# Clustering algorithms

Konstantinos Koutroumbas

## Unit 4

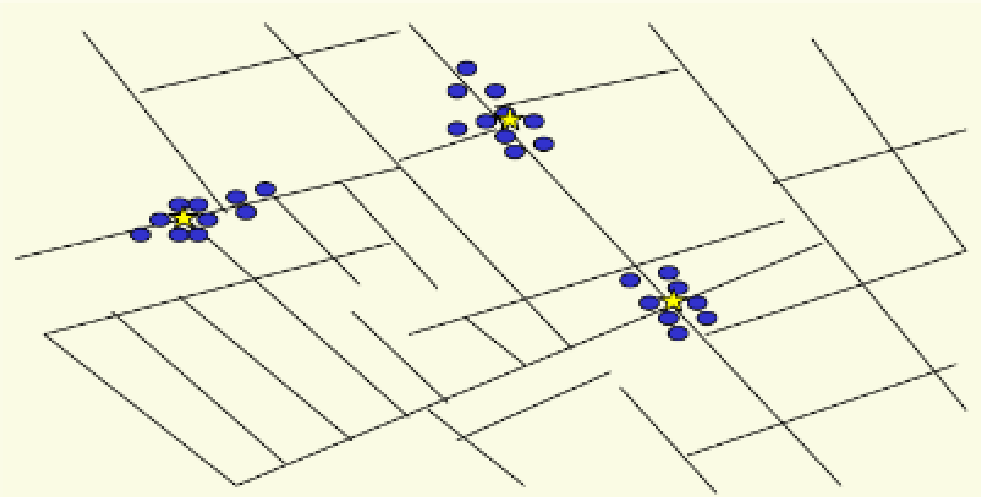
– Cost function optimization clustering algorithms: The hard clustering case

koutroum@noa.gr

# Clustering: A historic example(*)

Dr John Snow plotted the location of cholera deaths on a map during an outbreak at London in the 1850s

The locations were **clustered** around certain intersections where there were polluted wells!!!



**Questions:**

- Which are the entities and how they are represented?
- Which dissimilarity measure could be used?
- What is the form of the resulted clusters?
- What kind of clusters should be able to reveal the adopted clustering criterion?

(*) Nina Mishra HP Labs

# 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:**

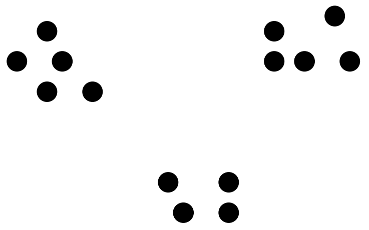
- $S(15,3) = 2{,}375{,}101$
- $S(20,4) = 45{,}232{,}115{,}901$
- $S(25,8) = 690{,}223{,}721{,}118{,}368{,}580$
- $S(100,5) \approx 10^{68}!!$

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

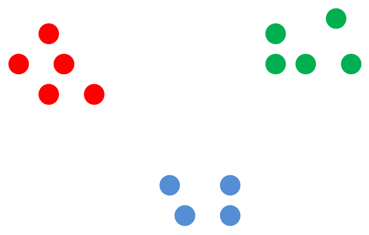
$\Rightarrow$

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

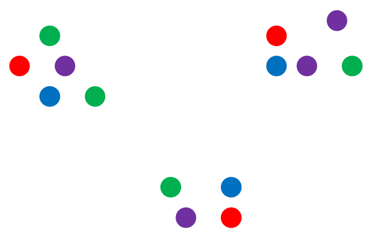
# Clustering algorithms



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 is a learning procedure that 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

⇒ 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, $q$.

- The clusters are **defined** with the aid of
  - ✓ An appropriately defined distance $d\ (x, C)$ of a point from a cluster.
  - ✓ A threshold $\Theta$ associated with the distance.

# Sequential clustering algorithms

**Basic Sequential Clustering Algorithm Scheme (BSAS)**

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

```
•$m = 1$  \{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}
```

-----------------------------------

(*) 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 Algorithm 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 $\Theta$.

- 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

**Example in MATLAB 1:**

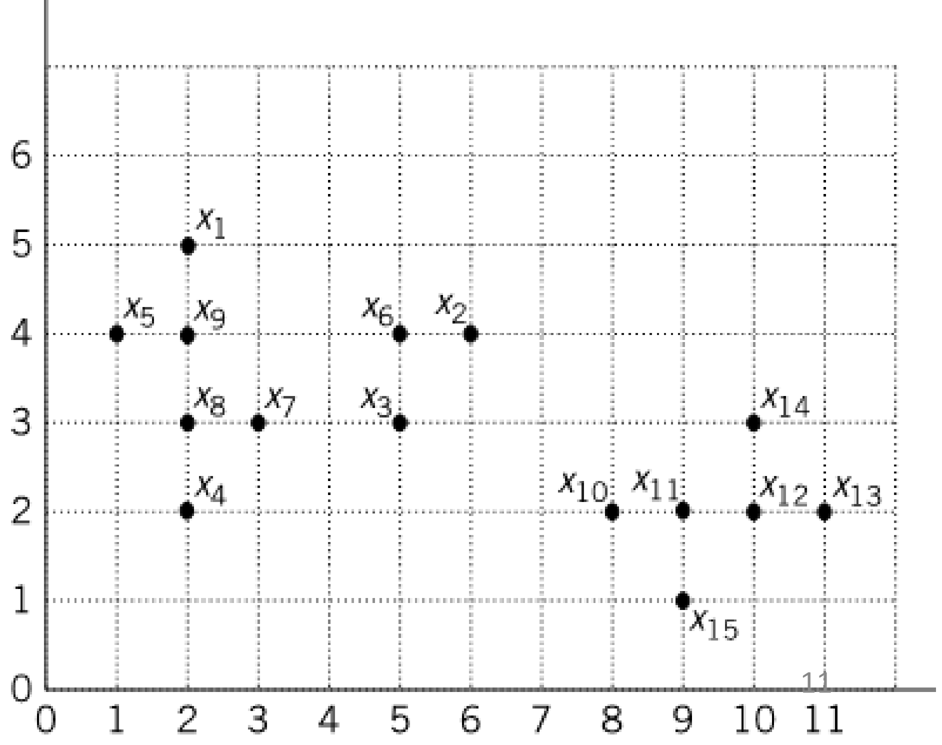
Consider the data vectors depicted in the figure below and perform a “visual” clustering on it.

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 $q = 15$.

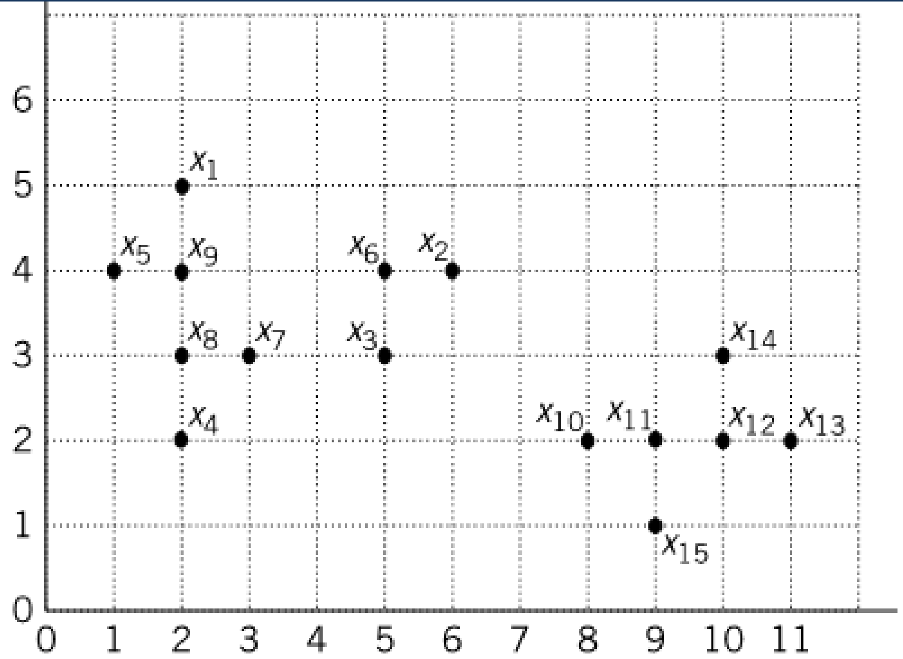
**2.** Repeat step 1, now with the order of presentation to the algorithm as $\boldsymbol{x}_7, \boldsymbol{x}_3, \boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_9, \boldsymbol{x}_6, \boldsymbol{x}_8, \boldsymbol{x}_4, \boldsymbol{x}_2, \boldsymbol{x}_{10}, \boldsymbol{x}_{15}, \boldsymbol{x}_{13},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

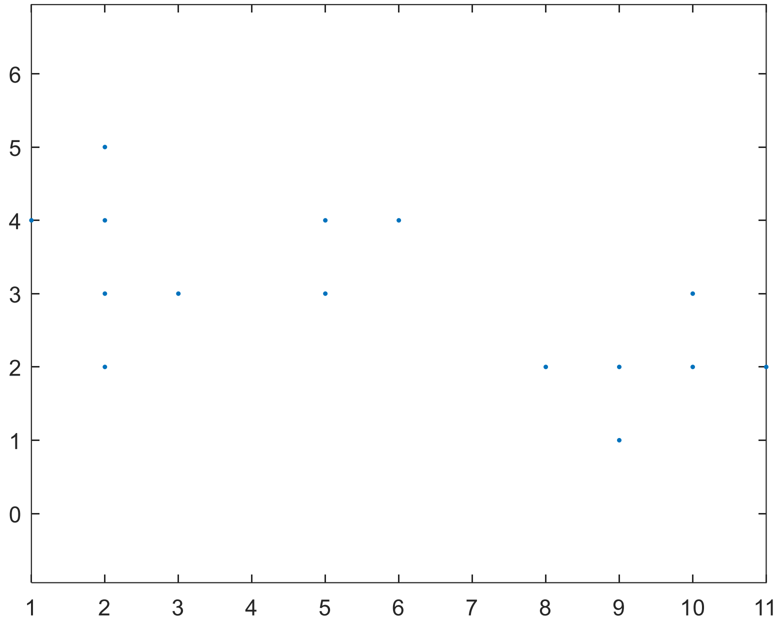
**3.** Repeat step 1, now with $\Theta = 1.4$.

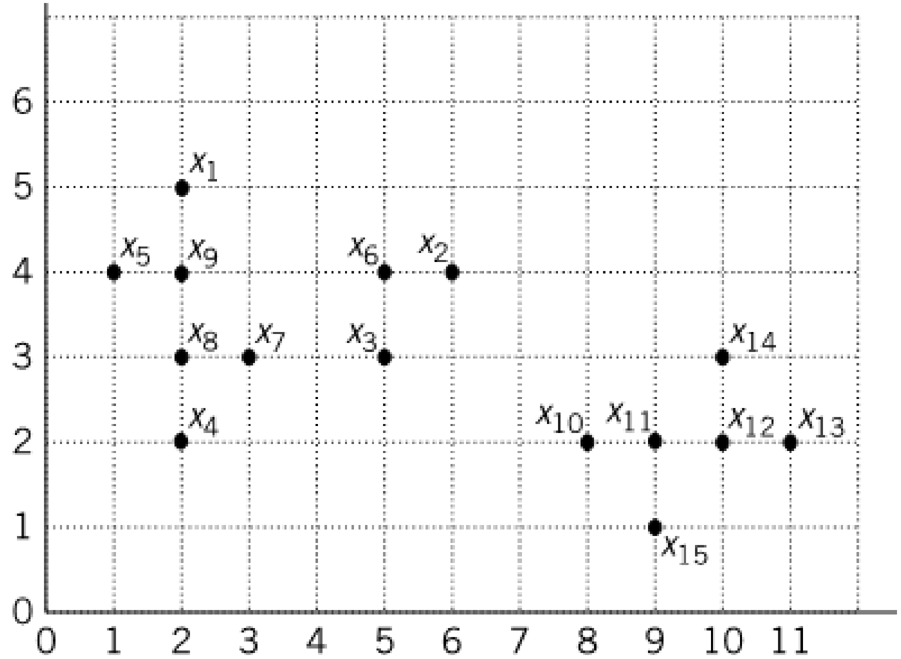
**4.** Repeat step 1, now with $q = 2$.

