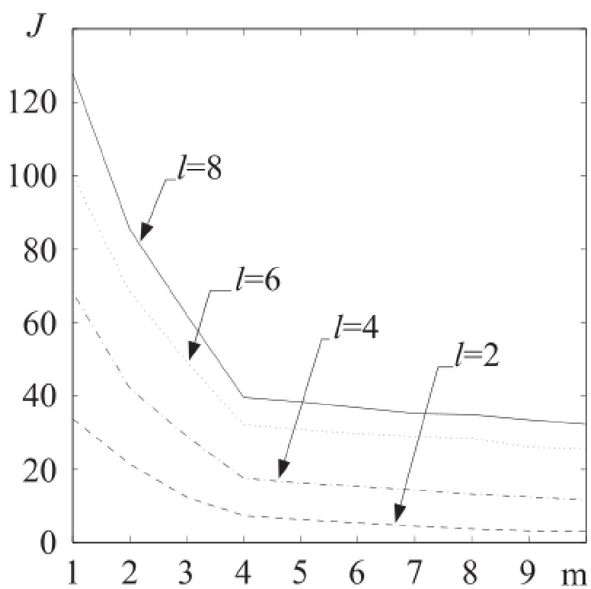
<u>*Generalized Hard Algorithmic Scheme (GHAS)*</u>

The Isodata or k-Means or c-Means algorithm

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

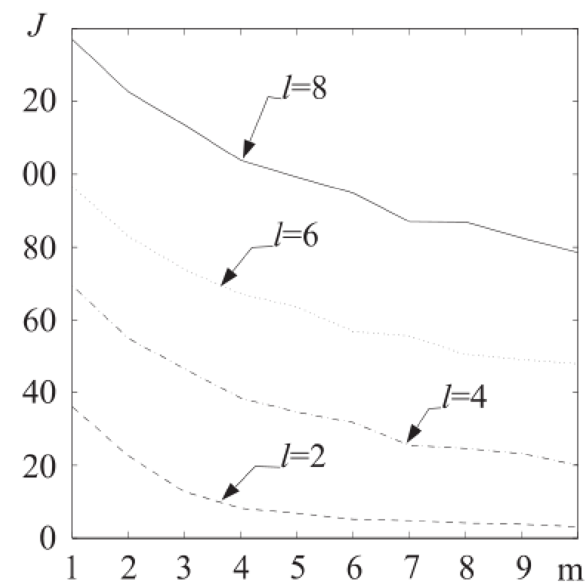
Strategies for facing drawback 2 (cont.):

- Estimate m as follows:
  - Run the $\boldsymbol{k}$**-means** algorithm for different values of the number of clusters m.
  - For each of the resulting clusterings compute the value of J.
  - **Plot** J **versus** the number of clusters m and identify the most significant knee in the graph. Its position indicates the number of physical clusters.



Clustered data

Non-clustered data



CFO hard clustering algorithms

<u>*Generalized Hard Algorithmic Scheme (GHAS)*</u>

The Isodata or k-Means or c-Means algorithm

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.