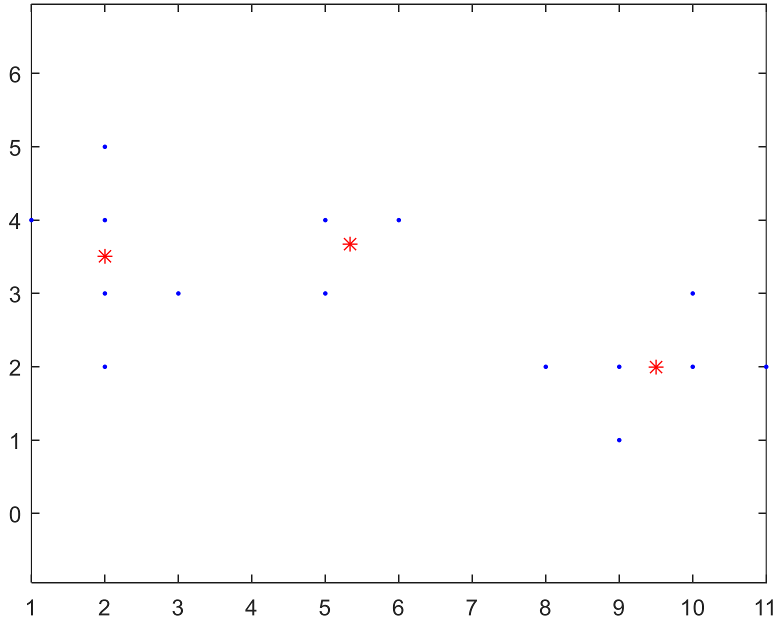


# Sequential clustering algorithms

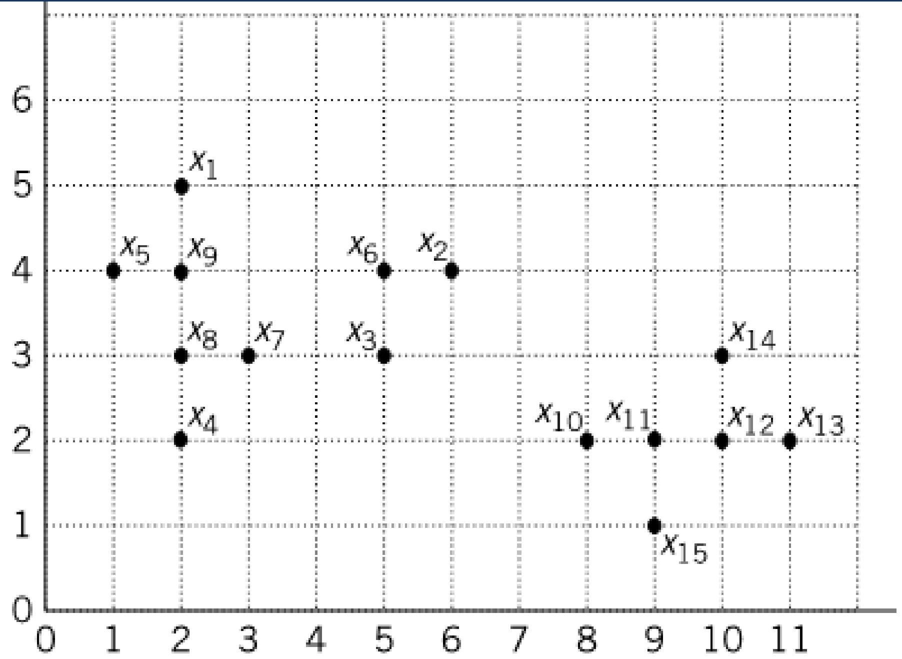


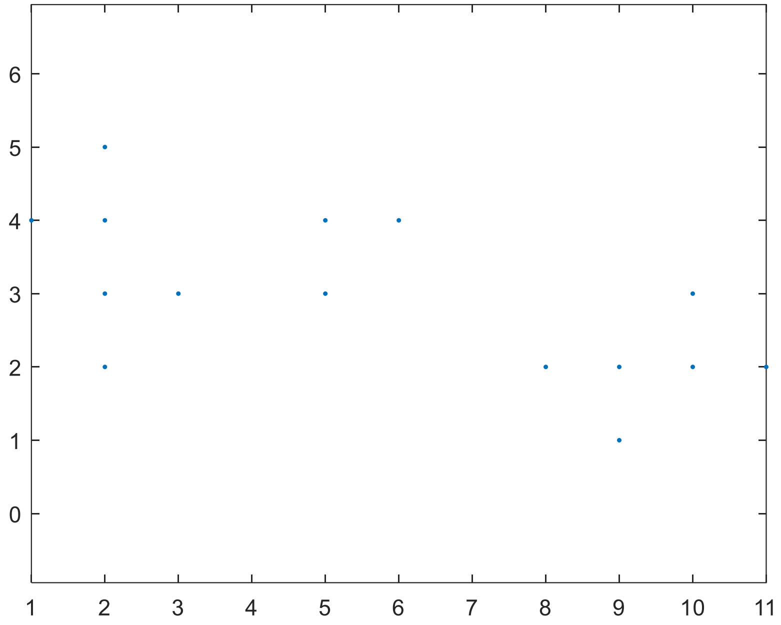


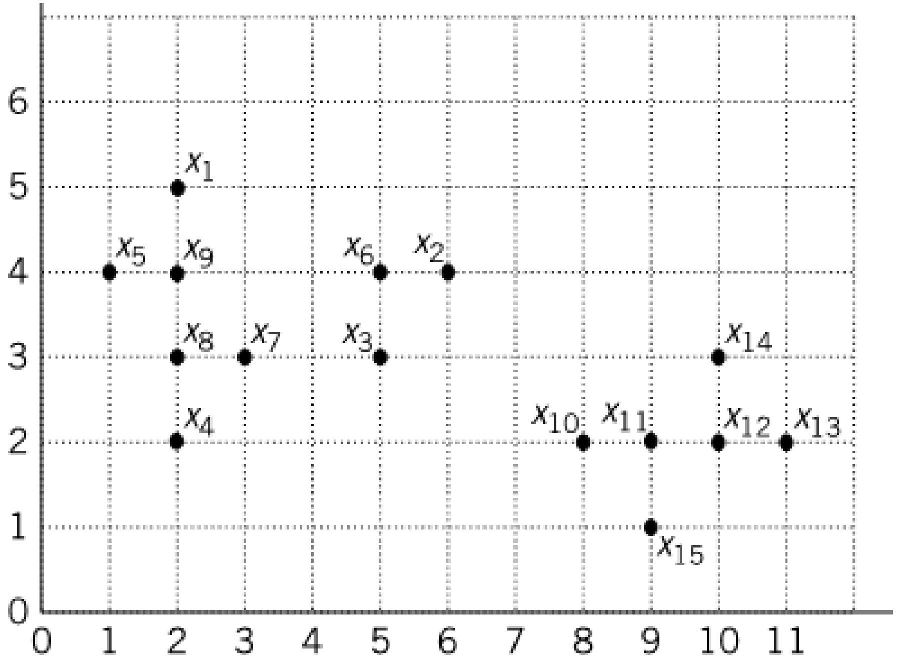


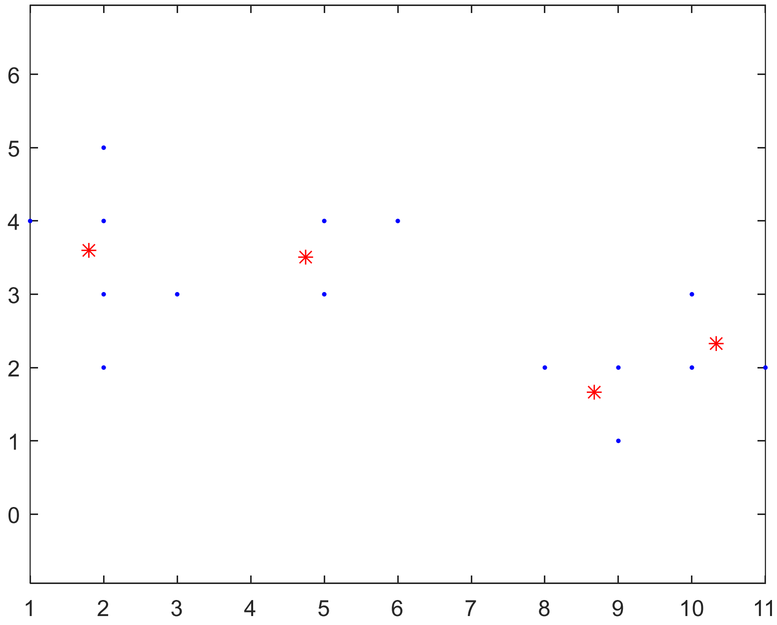


# Sequential clustering algorithms

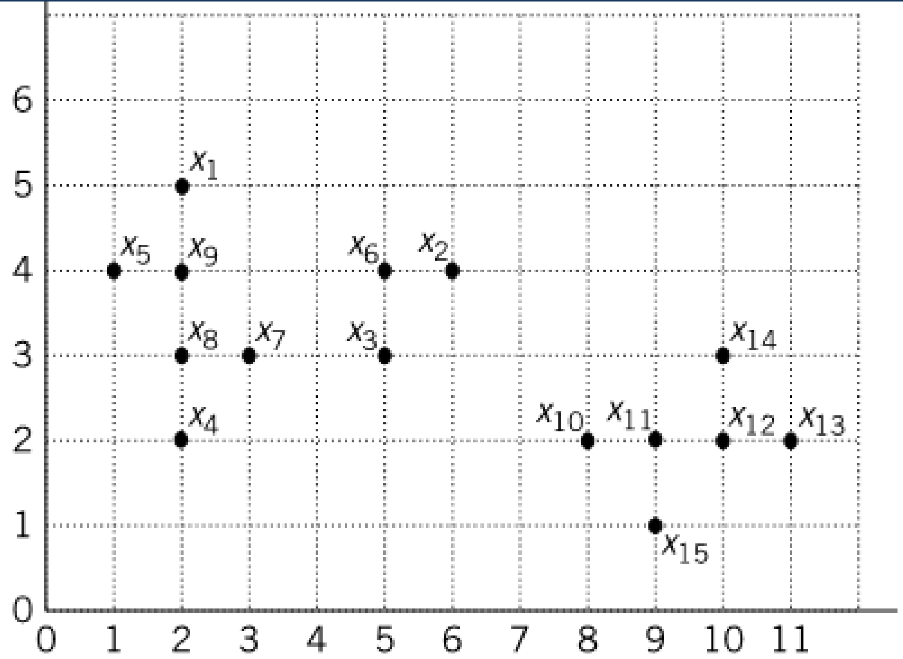


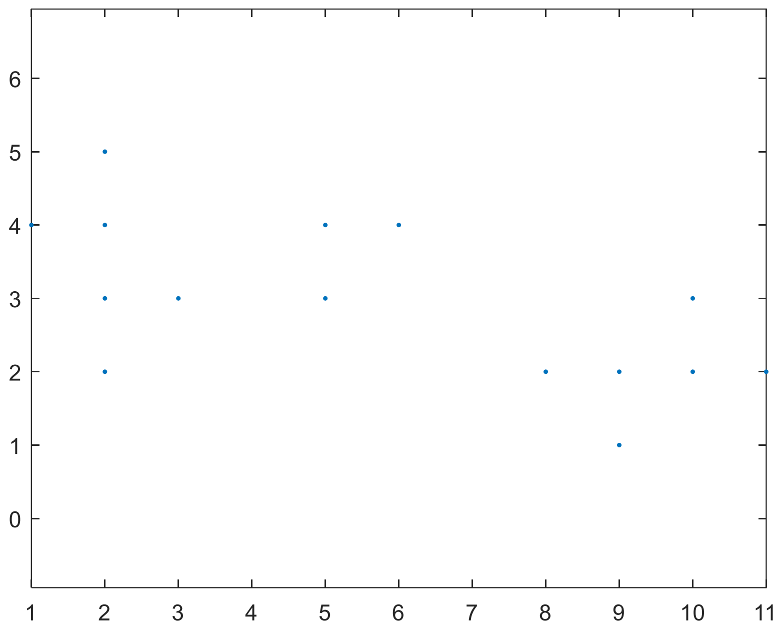


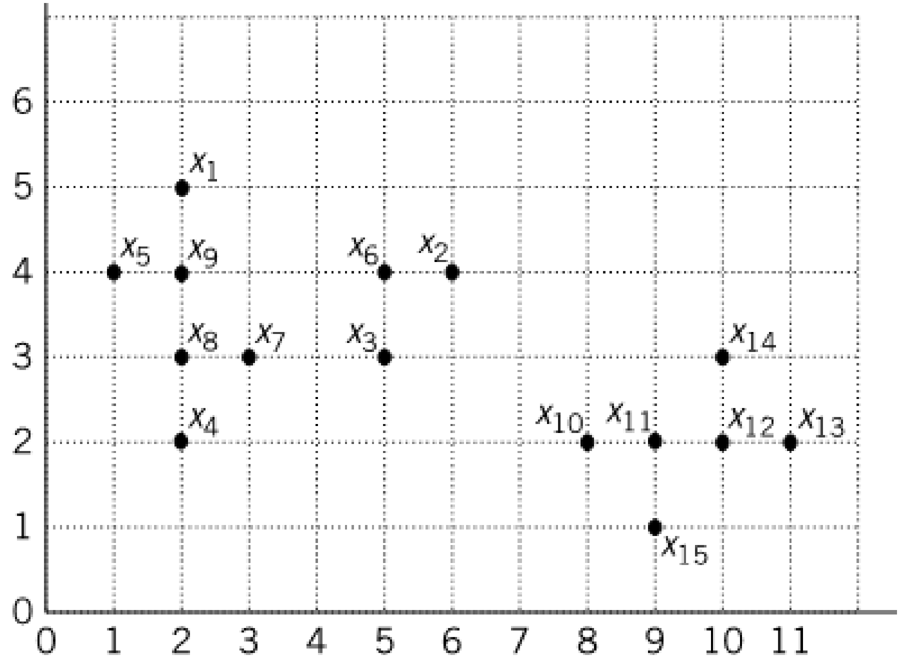


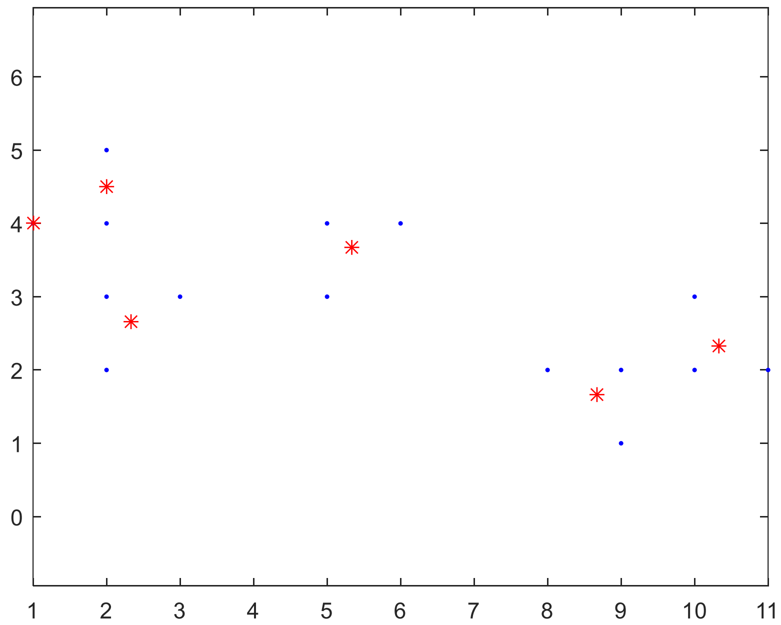


# Sequential clustering algorithms

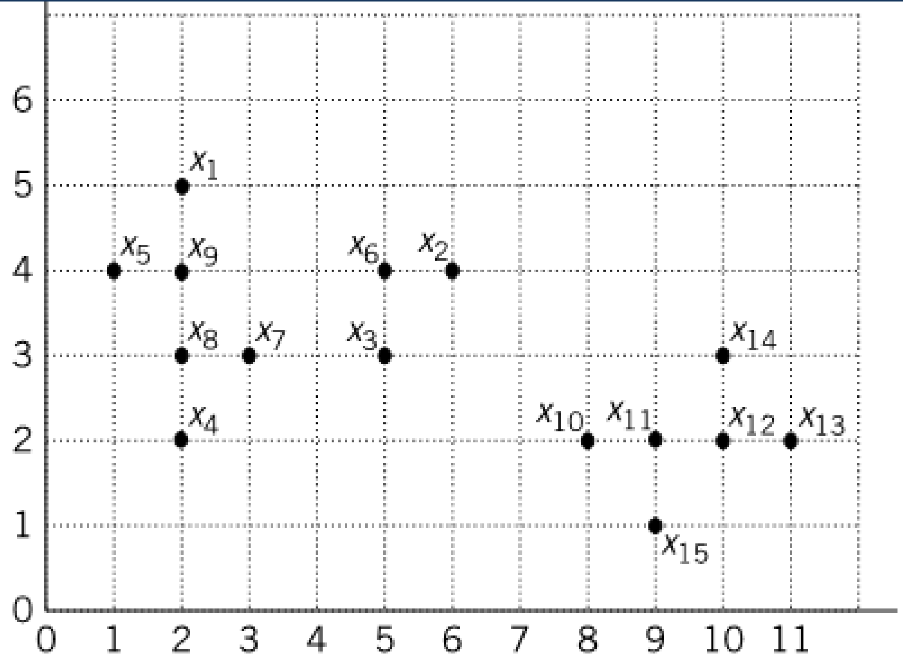


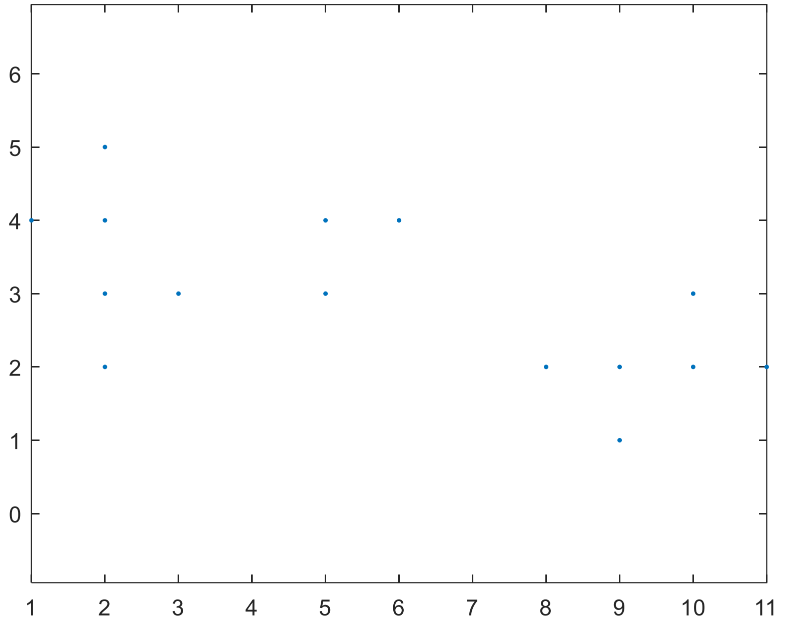


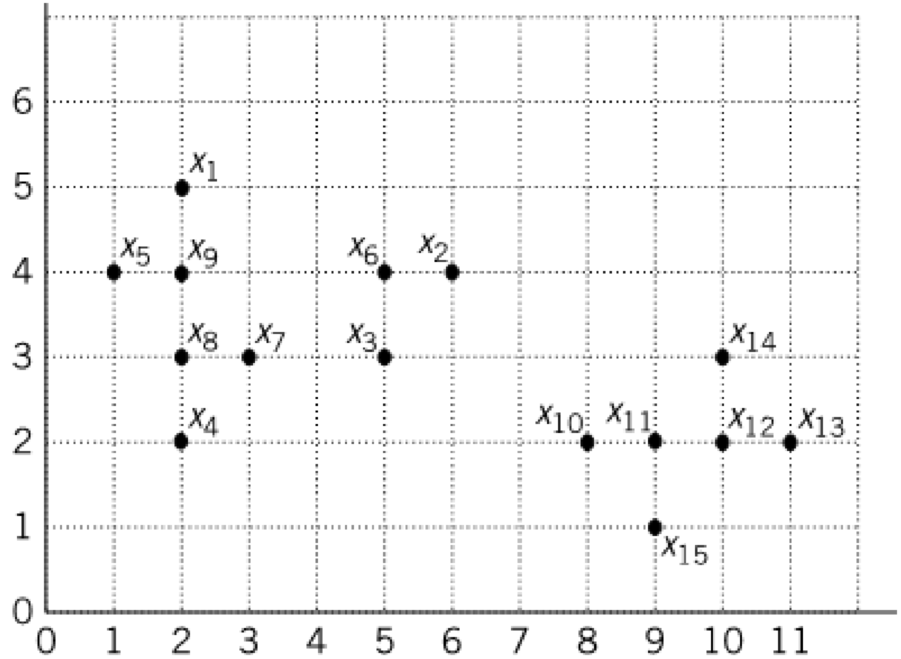


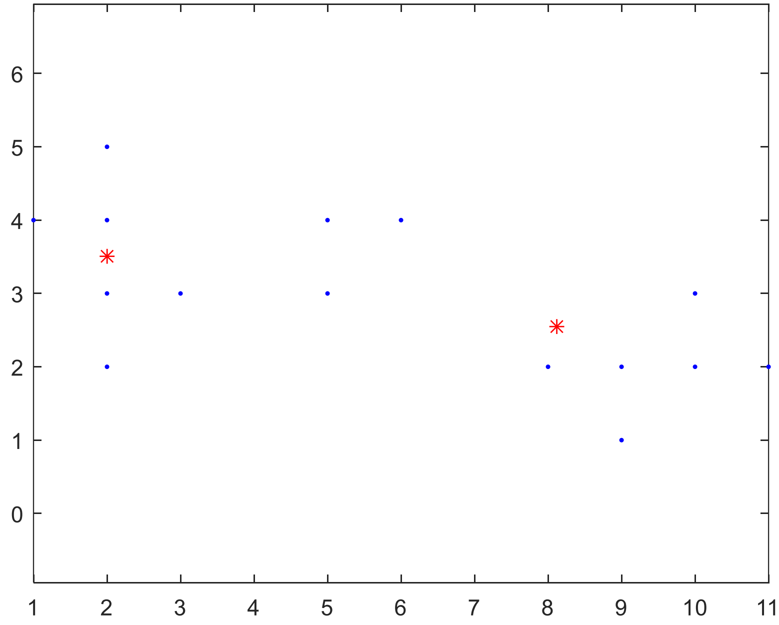


# Sequential clustering algorithms









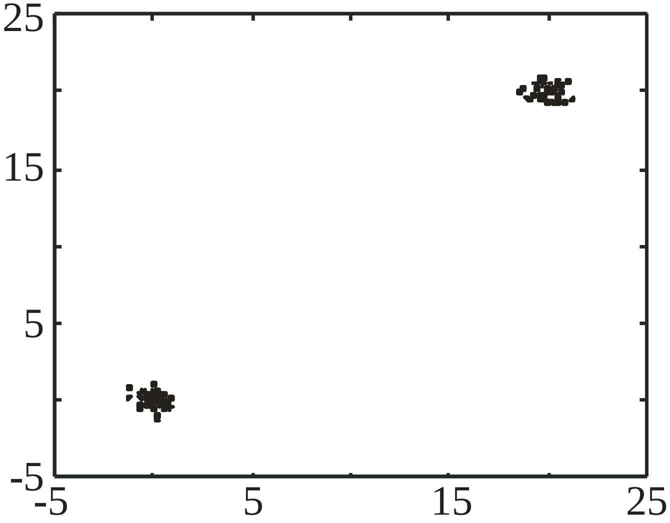
# Sequential clustering algorithms

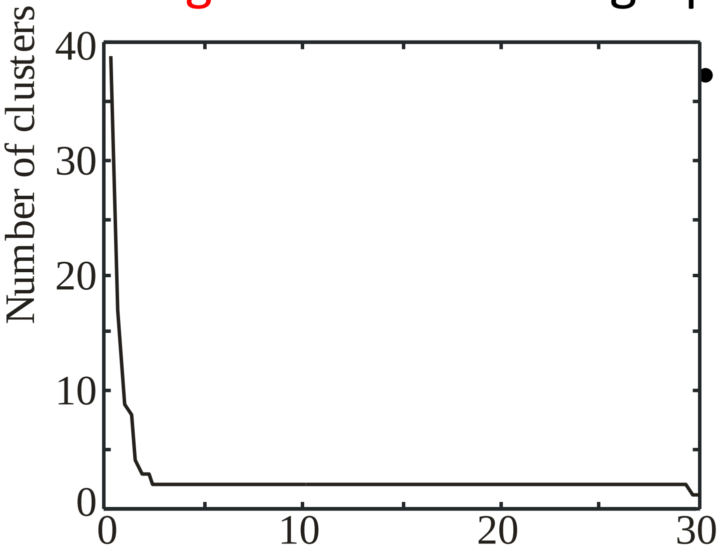
**Basic Sequential Clustering Algorithm Scheme (BSAS)**

***Estimating the number of clusters in the data set:***

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

- 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 $m_\Theta$, as the most frequent number resulting from the $s$ runs of $BSAS(\Theta)$.
- Next $\Theta$
- **Plot** $m_\Theta$ versus $\Theta$ 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 $\Theta$ 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 deals with this issue, at the cost of processing the data twice.

- 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 $\underline{x}_i$ \}
• End {for}

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

# 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 $S_1 = I$, $S_2 = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1.5 \end{bmatrix}$, $S_3 = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1.1 \end{bmatrix}$, $S_2 = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.5 \end{bmatrix}$. 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 $\Theta$ for which the BSAS will run. These may be defined as $\Theta_{min}$, $\Theta_{min} + s$, $\Theta_{min} + 2s$,…,$\Theta_{max}$, where $\Theta_{min} = 0.25\frac{d_{min}+d_{max}}{2}$, $\Theta_{max} = 1.75\frac{d_{min}+d_{max}}{2}$ and $s = \frac{\Theta_{min}+\Theta_{max}}{n_\Theta}$, $n_\Theta$ is the number of successive values of $\Theta$ that will be considered. Use $n_\Theta = 50$.

# Sequential clustering algorithms

**Example in MATLAB 2 (cont.):**

c. For each of the previously defined values of $\Theta$, 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, $m_{\Theta}$, as the most accurate. Let $\boldsymbol{m}_{tot}$ be the $n_{\Theta}$-dimensional vector, which contains the $m_{\Theta}$ values.

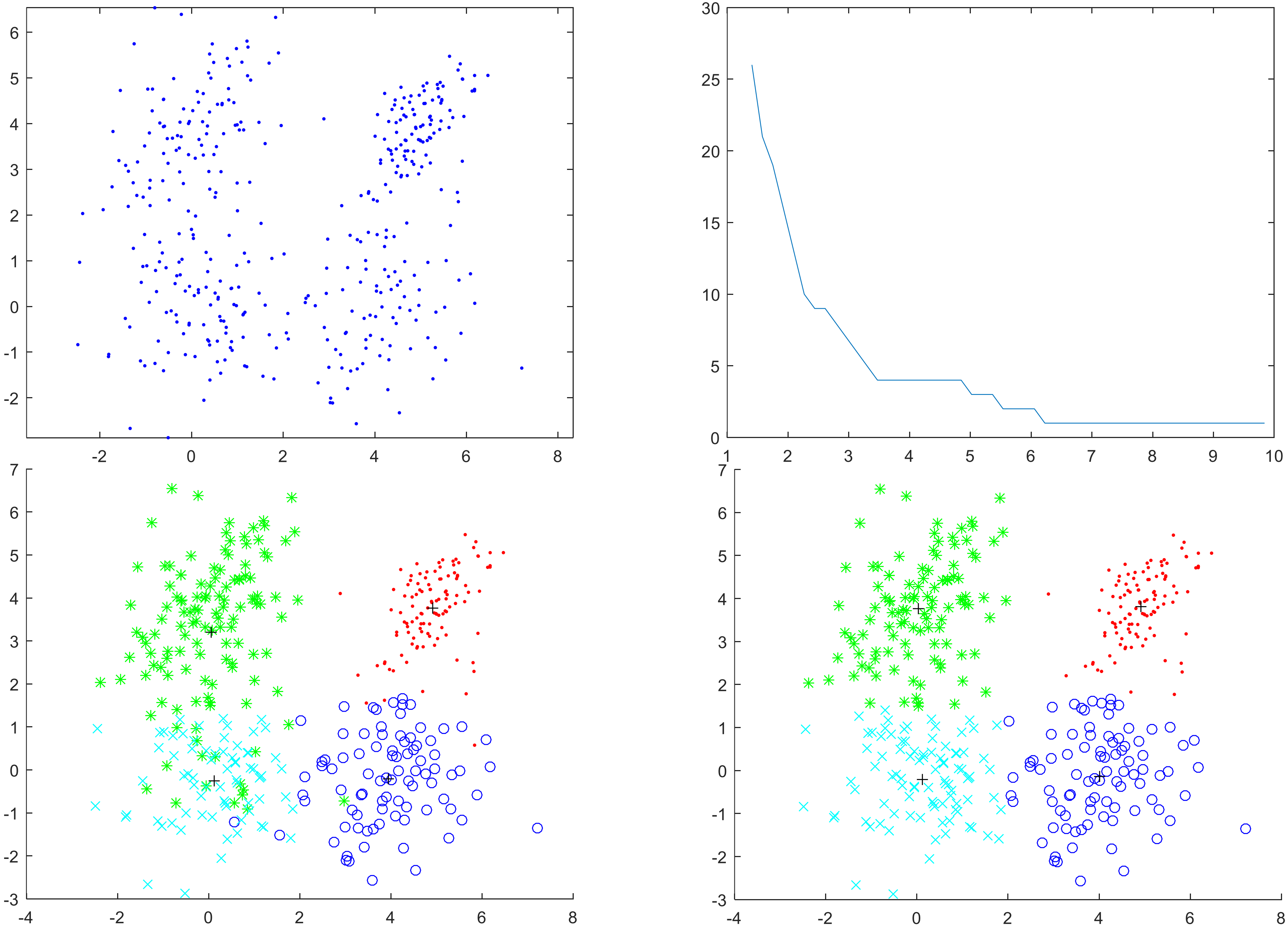
d. Plot $m_{\Theta}$ versus $\Theta$. Determine the widest flat region, $r$, of $\Theta$'s (excluding the one that corresponds to the single-cluster case) and let $n_r$ be the number of $\Theta$'s in $\{\Theta_{min}, \Theta_{min} + s, \dots, \Theta_{max}\}$ that also lie in $r$. If $n_r$ is "significant"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



# 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 $\Theta_1$ and $\Theta_2$ ($\Theta_1 < \Theta_2$) 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 $\Theta_2$ then:
    –A new cluster represented by $\boldsymbol{x}$ is created.
Else if $d(x, C) < \Theta_1$ then
    –$x$ 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**

$W$ 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 ($\Theta$) 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.

# Cost Function Optimization 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 $P(j|\boldsymbol{x}_i)$

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

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

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

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:

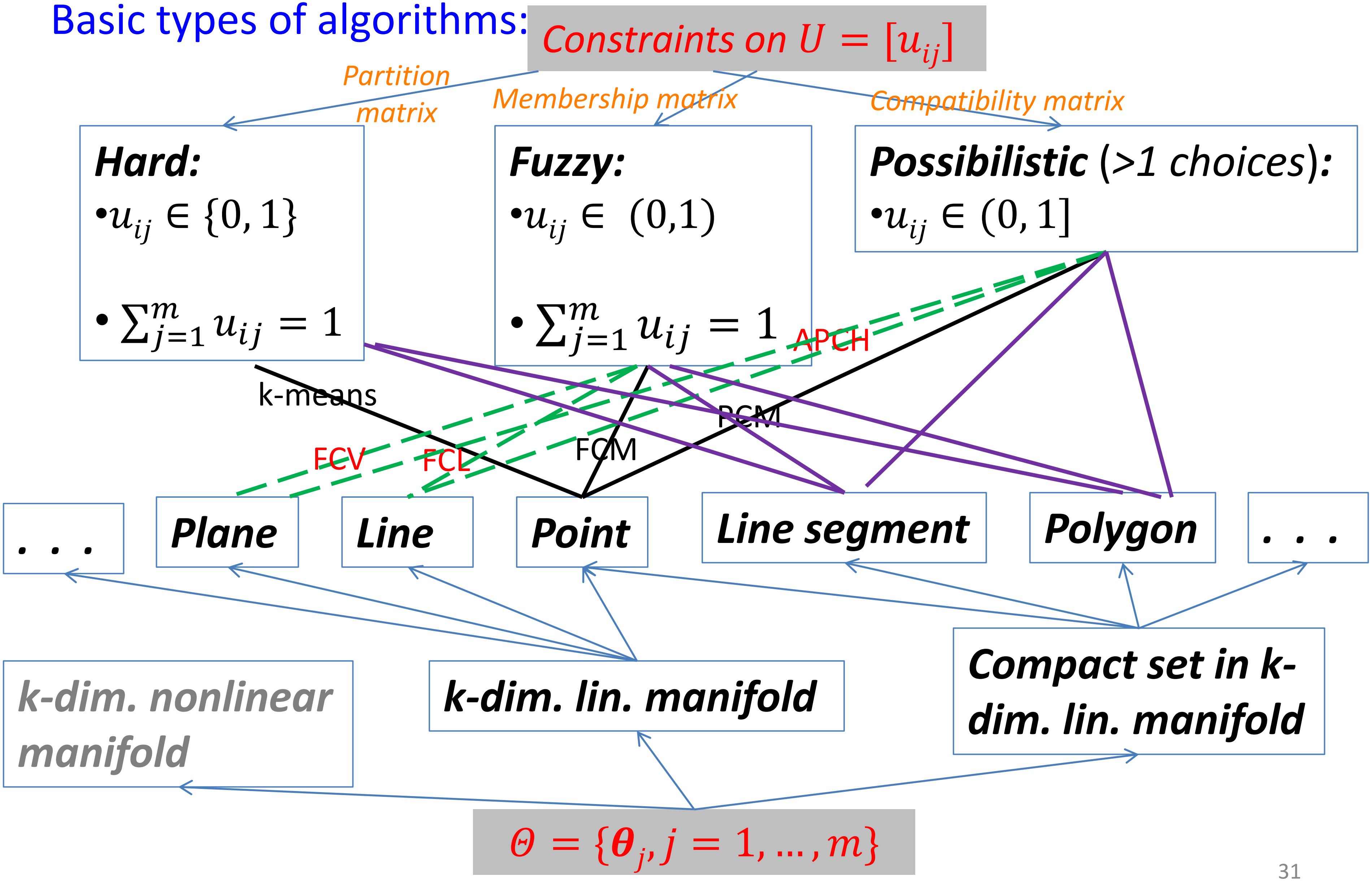
✓ For **fixed** $\boldsymbol{\theta}_j$'s, $J(\Theta, U)$ is a weighted sum of **fixed** distances $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

⇒ **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:



Constraints on $U = [u_{ij}]$

Partition matrix

Membership matrix

Compatibility matrix

**Hard:**

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

**Fuzzy:**

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

**Possibilistic** (>1 choices)**:**

- $u_{ij} \in (0, 1]$

k-means

FCV

FCL

FCM

PCM

APCH

**. . .**

**Plane**

**Line**

**Point**

**Line segment**

**Polygon**

**. . .**

**k-dim. nonlinear manifold**

**k-dim. lin. manifold**

**Compact set in k-dim. lin. manifold**

$\Theta = \{\boldsymbol{\theta}_j, j = 1, \dots, m\}$

# 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

“Array of CFO algorithms”

$C(U)$

$\boldsymbol{\theta}_j$

| | Hard Constr. | Fuzzy Constr. | Possib. Constr. | . . . |
|---|---|---|---|---|
| Point | Hard CFO scheme | Fuzzy CFO scheme | Possib. CFO scheme | |
| Line | | | | |
| Hyperplane | | | | |
| Hyperellipsoid | | | | |
| . . . | | | | |

CFO scheme

# CFO clustering algorithms: A unified view

"Array of CFO algorithms"

$C(U)$

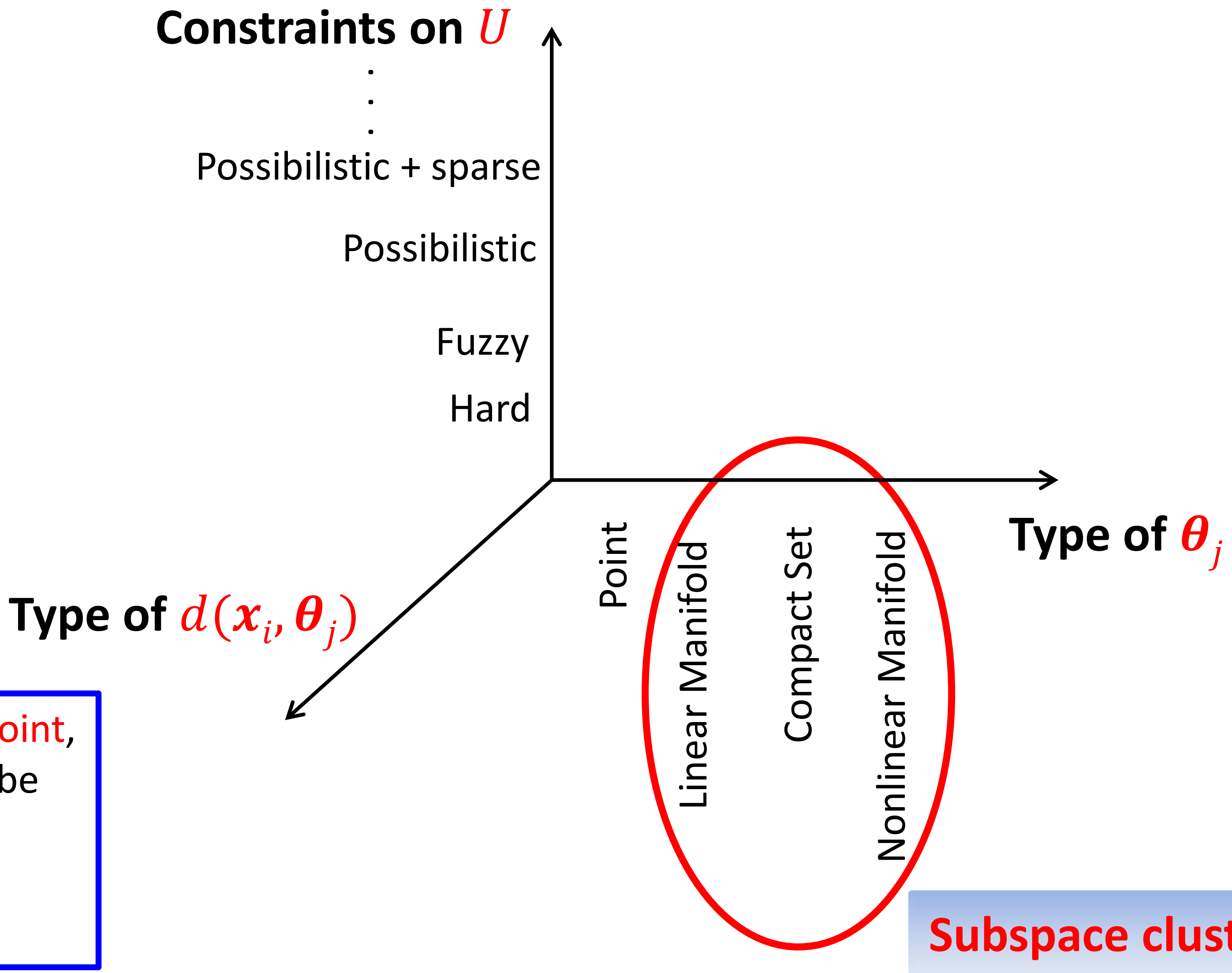
$\boldsymbol{\theta}_j$

| | 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

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

✓ *Initialize* $\Theta = \Theta(0)$

✓ $t = 0$

✓ **Repeat**

- $U(t) = argmin_U\ J(\Theta(t), U)$ , s.t. $C(U(t))$
- $t = t + 1$
- $\Theta(t) = argmin_\Theta\ J(\Theta, U(t-1))$

fixed

✓ **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

CFO scheme

# Cost function optimization (CFO) algorithms

**Hard clustering algorithms:**

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ 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

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

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

*Generalized Hard Algorithmic Scheme (GHAS)*

```
• Choose $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j, j = 1, \dots, m$.
• $t = 0$
• Repeat
    – For $i = 1$ to $N$  % Determination of 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

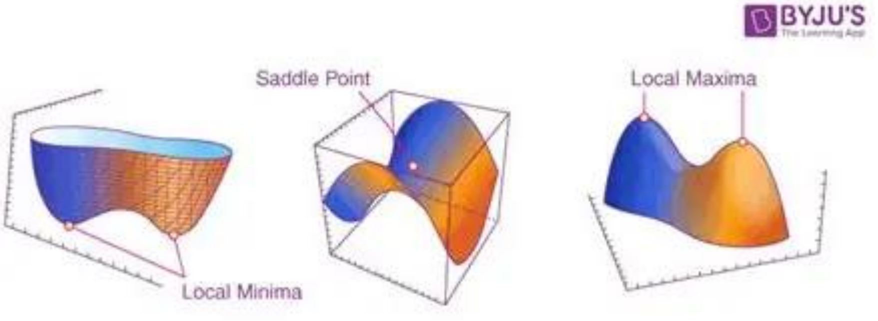
*Generalized Hard Algorithmic Scheme (GHAS)*

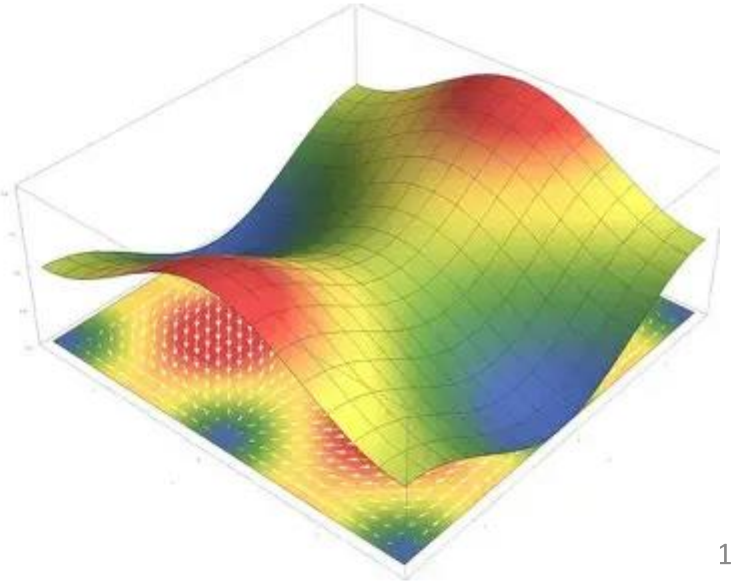
**Remarks:**

- 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 $J(U, \Theta)$.





# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

## 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

*Generalized Hard Algorithmic Scheme (GHAS)*

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**

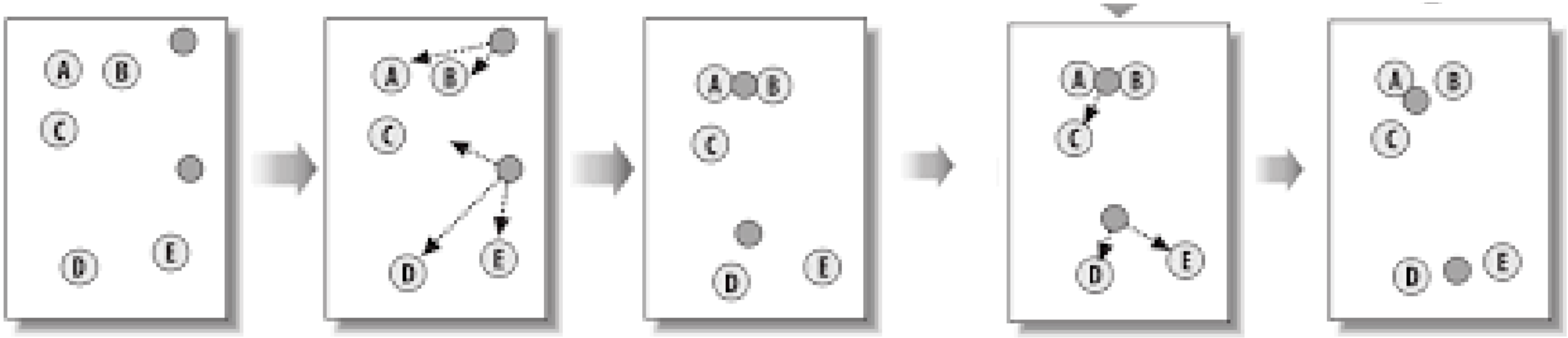
```
– For $i = 1$ to $N$   % Determination of the partition
      o For $j=1$ to $m$
          $u_{ij}(t) = \begin{cases} 1, & if\ ||\boldsymbol{x}_i - \boldsymbol{\theta}_j(t)||^2 = min_{q=1,\ldots,m}||\boldsymbol{x}_i - \boldsymbol{\theta}_q(t)||^2 \\ 0, & otherwise \end{cases}$
      o End {For-$j$}
– 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

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

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 $O(q \cdot m \cdot N)$ ($q$ is the number of iterations).
- It requires the number of clusters $m$ to be known a priori.

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

The Isodata or $k$-Means or $c$-Means algorithm

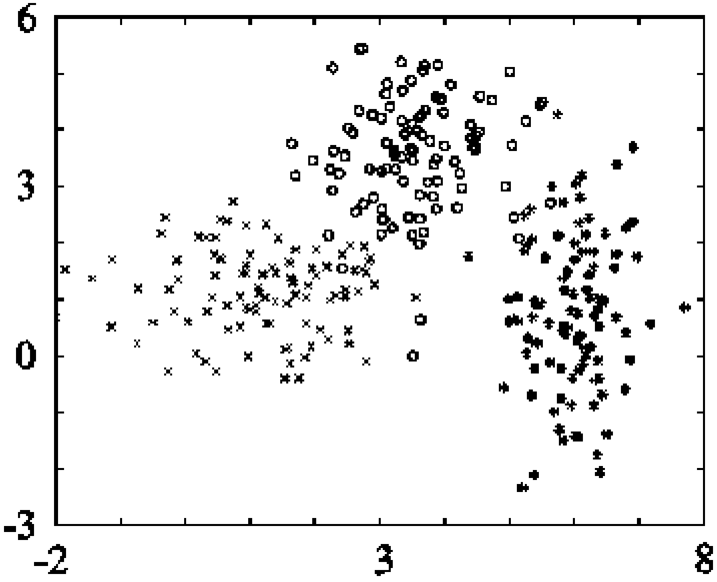
Example 1: (a) Consider three two-dimensional normal distributions with mean values:

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

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.

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

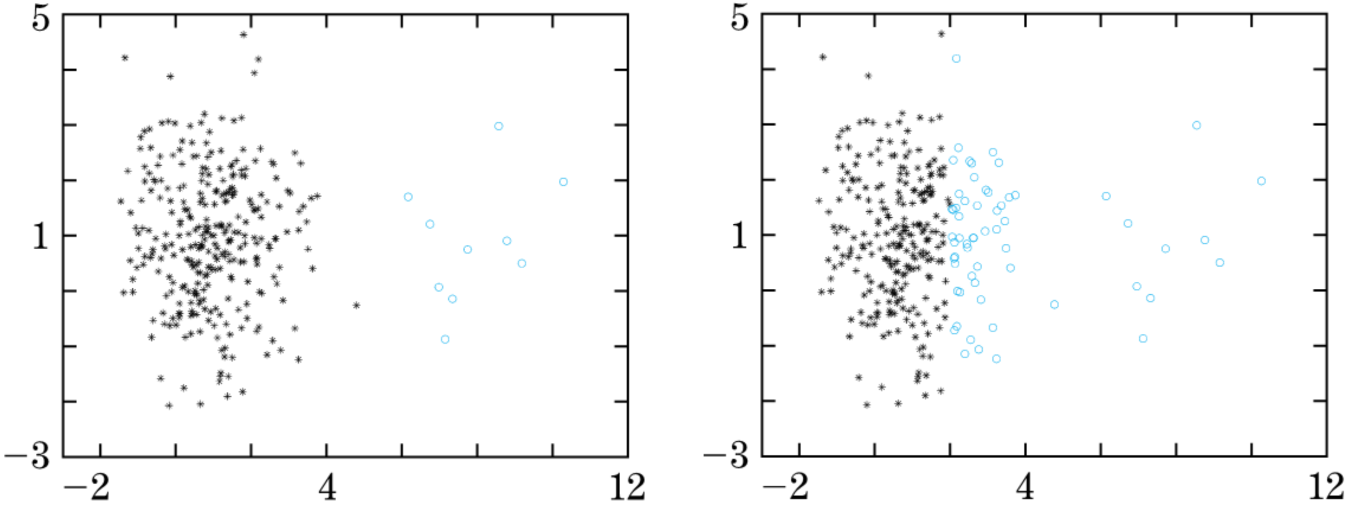
# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

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 $1^{\text{st}}$ distribution and $10$ points from the $2^{\text{nd}}$ distribution. (iii) Set $m = 2$ and initialize randomly $\boldsymbol{\theta}_j$'s ($\boldsymbol{\theta}_j \equiv \boldsymbol{\mu}_j$).

- 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.



# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

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 of the cost function.*

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 (associated with the minimum cost function value).
- *Utilization of tools from stochastic optimization techniques* (simulated annealing, genetic algorithms etc).

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

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 $\Theta$.
  - 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.

# CFO hard clustering algorithms

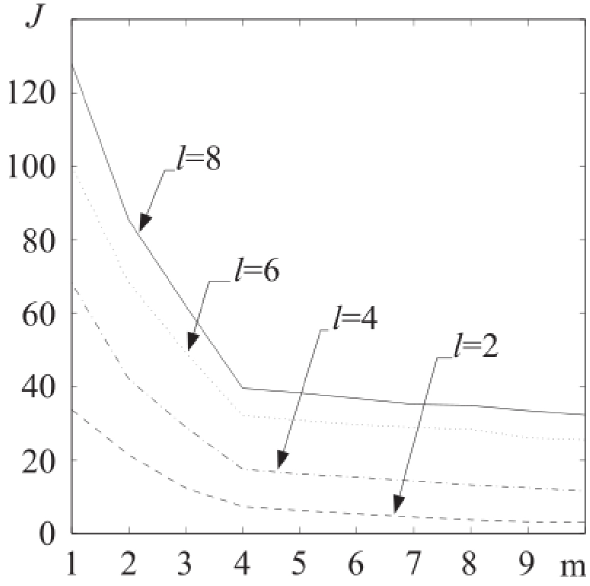
*Generalized Hard Algorithmic Scheme (GHAS)*

The Isodata or $k$-Means or $c$-Means algorithm

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

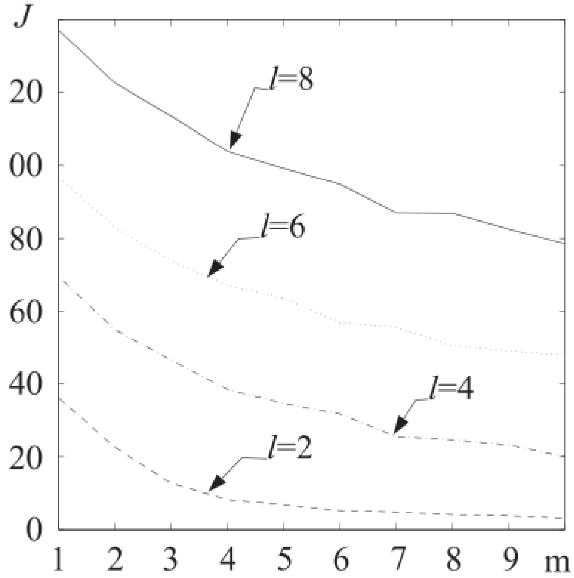
Strategies for facing drawback 2 (cont.):

- 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

*Generalized Hard Algorithmic Scheme (GHAS)*

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.

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*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.

In the $k$-medoids algorithms ONLY **point representatives** can be used to represent clusters

# CFO hard clustering algorithms

## *Generalized Hard Algorithmic Scheme (GHAS)*

### *k-Medoids Algorithms*

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

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

with

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

# CFO hard clustering algorithms

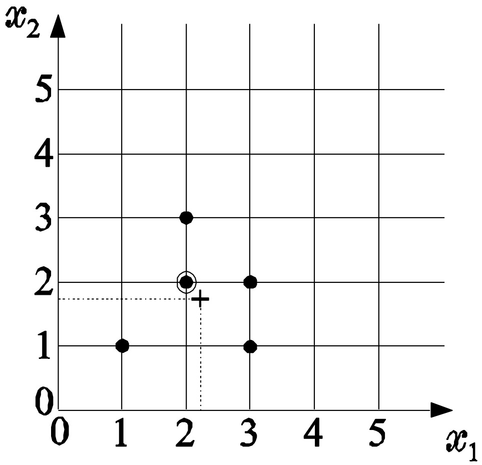
*Generalized Hard Algorithmic Scheme (GHAS)*

*k-Medoids Algorithms*

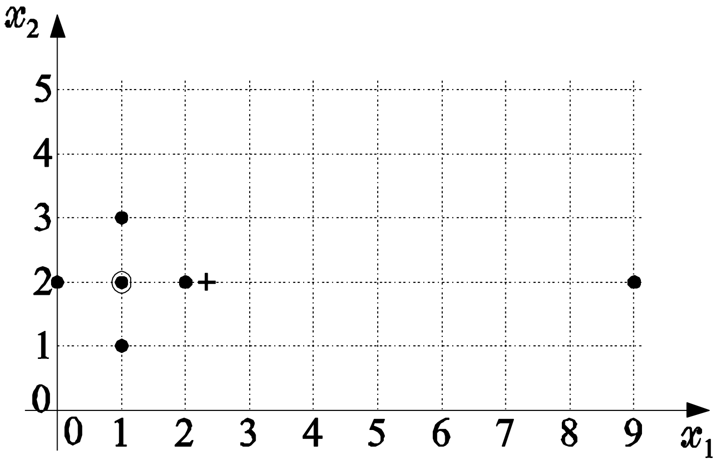
Example 3:

(a) The five-point two-dimensional set stems from the discrete domain $D = \{1,2,3,4, \ldots\} \times \{1,2,3,4, \ldots\}$. 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)

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

Representing clusters with mean values **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 possesses a clear geometrical and statistical meaning** | 3. The medoid has not a clear geometrical meaning |
| **4. Algorithms using means are less computationally demanding** | 4. Algorithms using medoids are more computationally demanding |

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*k-Medoids Algorithms*

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 $\boldsymbol{x}_j$, $j \in I_{X-\Theta}$ **replaces** $\boldsymbol{x}_i$, $i \in I_\Theta$.

- Let $\Delta J_{ij} = J(\Theta_{ij}, U_{ij}) - J(\Theta, U)$.

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

## ***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 $\Delta J_{qr} < 0$ then
    - o Replace $\Theta$ by $\Theta_{qr}$
    - o Go to **(A)**
  - – End

$$\Delta J_{qr} < 0 \Leftrightarrow J(\Theta_{qr}, U_{qr}) - J(\Theta, U) < 0$$
$$\Leftrightarrow J(\Theta_{qr}, U_{qr}) < J(\Theta, U)$$

- *Assignment of points to clusters*
  - – Assign each $\boldsymbol{x} \in X - \Theta$ to the cluster represented by the closest to $\boldsymbol{x}$ medoid.

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*The PAM algorithm*

Computation of $\Delta J_{ij}$.

It is defined as:

$$\Delta J_{ij} = J(\Theta_{ij}, U_{ij}) - J(\Theta, U) = \sum_{s \in I_{X-\Theta_{ij}}} \sum_{t \in I_{\Theta_{ij}}} u_{st} d(\boldsymbol{x}_s, \boldsymbol{x}_t) - \sum_{s \in I_{X-\Theta}} \sum_{t \in I_{\Theta}} u_{st} d(\boldsymbol{x}_s, \boldsymbol{x}_t)$$

$$\equiv \sum_{h \in I_{X-\Theta}} C_{hij}$$

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

For the computation of $C_{hij}$ associated with a specific $\boldsymbol{x}_h \in X - \Theta$ it is required

- The **distance** of $\boldsymbol{x}_h$ from its **closest medoid** in $\Theta$
- The **distance** of $\boldsymbol{x}_h$ from its **next to closest medoid** in $\Theta$.
- The **distance** of $\boldsymbol{x}_h$ from the **newly inserted medoid** in $\Theta_{ij}$, $\boldsymbol{x}_j$.

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*The PAM algorithm (cont.)*

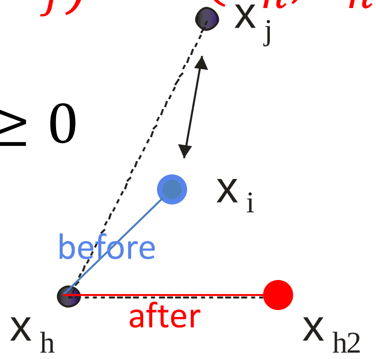
Computation of $C_{hij}$:

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

$$C_{hij} = d(\boldsymbol{x}_h, \boldsymbol{x}_{h2}) - d(\boldsymbol{x}_h, \boldsymbol{x}_i) \geq 0$$

**Contribution** of $\boldsymbol{x}_h$ to $J(\Theta_{ij}, U_{ij})$

**Contribution** of $\boldsymbol{x}_h$ to $J(\Theta, U)$

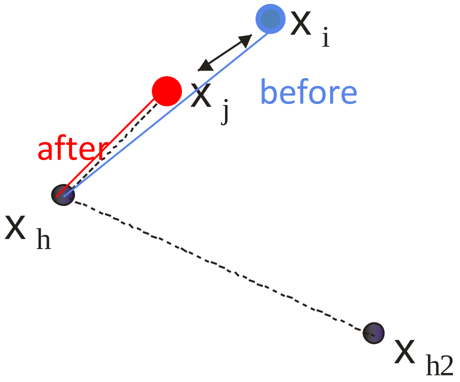


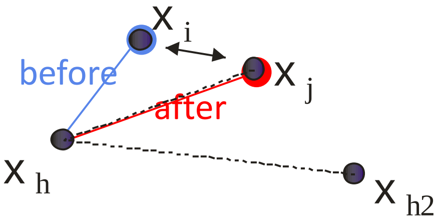
$\boldsymbol{x}_h$ **belongs** to the cluster represented by $\boldsymbol{x}_i$ ($\boldsymbol{x}_{h2} \in \Theta$ denotes the second closest to $\boldsymbol{x}_h$ representative) **and** $d(\boldsymbol{x}_h, \boldsymbol{x}_j) \leq d(\boldsymbol{x}_h, \boldsymbol{x}_{h2})$. Then

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

**Contribution** of $\boldsymbol{x}_h$ to $J(\Theta_{ij}, U_{ij})$

**Contribution** of $\boldsymbol{x}_h$ to $J(\Theta, U)$





# CFO hard clustering algorithms

## *Generalized Hard Algorithmic Scheme (GHAS)*

### *The PAM algorithm (cont.)*

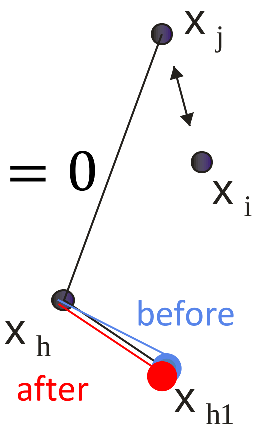
Computation of $C_{hij}$ (cont.):

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

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

**Contribution of** $\boldsymbol{x}_h$ to $J(\Theta_{ij}, U_{ij})$

**Contribution of** $\boldsymbol{x}_h$ to $J(\Theta, U)$



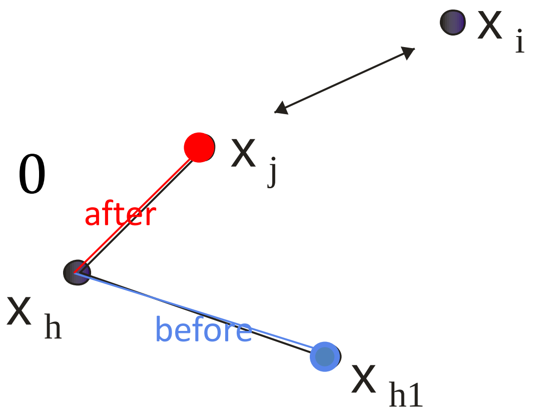
**Most frequent scenario**

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

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

**Contribution of** $\boldsymbol{x}_h$ to $J(\Theta_{ij}, U_{ij})$

**Contribution of** $\boldsymbol{x}_h$ to $J(\Theta, U)$



# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*The PAM algorithm (cont.)*

**Remarks:**

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

# CFO hard clustering algorithms

## Generalized Hard Algorithmic Scheme (GHAS)

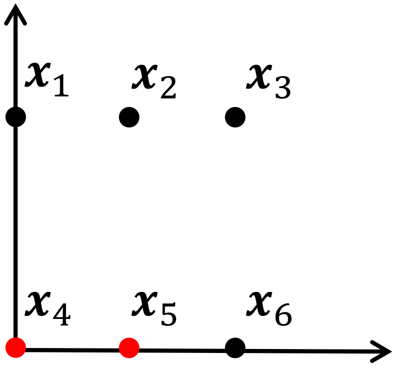
### The PAM algorithm (Example)

**Data set:** $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$, with

$\boldsymbol{x}_1 = [0,3]^T, \boldsymbol{x}_2 = [1,3]^T, \boldsymbol{x}_3 = [2,3]^T, \boldsymbol{x}_4 = [0,0]^T, \boldsymbol{x}_5 = [1,0]^T, \boldsymbol{x}_6 = [2,0]^T.$

**Set of medoids:** $\Theta = \{\boldsymbol{x}_4, \boldsymbol{x}_5\}$

Computation of $J(\Theta, U)$ (**Squared Euclidean distance** is considered):



$$\boldsymbol{x}_1 \longrightarrow d(\boldsymbol{x}_1, \boldsymbol{x}_4) = 9 < 10 = d(\boldsymbol{x}_1, \boldsymbol{x}_5) \longrightarrow u_{14} = 1, u_{15} = 0$$

$$\boldsymbol{x}_2 \longrightarrow d(\boldsymbol{x}_2, \boldsymbol{x}_4) = 10 > 9 = d(\boldsymbol{x}_2, \boldsymbol{x}_5) \longrightarrow u_{24} = 0, u_{25} = 1$$

$$\boldsymbol{x}_3 \longrightarrow d(\boldsymbol{x}_3, \boldsymbol{x}_4) = 13 > 10 = d(\boldsymbol{x}_3, \boldsymbol{x}_5) \longrightarrow u_{34} = 0, u_{35} = 1$$

$$\boldsymbol{x}_4 \longrightarrow d(\boldsymbol{x}_4, \boldsymbol{x}_4) = 0 < 1 = d(\boldsymbol{x}_4, \boldsymbol{x}_5) \longrightarrow u_{44} = 1, u_{45} = 0$$

$$\boldsymbol{x}_5 \longrightarrow d(\boldsymbol{x}_5, \boldsymbol{x}_4) = 1 > 0 = d(\boldsymbol{x}_5, \boldsymbol{x}_5) \longrightarrow u_{54} = 0, u_{55} = 1$$

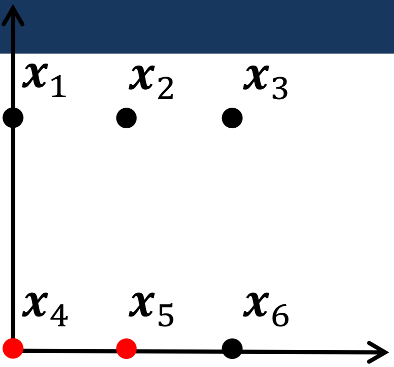
$$\boldsymbol{x}_6 \longrightarrow d(\boldsymbol{x}_6, \boldsymbol{x}_4) = 2 > 1 = d(\boldsymbol{x}_6, \boldsymbol{x}_5) \longrightarrow u_{64} = 0, u_{65} = 1$$

$$J(\Theta, U) = \begin{array}{ll} u_{14}d(\boldsymbol{x}_1, \boldsymbol{x}_4) + & u_{15}d(\boldsymbol{x}_1, \boldsymbol{x}_5) + \\ u_{24}d(\boldsymbol{x}_2, \boldsymbol{x}_4) + & u_{25}d(\boldsymbol{x}_2, \boldsymbol{x}_5) + \\ u_{34}d(\boldsymbol{x}_3, \boldsymbol{x}_4) + & u_{35}d(\boldsymbol{x}_3, \boldsymbol{x}_5) + \\ u_{44}d(\boldsymbol{x}_4, \boldsymbol{x}_4) + & u_{45}d(\boldsymbol{x}_4, \boldsymbol{x}_5) + \\ u_{54}d(\boldsymbol{x}_5, \boldsymbol{x}_4) + & u_{55}d(\boldsymbol{x}_5, \boldsymbol{x}_5) + \\ u_{64}d(\boldsymbol{x}_6, \boldsymbol{x}_4) + & u_{65}d(\boldsymbol{x}_6, \boldsymbol{x}_5) \end{array} = \begin{array}{ll} 1 \cdot 9 + & 0 \cdot 10 + \\ 0 \cdot 10 + & 1 \cdot 9 + \\ 0 \cdot 13 + & 1 \cdot 10 + \\ 1 \cdot 0 + & 0 \cdot 1 + \\ 0 \cdot 1 + & 1 \cdot 0 + \\ 0 \cdot 2 + & 1 \cdot 1 \end{array} = \mathbf{29}$$

# CFO hard clustering algorithms

## Generalized Hard Algorithmic Scheme (GHAS)

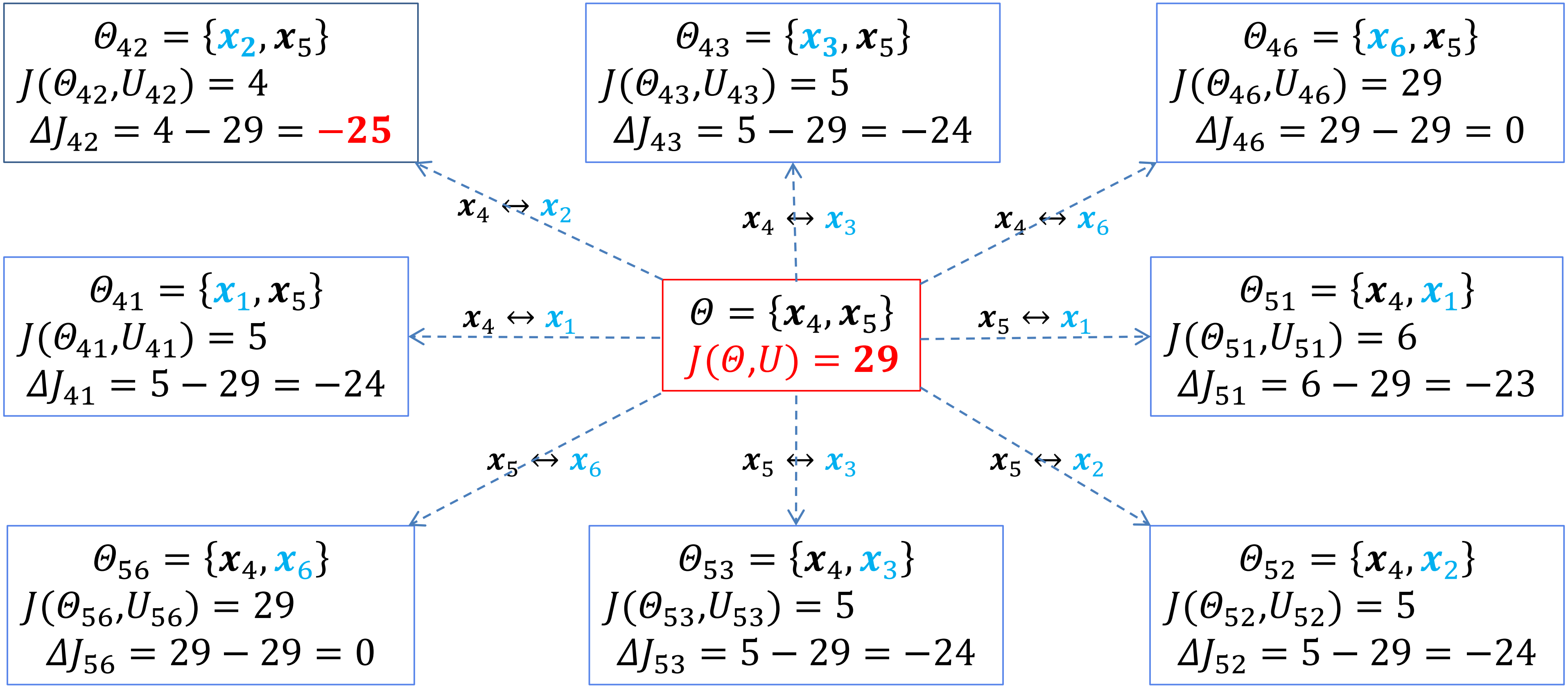
### The PAM algorithm (Example)



**Data set:** $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$, with

$\boldsymbol{x}_1 = [0,3]^T, \boldsymbol{x}_2 = [1,3]^T, \boldsymbol{x}_3 = [2,3]^T, \boldsymbol{x}_4 = [0,0]^T, \boldsymbol{x}_5 = [1,0]^T, \boldsymbol{x}_6 = [2,0]^T$.

**Set of medoids:** $\Theta = \{\boldsymbol{x}_4, \boldsymbol{x}_5\}$



$\Theta = \{\boldsymbol{x}_4, \boldsymbol{x}_5\}$
$J(\Theta,U) = \mathbf{29}$

$\boldsymbol{x}_4 \leftrightarrow \boldsymbol{x}_2$
$\Theta_{42} = \{\boldsymbol{x}_2, \boldsymbol{x}_5\}$
$J(\Theta_{42},U_{42}) = 4$
$\Delta J_{42} = 4 - 29 = \mathbf{-25}$

$\boldsymbol{x}_4 \leftrightarrow \boldsymbol{x}_3$
$\Theta_{43} = \{\boldsymbol{x}_3, \boldsymbol{x}_5\}$
$J(\Theta_{43},U_{43}) = 5$
$\Delta J_{43} = 5 - 29 = -24$

$\boldsymbol{x}_4 \leftrightarrow \boldsymbol{x}_6$
$\Theta_{46} = \{\boldsymbol{x}_6, \boldsymbol{x}_5\}$
$J(\Theta_{46},U_{46}) = 29$
$\Delta J_{46} = 29 - 29 = 0$

$\boldsymbol{x}_4 \leftrightarrow \boldsymbol{x}_1$
$\Theta_{41} = \{\boldsymbol{x}_1, \boldsymbol{x}_5\}$
$J(\Theta_{41},U_{41}) = 5$
$\Delta J_{41} = 5 - 29 = -24$

$\boldsymbol{x}_5 \leftrightarrow \boldsymbol{x}_1$
$\Theta_{51} = \{\boldsymbol{x}_4, \boldsymbol{x}_1\}$
$J(\Theta_{51},U_{51}) = 6$
$\Delta J_{51} = 6 - 29 = -23$

$\boldsymbol{x}_5 \leftrightarrow \boldsymbol{x}_6$
$\Theta_{56} = \{\boldsymbol{x}_4, \boldsymbol{x}_6\}$
$J(\Theta_{56},U_{56}) = 29$
$\Delta J_{56} = 29 - 29 = 0$

$\boldsymbol{x}_5 \leftrightarrow \boldsymbol{x}_3$
$\Theta_{53} = \{\boldsymbol{x}_4, \boldsymbol{x}_3\}$
$J(\Theta_{53},U_{53}) = 5$
$\Delta J_{53} = 5 - 29 = -24$

$\boldsymbol{x}_5 \leftrightarrow \boldsymbol{x}_2$
$\Theta_{52} = \{\boldsymbol{x}_4, \boldsymbol{x}_2\}$
$J(\Theta_{52},U_{52}) = 5$
$\Delta J_{52} = 5 - 29 = -24$

It is $\Delta J_{42} = min_{i \in I_{\Theta},\, j \in I_{X-\Theta}} \Delta J_{ij} = -25 < 0$

Thus, according to **PAM**, $\Theta$ will be **replaced** by $\Theta_{42}$.

# CFO hard clustering algorithms

## Generalized Hard Algorithmic Scheme (GHAS)

### The PAM algorithm (Example)

**Data set:** $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$, with

$\boldsymbol{x}_1 = [0,3]^T, \boldsymbol{x}_2 = [1,3]^T, \boldsymbol{x}_3 = [2,3]^T, \boldsymbol{x}_4 = [0,0]^T, \boldsymbol{x}_5 = [1,0]^T, \boldsymbol{x}_6 = [2,0]^T.$

**Set of medoids:** $\Theta_{42} = \{\boldsymbol{x}_2, \boldsymbol{x}_5\}$

Computation of $J(\Theta_{42}, U_{42})$ (**Squared Euclidean distance** is considered):

$$\boldsymbol{x}_1 \longrightarrow d(\boldsymbol{x}_1,\boldsymbol{x}_2) = 1 < 10 = d(\boldsymbol{x}_1,\boldsymbol{x}_5) \longrightarrow u_{12} = 1, u_{15} = 0$$

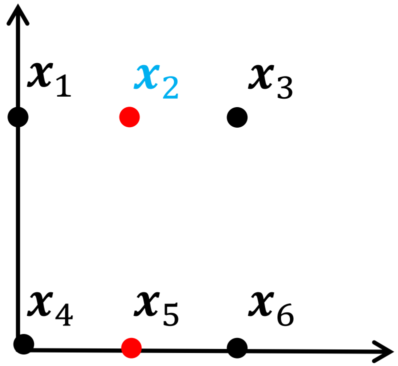
$$\boldsymbol{x}_2 \longrightarrow d(\boldsymbol{x}_2,\boldsymbol{x}_2) = 0 < 9 = d(\boldsymbol{x}_2,\boldsymbol{x}_5) \longrightarrow u_{22} = 1, u_{25} = 0$$

$$\boldsymbol{x}_3 \longrightarrow d(\boldsymbol{x}_3,\boldsymbol{x}_2) = 1 < 10 = d(\boldsymbol{x}_3,\boldsymbol{x}_5) \longrightarrow u_{32} = 1, u_{35} = 0$$

$$\boldsymbol{x}_4 \longrightarrow d(\boldsymbol{x}_4,\boldsymbol{x}_2) = 10 > 1 = d(\boldsymbol{x}_4,\boldsymbol{x}_5) \longrightarrow u_{42} = 0, u_{45} = 1$$

$$\boldsymbol{x}_5 \longrightarrow d(\boldsymbol{x}_5,\boldsymbol{x}_2) = 9 > 0 = d(\boldsymbol{x}_5,\boldsymbol{x}_5) \longrightarrow u_{52} = 0, u_{55} = 1$$

$$\boldsymbol{x}_6 \longrightarrow d(\boldsymbol{x}_6,\boldsymbol{x}_2) = 10 > 1 = d(\boldsymbol{x}_6,\boldsymbol{x}_5) \longrightarrow u_{62} = 0, u_{65} = 1$$



$$J(\Theta_{42}, U_{42}) = \begin{array}{ll} u_{12}d(\boldsymbol{x}_1,\boldsymbol{x}_2) + & u_{15}d(\boldsymbol{x}_1,\boldsymbol{x}_5) + \\ u_{22}d(\boldsymbol{x}_2,\boldsymbol{x}_2) + & u_{25}d(\boldsymbol{x}_2,\boldsymbol{x}_5) + \\ u_{32}d(\boldsymbol{x}_3,\boldsymbol{x}_2) + & u_{35}d(\boldsymbol{x}_3,\boldsymbol{x}_5) + \\ u_{42}d(\boldsymbol{x}_4,\boldsymbol{x}_2) + & u_{45}d(\boldsymbol{x}_4,\boldsymbol{x}_5) + \\ u_{52}d(\boldsymbol{x}_5,\boldsymbol{x}_2) + & u_{55}d(\boldsymbol{x}_5,\boldsymbol{x}_5) + \\ u_{62}d(\boldsymbol{x}_6,\boldsymbol{x}_2) + & u_{65}d(\boldsymbol{x}_6,\boldsymbol{x}_5) \end{array} = \begin{array}{ll} 1 \cdot 1 + & 0 \cdot 10 + \\ 1 \cdot 0 + & 0 \cdot 9 + \\ 1 \cdot 1 + & 0 \cdot 10 + \\ 0 \cdot 10 + & 1 \cdot 1 + \\ 0 \cdot 9 + & 1 \cdot 0 + \\ 0 \cdot 10 + & 1 \cdot 1 \end{array} = \mathbf{4}$$

# CFO hard clustering algorithms

## Generalized Hard Algorithmic Scheme (GHAS)

### The PAM algorithm (Example)

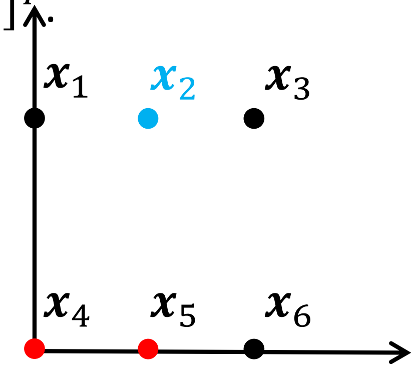
**Data set:** $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$, with

$\boldsymbol{x}_1 = [0,3]^T, \boldsymbol{x}_2 = [1,3]^T, \boldsymbol{x}_3 = [2,3]^T, \boldsymbol{x}_4 = [0,0]^T, \boldsymbol{x}_5 = [1,0]^T, \boldsymbol{x}_6 = [2,0]^T$.

**Sets of medoids:** $\Theta = \{\boldsymbol{x}_4, \boldsymbol{x}_5\}$, $\Theta_{42} = \{\boldsymbol{x}_2, \boldsymbol{x}_5\}$

Computation of $\Delta J_{42}$ as

$\Delta J_{42} = J(\Theta_{42}, U_{42}) - J(\Theta, U) = \sum_{h \in X - \Theta} C_{h42}$ (**Sq. Eucl. dist.** is used):



| | Dist. from Closest repr. in $\Theta = \{\boldsymbol{x}_4, \boldsymbol{x}_5\}$ | Dist. from Next closest repr. in $\Theta = \{\boldsymbol{x}_4, \boldsymbol{x}_5\}$ | Dist. from closest repr. In $\Theta_{42} = \{\boldsymbol{x}_2, \boldsymbol{x}_5\}$ | $C_{h42}$ |
|---|---|---|---|---|
| $\boldsymbol{x}_1$ | 9 ($\boldsymbol{x}_4$) | 10 ($\boldsymbol{x}_5$) | 1 ($\boldsymbol{x}_2$) | $1 - 9 = -8$ |
| $\boldsymbol{x}_2$ | 9 ($\boldsymbol{x}_5$) | 10 ($\boldsymbol{x}_4$) | 0 ($\boldsymbol{x}_2$) | $0 - 9 = -9$ |
| $\boldsymbol{x}_3$ | 10 ($\boldsymbol{x}_5$) | 13 ($\boldsymbol{x}_4$) | 1 ($\boldsymbol{x}_2$) | $1 - 10 = -9$ |
| $\boldsymbol{x}_4$ | 0 ($\boldsymbol{x}_4$) | 1 ($\boldsymbol{x}_5$) | 1($\boldsymbol{x}_5$) | $1 - 0 = 1$ |
| $\boldsymbol{x}_5$ | 0 ($\boldsymbol{x}_5$) | 1 ($\boldsymbol{x}_4$) | 0 ($\boldsymbol{x}_5$) | $0 - 0 = 0$ |
| $\boldsymbol{x}_6$ | 1 ($\boldsymbol{x}_5$) | 2 ($\boldsymbol{x}_4$) | 1 ($\boldsymbol{x}_5$) | $1 - 1 = 0$ |
| $\Delta J_{42}$ | | | | $\boldsymbol{-25}$ |

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*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 $X$, $\Theta'$ is expected to be a good approximation of $\Theta$, which would have been produced if PAM were run on the entire $X$.
- The algorithm:
  - Draw s sample subsets of size $N'$ from $X$, denoted by $X'_1, \dots, X'_s$ (typically $s = 5, N' = 40 + 2m$).
  - Run PAM on each one of them and identify $\Theta'_1, \dots, \Theta'_s$.
  - Choose the set $\Theta'_j$ that minimizes

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

based on the entire data set $X$.

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*The CLARANS algorithm*

- It is more suitable for large data sets.
- It follows the philosophy of PAM with the difference that only a randomly selected fraction $q(< m(N-m))$ of the neighbors of the current medoid set is considered.
- It performs several runs ($s$) starting from different initial choices 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 $\Theta^i$, assign each $\boldsymbol{x} \in X - \Theta$ to the cluster whose representative is closest to $\boldsymbol{x}$
```

# CFO hard clustering algorithms

*Generalized Hard Algorithmic Scheme (GHAS)*

*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.

# Probability and statistics: a brief review

**Random variable (RV):** It models the output of an experiment.

**RV types:**

- Discrete
- continuous

**Discrete random variables:**

- A **discrete RV** $x$ can take any value $x$ from a finite or countably infinite set $X$.
- $X$: sample space or state space.
- **Event:** Any subset of $X$.
- **Elementary** or **simple event**: A single element subset of $X$.
- **Example:** Consider the die roll experiment. X={1,2,3,4,5,6}
- Events: “Odd number”, “number>3”, “2”, “5”

Elementary events

# Probability and statistics: a brief review

**Discrete random variables** (cont.)**:**

•**Notation:** Probability of the event $x = x \in X$: $P(x = x) \equiv P(x)$

•$P(.)$:A function called probability mass function (pmf) satisfying

✓ $P(x) \geq 0, \forall x \in X$

✓ $\sum_{x \in X} P(x) = 1$

# Probability and statistics: a brief review

**Discrete random variables** (cont.):

*The case of more than one random variables: Definitions*

| Discrete RV | $x$ | $y$ |
|---|---|---|
| Sample space | $X=\{x_1,\ldots,x_{nx}\}$ | $Y=\{y_1,\ldots,y_{ny}\}$ |

**Joint probability**: $P(x_i, y_j) \equiv P(x=x_i \text{ AND } y=y_j)$

- It corresponds to the case where $x$ takes the value $x_i$ **AND** $y$ takes the value $y_j$, **simultaneously**.

**Marginal probabilities**: $P(x_i) \equiv P(x=x_i)$, $P(y_j) = P(y=y_j)$

- This terminology is used only when more than one rvs are involved.

**Conditional probability**: $P(x_i | y_j) \equiv P(x=x_i \mid y=y_j) = P(x_i,y_j) / P(y_j)$

- It corresponds to the case where $x$ takes the value $x_i$ **given that** $y$ takes the value $y_j$.

# Probability and statistics: a brief review

**Discrete random variables** (cont.):

*The case of more than one variables: Properties*

| Discrete RV | $x$ | $y$ |
|---|---|---|
| Sample space | $X=\{x_1,\ldots,x_{nx}\}$ | $Y=\{y_1,\ldots,y_{ny}\}$ |

**Sum rule:** $P(x)=\sum_{y\in Y}P(x,y), \quad \forall x\in X$

**Product rule:** $P(x,y)=P(x\,|\,y)P(y)$

Statistical independence: $P(x,y)=P(x)P(y)$

A consequence: $P(x\,|\,y)=P(x) \quad P(y\,|\,x)=P(y)$

**Bayes rule:** $$P(y\,|\,x)=\frac{P(x\,|\,y)P(y)}{P(x)}$$

It plays a key role in ML.

or

$$P(y\,|\,x)=\frac{P(x\,|\,y)P(y)}{\sum_{y\in Y}P(x\,|\,y)P(y)}$$

# Probability and statistics: a brief review

**Continuous random variables:**

- A **continuous RV** $x$ can take any value $x \in R$.

- Sample space or state space: $R$

- **Events:** $\{x \leq x\}$, $\{x_1 < x \leq x_2\}$, $\{x \geq x\}$

- **Cumulative distribution function (cdf):** $F_x(x) = P(x \leq x)$

Corresponds to the probability mass function from the discrete case.

It assigns “mass” to events.

- It is $F_x(\infty) = P(x < \infty) = 1$

- **Probability of events** in terms of **cdf**:
  - $P(x \leq x) = F_x(x)$
  - $P(x_1 < x \leq x_2) = P(x \leq x_2) - P(x \leq x_1) = F_x(x_2) - F_x(x_1)$
  - $P(x \geq x) = = P(x \leq \infty) - P(x \leq x) = 1 - P(x \leq x) = 1 - F_x(x)$

# Probability and statistics: a brief review

**Continuous random variables** (cont.)**:**

- **Assumption:** $F_x(x)$ is *continuous* and *differentiable*.

- **Probability density function (pdf**):

$$p_x(x) = \frac{dF_x(x)}{dx}$$

It assigns "mass" to values.

- **cdf** in terms of **pdf:**

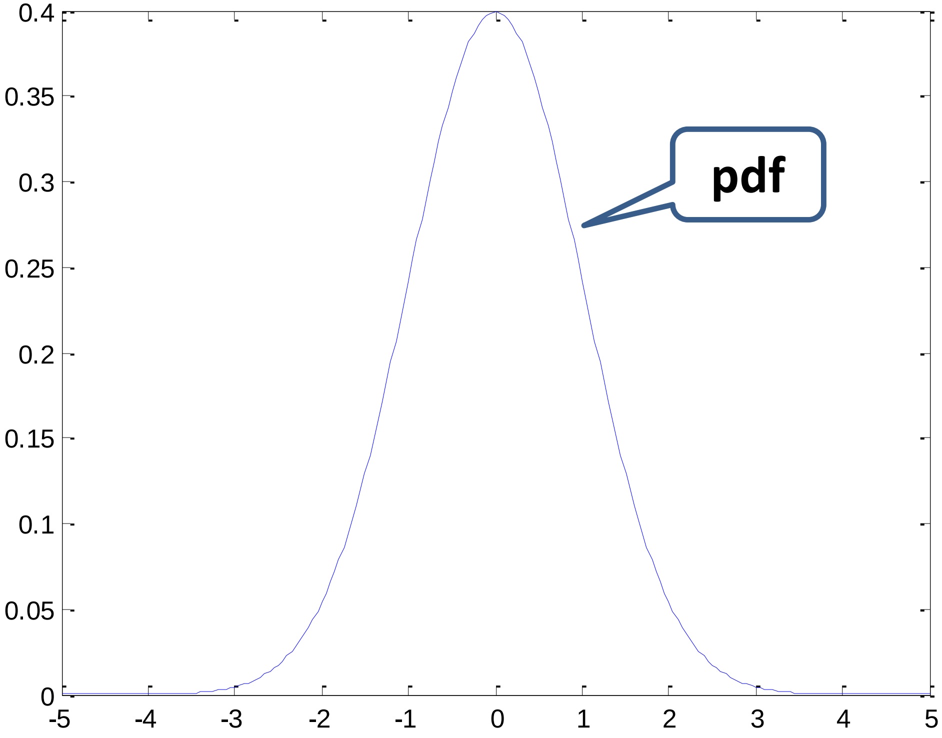
$$F_x(x) = \int_{-\infty}^{x} p_x(z)dz$$

- **Probability of events** in terms of **pdf**:

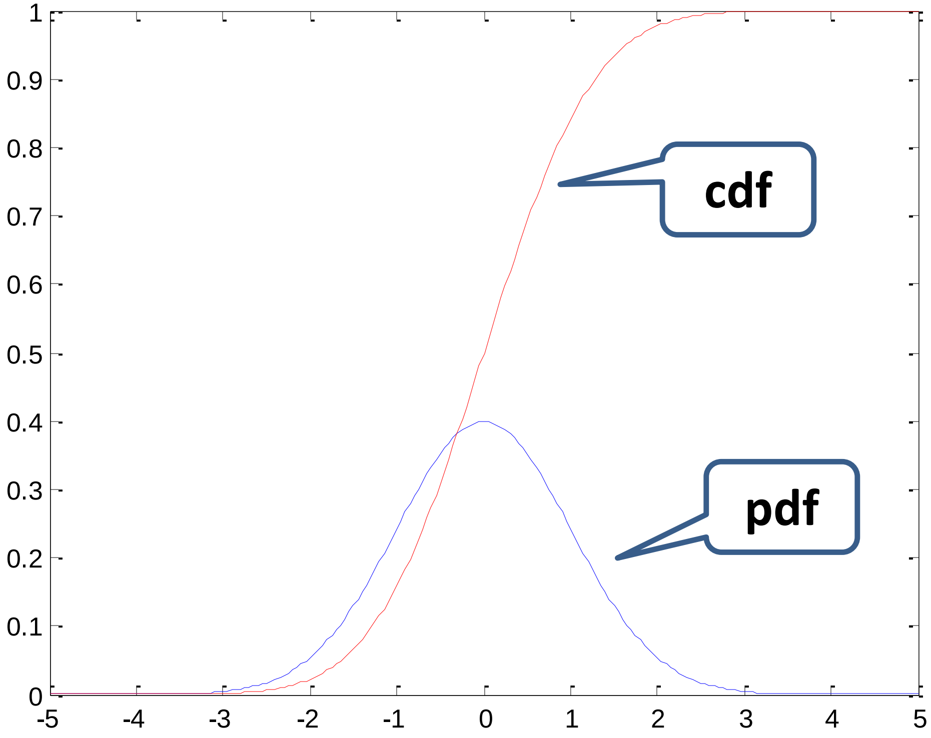
  - $P(x \leq x) = F_x(x) = \int_{-\infty}^{x} p_x(z)dz$
  - $P(x_1 < x \leq x_2) = P(x \leq x_2) - P(x \leq x_1) = F_x(x_2) - F_x(x_1) = \int_{x_1}^{x_2} p_x(x)dx$
  - $P(x \geq x) = = P(x \leq \infty) - P(x \leq x) = 1 - P(x \leq x) = 1 - F_x(x) = \int_{-\infty}^{x} p_x(z)dz$

# Probability and statistics: a brief review

**Continuous random variables** (cont.):



pdf



cdf

pdf

# Probability and statistics: a brief review

**Continuous random variables** (cont.):

- Since $P(-\infty<x<+\infty)=1$ it is: $\int_{-\infty}^{+\infty} p_{\mathrm{x}}(x)dx = 1$

- It is $P(x < \mathrm{x} \le x+\Delta x) = \int_{x}^{x+\Delta x} p_{\mathrm{x}}(z)dz \approx p_{\mathrm{x}}(x)\Delta x$

  As $\Delta x \to 0$, $P(x<x<x+\Delta x)=P(x=x)=0$.

The probability of a continuous rv to take a single value is zero.

*The case of more than one variables:*

| Continuous RV | $x$ | $y$ |
|---|---|---|
| Sample space | $R$ | $R$ |

**NOTE: All rules** stated for the probability mass function in the discrete case are stated for the pdf in the continuous case.

**Product rule**

$$p(x,y) = p(x\,|\,y)\,p(y)$$

We drop the name of rv from the subscript of *p*.

**Sum rule**

$$p(x) = \int_{-\infty}^{+\infty} p(x,y)dy$$

# Probability and statistics: a brief review

**Useful quantities related to (continuous) rvs:**

For **discrete** rv's, the integrals become summations.

- **Mean** (**expected**) **value** of a **rv** *x*: $\mathrm{E[x]} = \int_{-\infty}^{+\infty} x p(x) dx$
- **Variance** of a **rv** *x* : $\sigma_{\mathrm{x}}^2 = \int_{-\infty}^{+\infty} (x - \mathrm{E[x]})^2 p(x) dx = \mathrm{E[(x - E(x))^2]}$
- **Mean** (**expected**) **value** of a **function** of an **rv** *x* : $\mathrm{E}[f(\mathrm{x})] = \int_{-\infty}^{+\infty} f(x) p(x) dx$
- **Mean** of a **function** of two **rv's** *x*, *y*: $\mathrm{E}_{\mathrm{x,y}}[f(\mathrm{x,y})] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y) p(x,y) dxdy$
- **Conditional mean** of an rv *y* given *x = x*: $\mathrm{E[y} \mid x] = \int_{-\infty}^{+\infty} y p(y \mid x) dy$
- It is $\mathrm{E}_{\mathrm{x,y}}[f(\mathrm{x,y})] = E_{\mathrm{x}}[E_{\mathrm{y|x}}[f(\mathrm{x,y})]]$
- **Covariance** between two **rvs** *x* and *y*: $\mathrm{cov(x,y) = E[(x - E[x])(y - E[y])]}$
- **Correlation** between two **rv's** *x* and *y*: $r_{\mathrm{xy}} \equiv \mathrm{E(xy)} = \mathrm{cov}(x,y) + \mathrm{E[x]E[y]}$
- **Correlation coefficient** $r_{xy} = \frac{E[x-E[x])(y-E[y])]}{\sigma_x \sigma_y}$

# Probability and statistics: a brief review

## Random vectors

- A **collection** of **rvs:** $\boldsymbol{x}=[x_1,x_2,\ldots x_l]^T$

- **Probability density function** (**pdf**) of $\boldsymbol{x}$ : The joint pdf of $x_1,x_2,\ldots x_l$.

  $p(\boldsymbol{x})=p(x_1,x_2,\ldots x_l)$

- **Covariance matrix** of $\boldsymbol{x}$ :

$$\mathrm{cov}(\mathbf{x})=\mathrm{E}[(\mathbf{x}-\mathrm{E}[\mathbf{x}])(\mathbf{x}-\mathrm{E}[\mathbf{x}])^{\mathrm{T}}]=\begin{bmatrix}\mathrm{cov}(\mathrm{x}_1,\mathrm{x}_1) & \cdots & \mathrm{cov}(\mathrm{x}_1,\mathrm{x}_l)\\ \vdots & \ddots & \vdots \\ \mathrm{cov}(\mathrm{x}_l,\mathrm{x}_1) & \cdots & \mathrm{cov}(\mathrm{x}_l,\mathrm{x}_l)\end{bmatrix}$$

- **Correlation matrix** of $\boldsymbol{x}$: $R_{\mathbf{x}}=\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}]=\begin{bmatrix}\mathrm{E}(\mathrm{x}_1\mathrm{x}_1) & \cdots & \mathrm{E}(\mathrm{x}_1\mathrm{x}_l)\\ \vdots & \ddots & \vdots \\ \mathrm{E}(\mathrm{x}_l\mathrm{x}_1) & \cdots & \mathrm{E}(\mathrm{x}_l\mathrm{x}_l)\end{bmatrix}$

- It is $R_{\mathbf{x}}\equiv\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}]=\mathrm{cov}(\mathbf{x})+\mathrm{E}[\mathbf{x}]\mathrm{E}[\mathbf{x}^{\mathrm{T}}]$

**Exercise:** Prove this identity

# Probability and statistics: a brief review

## Random vectors (cont.)

**Exercise:** Prove these statements

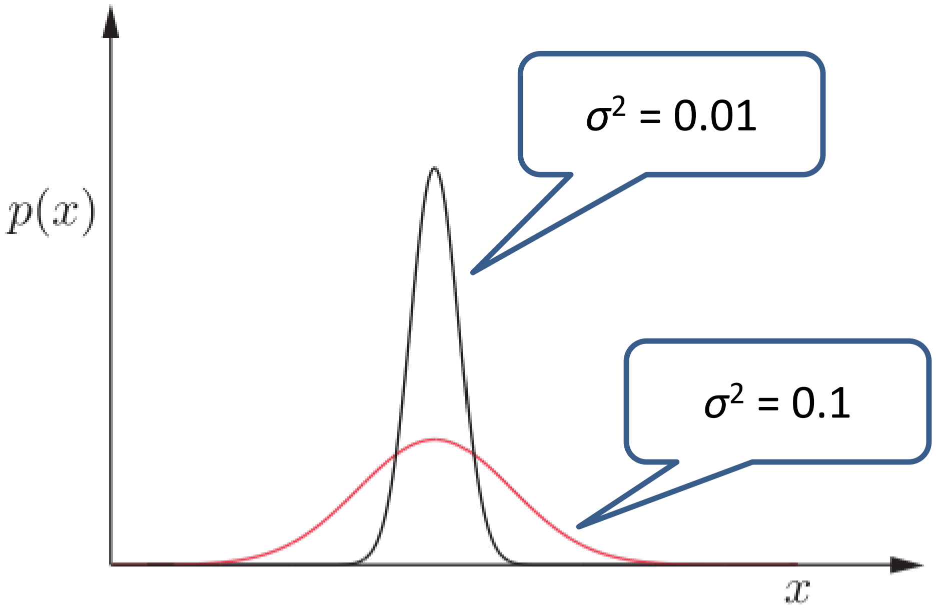
- **Remark:** Both $R_{\boldsymbol{x}}$ and cov($\boldsymbol{x}$) are symmetric and positive definite $l$x$l$ matrices.

A square matrix A is symmetric iff $A^{\mathrm{T}}=A$.

A square matrix A is positive definite iff $\boldsymbol{z}^{\mathrm{T}}A\boldsymbol{z}>0,\ \forall \boldsymbol{z}\in \mathrm{R}^{l}$.

# Probability and statistics: a brief review

- **One dim. normal (Gaussian) distribution $x \sim N(\mu,\sigma^2)$ or $N(x|\mu,\sigma^2)$ :**
  - Sample space: $R$
  - It is
    - $$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$
    - E[x]=$\mu$
    - $\sigma_x^2 = \sigma^2$.



$\sigma^2 = 0.01$

$\sigma^2 = 0.1$

# Probability and statistics: a brief review

- **Multi dim. normal (Gaussian) distribution $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ :**

  - $l$-dim. case
  - It is

    - $p(\boldsymbol{x}) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{1/2}} \exp(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}{2})$

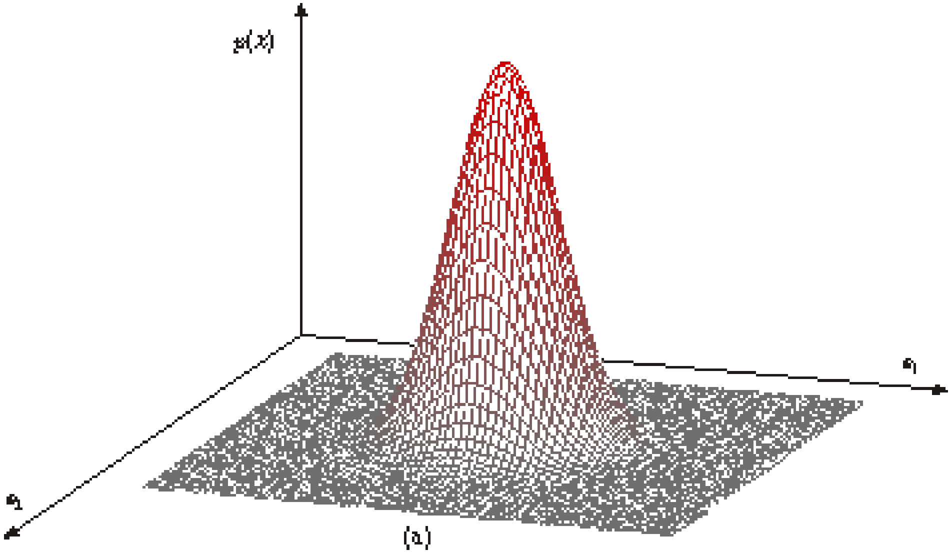
    - $E[\mathbf{x}] = \boldsymbol{\mu}$
    - $cov(\mathbf{x}) = \Sigma.$

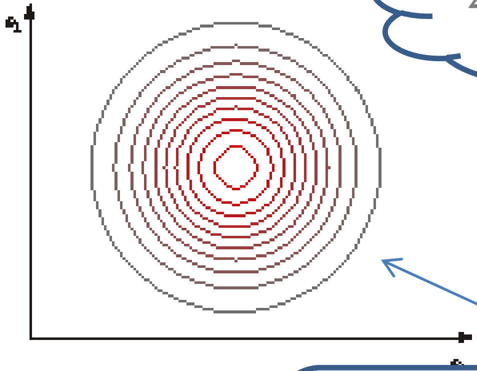
(*) For the 2-d case $\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$

# Probability and statistics: a brief review

- **Multi dim. normal (Gaussian) distribution $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$:**

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$$

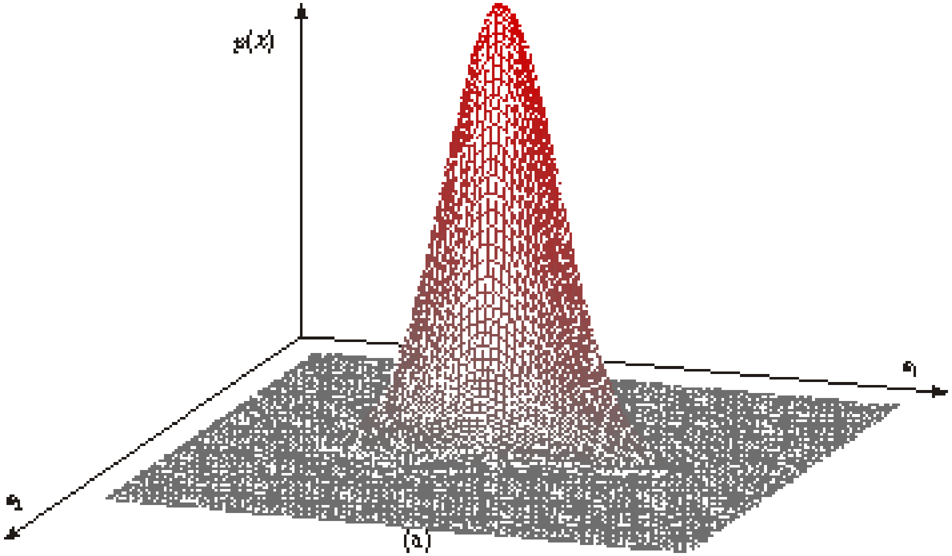


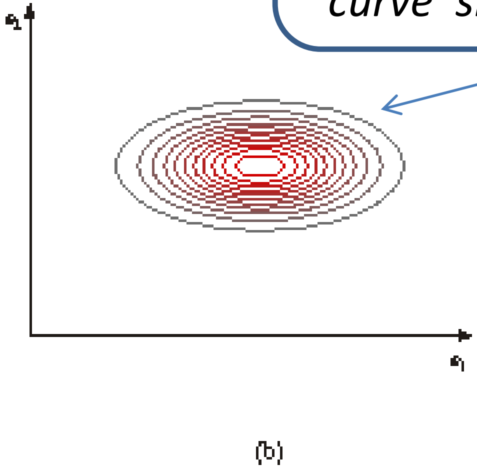


$\Sigma$: **diagonal** with $\sigma_1^2 = \sigma_2^2$

Isovalued curves:

- $(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$ =*const.*
- *All points on each isovalue curve share the value* $p(\boldsymbol{x})$.



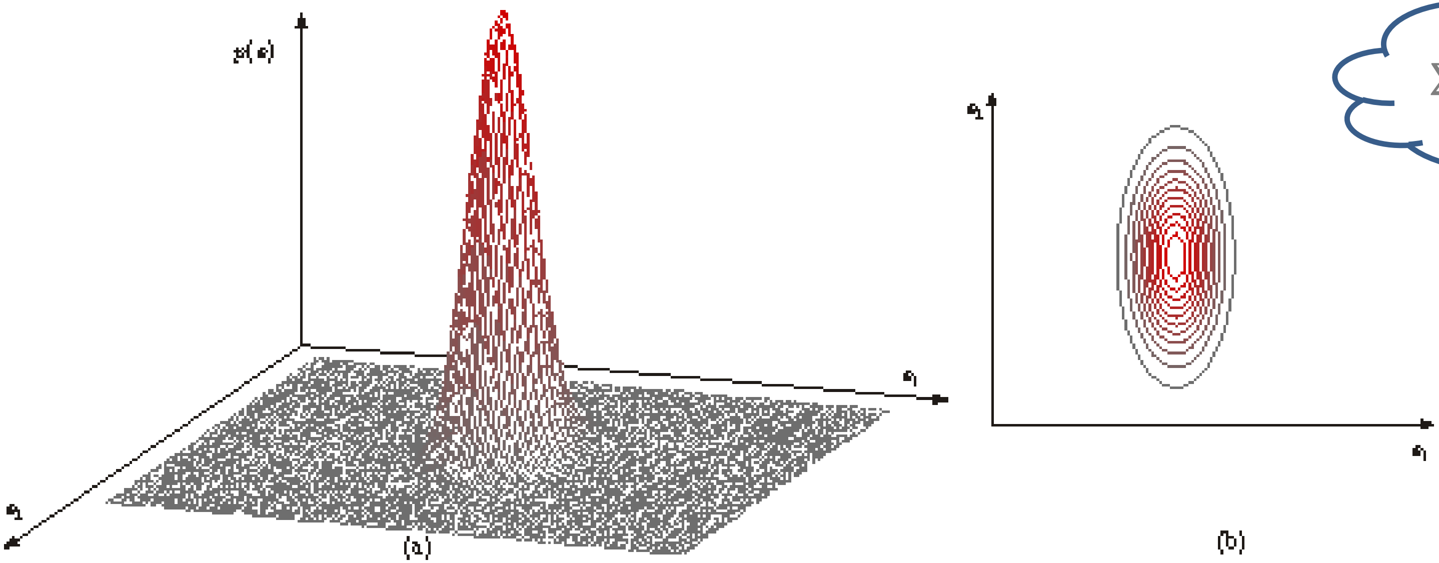


$\Sigma$: **diagonal** with $\sigma_1^2 \gg \sigma_2^2$

# Probability and statistics: a brief review

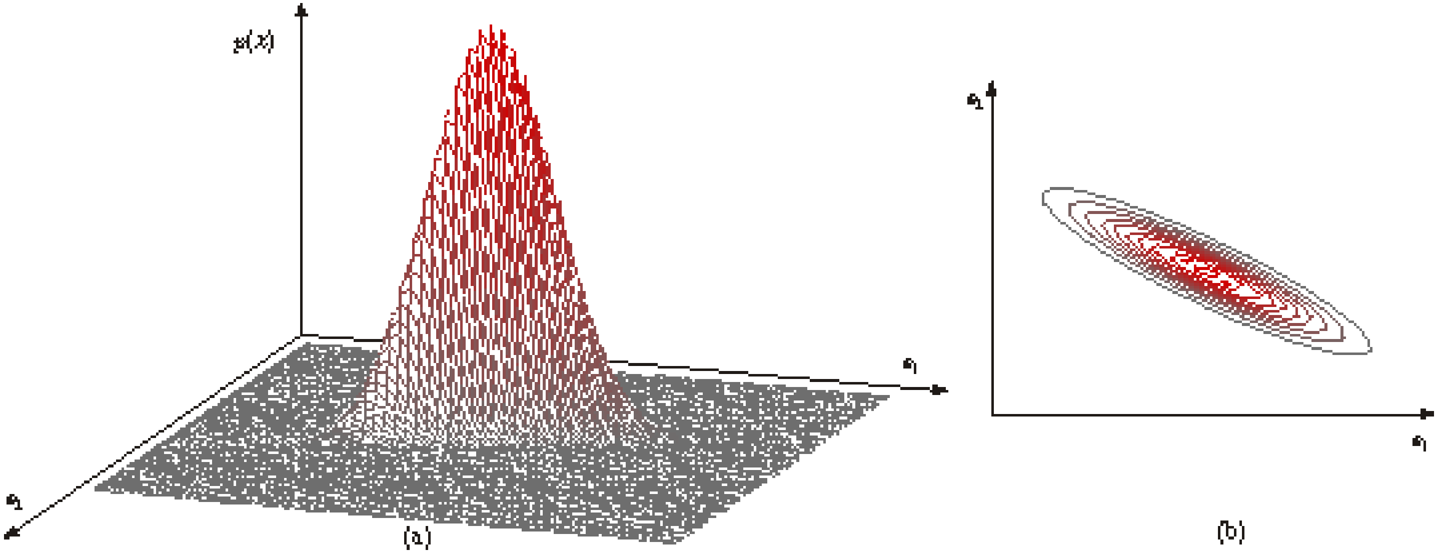
- **Multi dim. normal (Gaussian) distribution $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$:**

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$$



(a) (b)

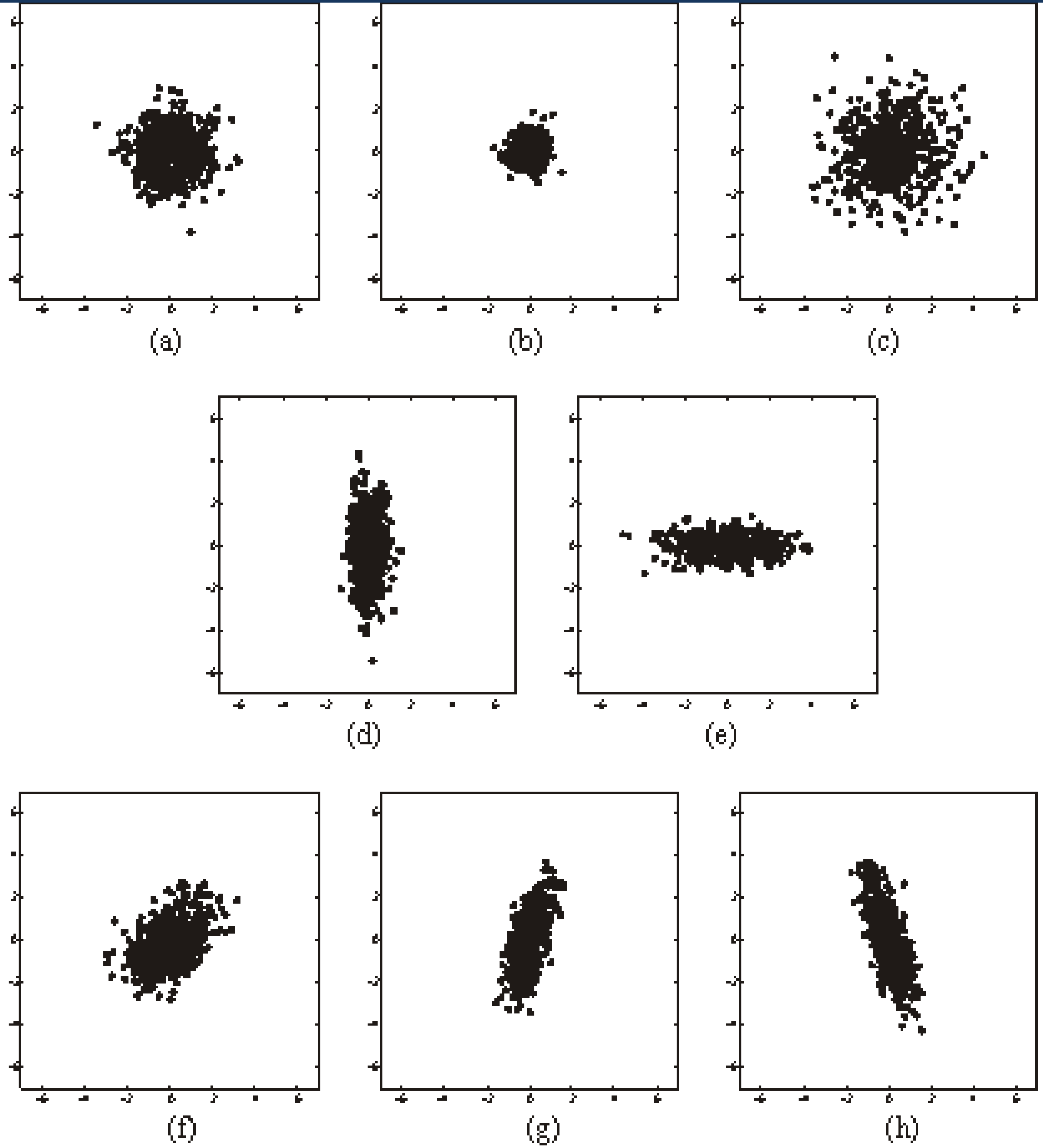
$\Sigma$: diagonal with$\sigma_1^2 \ll \sigma_2^2$



(a) (b)

$\Sigma$: non diagonal

# Probability and statistics: a brief review


(a) (b) (c) (d) (e) (f) (g) (h)

$$\Sigma = \begin{bmatrix} {\sigma_1}^2 & \sigma_{12} \\ \sigma_{12} & {\sigma_2}^2 \end{bmatrix}$$

(a) $\sigma_1^2 = \sigma_2^2 = 1, \sigma_{12} = 0$

(b)$\sigma_1^2 = \sigma_2^2 = 0.2, \sigma_{12} = 0$

(c)$\sigma_1^2 = \sigma_2^2 = 2, \sigma_{12} = 0$

(d) $\sigma_1^2 = 0.2, \ \sigma_2^2 = 2, \sigma_{12} = 0$

(e) $\sigma_1^2 = 2, \ \sigma_2^2 = 0.2, \sigma_{12} = 0$

(f) $\sigma_1^2 = \sigma_2^2 = 1, \sigma_{12} = 0.5$

(g) $\sigma_1^2 = 0.3, \sigma_2^2 = 2, \sigma_{12} = 0.5$

(h) $\sigma_1^2 = 0.3, \sigma_2^2 = 2, \sigma_{12} = -0.5$

# Probability and statistics: a brief review

**Continuous RV distributions** (cont.)

- **Multi dim. normal (Gaussian) distribution x~$N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$:**

From 1-dim. → 2-dim. case.

- 1-dim. case: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\frac{(x-\mu)\sigma^{-2}(x-\mu)}{2}\right)$

- A first extension to the 2-dim. case (independent rv's):
- $p(x_1, x_2) = p_1(x_1) \cdot p_2(x_2) =$
- $\frac{1}{(2\pi)^{1/2} \cdot \sigma_1} \exp\left(-\frac{(x_1-\mu_1){\sigma_1}^{-2}(x_1-\mu_1)}{2}\right) \cdot \frac{1}{(2\pi)^{1/2} \cdot \sigma_2} \exp\left(-\frac{(x_2-\mu_2){\sigma_2}^{-2}(x_2-\mu_2)}{2}\right) =$

$$\frac{1}{(2\pi)|\Sigma|^{1/2}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

- The final extension to the 2-dim. case (dependent rv's):
- $p(x_1, x_2) = \frac{1}{(2\pi)|\Sigma|^{1/2}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$$

# Probability and statistics: a brief review

- **Multi dim. normal (Gaussian) distribution $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $N(\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ :**

**Properties**

1. If the covariance matrix $\Sigma$ is diagonal, then, the rv's $\mathrm{x}_1, \dots, \mathrm{x}_l$ comprising $\mathbf{x}$ are statistically independent. It is

$$p(\boldsymbol{x}) = \prod_{i=1}^{l} p_i(x_i) = \prod_{i=1}^{l} \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left(-\frac{(x_i - \mu_\iota)^2}{2\sigma_i{}^2}\right)$$

2. *Central limit theorem:*

Let:

- $\mathrm{x}_1, \dots, \mathrm{x}_r$ independent rvs following different distributions
- $\mu_i, \sigma_i{}^2$ mean and variance of $\mathrm{x}_i$.
- Define $\mathrm{x} = \mathrm{x}_1 + \cdots + \mathrm{x}_r$, $\mu = \mu_1 + \cdots + \mu_r$ , $\sigma^2 = \sigma_1^2 + \cdots + \sigma_r^2$.
- Define $z = (\mathrm{x} - \mu)/\sigma$.

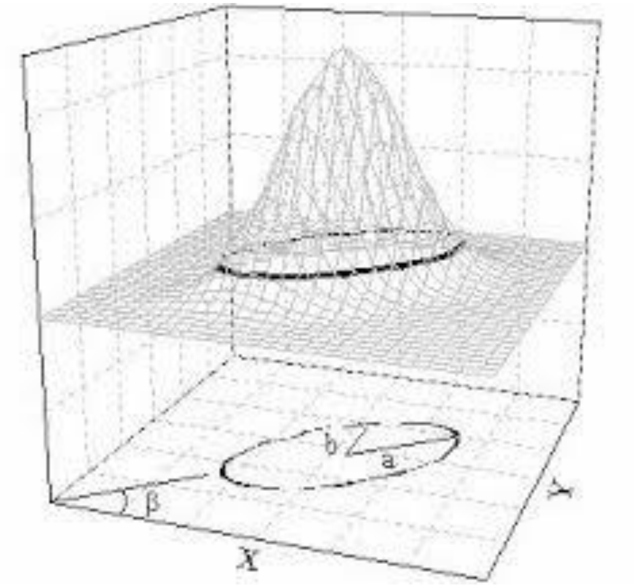
Then

- $p(z) \rightarrow N(z|0,1)$, as $r \rightarrow \infty$

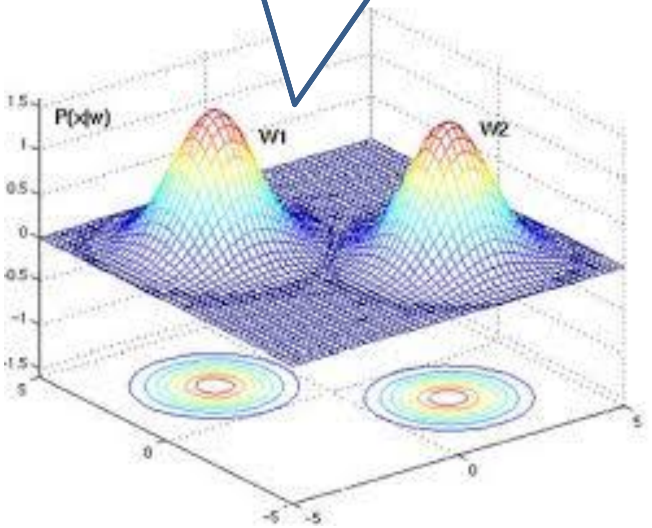
# Probability and statistics: a brief review

## **Continuous RV distributions** (cont.)

- **Other examples of multi-dimensional pdfs**



Two-dim. pdfs



Bimodal distribution

# Probability and statistics: a brief review

## Likelihood function

- Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ a set of independent data vectors
- Let $p_{\boldsymbol{\theta}}(\cdot)$ be a pdf belonging to a known parametric set of pdf functions of parameter vector $\boldsymbol{\theta}$.
- $p(\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}) \equiv p(\boldsymbol{x};\, \boldsymbol{\theta})$.

  ***Examples:***

  ➢ If $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is normal distribution parameterized on the mean vector $\boldsymbol{\mu}$, $\boldsymbol{\theta}$ will simply be $\boldsymbol{\mu}$.

  ➢ If $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is normal distribution parameterized on both the mean vector $\boldsymbol{\mu}$ and the cov. matrix $\Sigma$, $\boldsymbol{\theta}$ will contain the coordinates of both $\boldsymbol{\mu}$ and $\Sigma$.

Likelihood function of $\boldsymbol{\theta}$ wrt $X$: $p(X; \boldsymbol{\theta}) = p(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N; \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i; \boldsymbol{\theta})$

Log-likelihood function of $\boldsymbol{\theta}$ wrt $X$:

$$L(\boldsymbol{\theta}) = \ln p(X; \boldsymbol{\theta}) = \ln p(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N; \boldsymbol{\theta}) = \sum_{i=1}^{N} \ln p(\boldsymbol{x}_i; \boldsymbol{\theta})$$

# Probability and statistics: a brief review

## Likelihood function

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2})$$

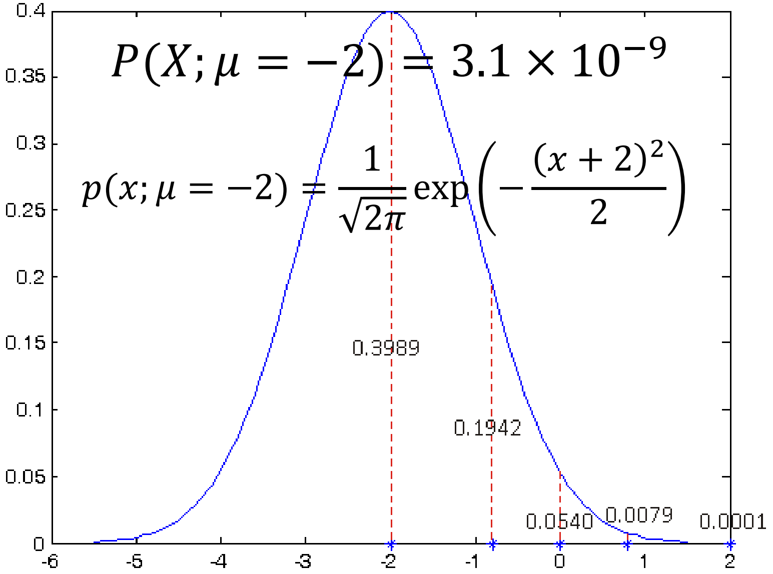
**Example:**

- $X = \{-2, -1, 0, 1, 2\}$
- Consider the parametric set of normal distributions of unit variance, parameterized on $\mu$.
- The likelihood of $\mu$ wrt $X$ is

$$p(X;\mu) = p(-2,-1,0,1,2;\mu) =$$
$$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(-2-\mu)^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(-1-\mu)^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(0-\mu)^2}{2}\right)$$
$$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(1-\mu)^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(2-\mu)^2}{2}\right)$$

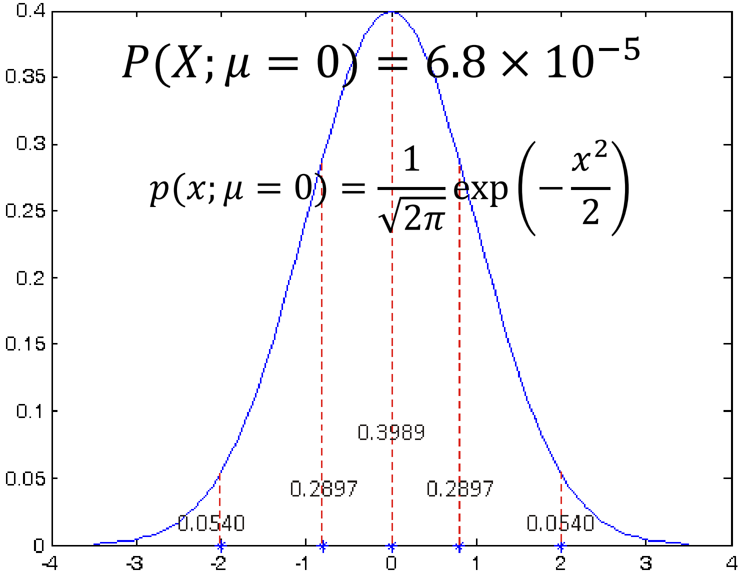
# Probability and statistics: a brief review

## Likelihood function



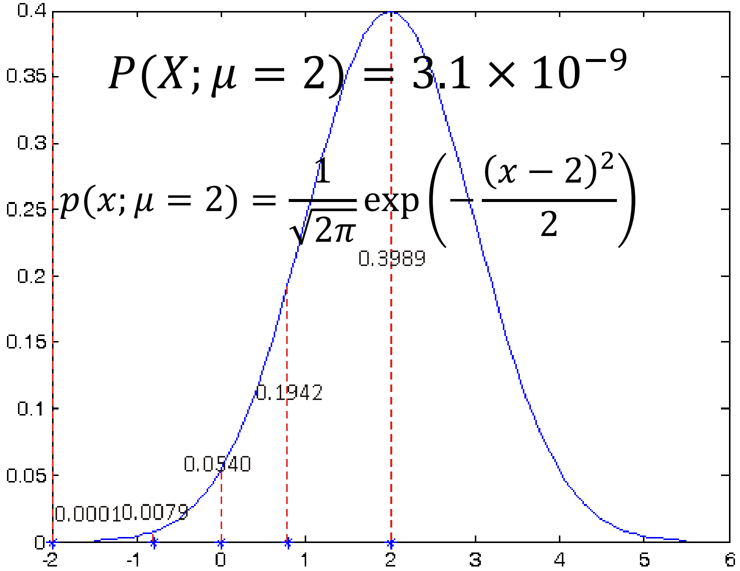
$$P(X;\mu=-2)=3.1\times10^{-9}$$

$$p(x;\mu=-2)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x+2)^2}{2}\right)$$



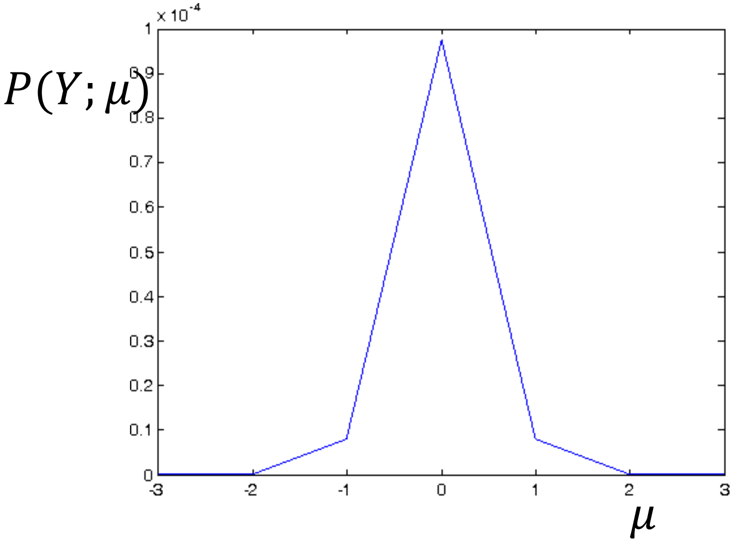
$$P(X;\mu=0)=6.8\times10^{-5}$$

$$p(x;\mu=0)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$$



$$P(X;\mu=2)=3.1\times10^{-9}$$

$$p(x;\mu=2)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-2)^2}{2}\right)$$

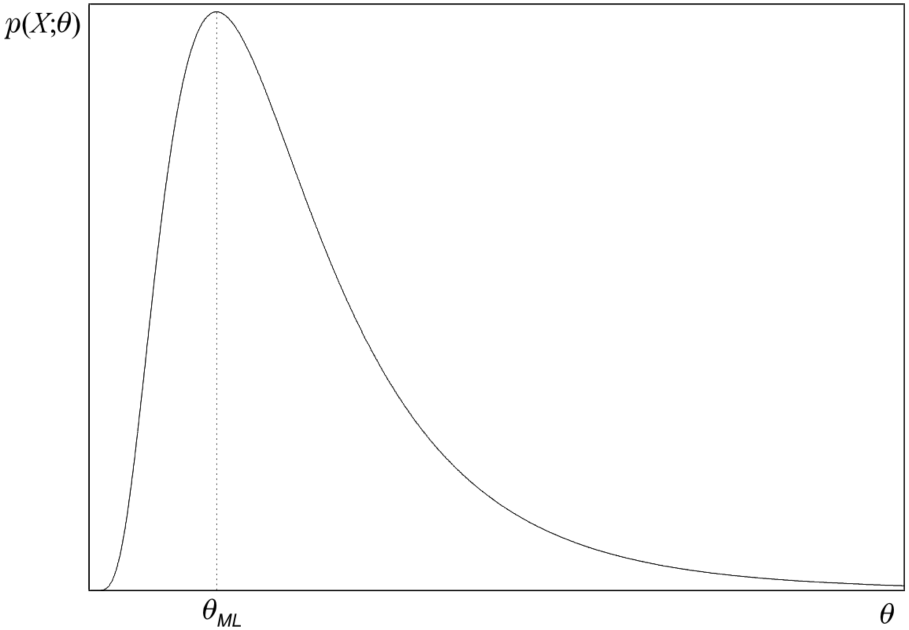


# Probabilistic CFO clustering algorithms

**Maximum likelihood (ML) method:**

Given a set of independent data vectors $Y = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$, estimate the parameter vector $\boldsymbol{\theta}$ as the maximum of the likelihood ($p(Y;\boldsymbol{\theta})$) or the log-likelihood ($L(\boldsymbol{\theta})$) function.

$$\boxed{\hat{\boldsymbol{\theta}}_{ML} = argmax_{\boldsymbol{\theta}}\, p(Y;\boldsymbol{\theta})} \rightarrow \boxed{\hat{\boldsymbol{\theta}}_{ML}: \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{\kappa=1}^{N} \frac{1}{p(\boldsymbol{x}_k;\theta)} \frac{\partial p(\boldsymbol{x}_k;\theta)}{\partial \boldsymbol{\theta}} = \mathbf{0}}$$



Since $\ln(\cdot)$ is an increasing function, $p(Y;\boldsymbol{\theta})$ and $L(\boldsymbol{\theta})$ share the same maxima.

# Probabilistic CFO clustering algorithms

**Maximum likelihood (ML) method:**

**Assuming that**

- the chosen model $p(\boldsymbol{x};\, \boldsymbol{\theta})$ is correct and
- there exists a true parameter $\boldsymbol{\theta}_{\boldsymbol{o}}$,

**the ML estimator**

(a) is asymptotically **unbiased** $lim_{N\to\infty}E\left[\widehat{\boldsymbol{\theta}}_{ML}\right] = \boldsymbol{\theta}_o$

(b) is asymptotically **consistent** $lim_{N\to\infty}Prob\left\{\left\|\widehat{\boldsymbol{\theta}}_{ML} - \boldsymbol{\theta}_o\right\|\right\} = 0$

(c) is asymptotically **efficient** (it achieves the Cramer-Rao lower bound)

The **pdf** of the ML estimator approaches the normal distribution with mean $\boldsymbol{\theta}_{\boldsymbol{o}}$, as $N\to\infty$.

# Maximum likelihood method

**Example 1:**

-Let $Y$ be a set of $N$ (independent from each other) data points, $\boldsymbol{x}_i$, $i = 1, \dots, N$, generated by a normal distribution $p(\boldsymbol{x};\ \boldsymbol{\theta})$ of known covariance matrix and unknown mean.

-Determine the ML estimate of the mean $\boldsymbol{\mu}$ of $p(\boldsymbol{x}; \boldsymbol{\theta})$, based on $Y$.

**Solution:**

-The unknown parameter vector in this case is the mean vector $\boldsymbol{\mu}$, i.e. $\boldsymbol{\theta} \equiv \boldsymbol{\mu}$.

-It is

$$p(\boldsymbol{x}; \boldsymbol{\theta}) \equiv p(\boldsymbol{x}; \boldsymbol{\mu}) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{1/2}} \cdot exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \Longrightarrow$$

$$\ln p(\boldsymbol{x}; \boldsymbol{\mu}) = \ln\frac{1}{(2\pi)^{l/2}|\Sigma|^{1/2}} - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = C - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$$

Then

$$L(\boldsymbol{\mu}) = \sum_{i=1}^{N} \ln p(\boldsymbol{x}_i; \boldsymbol{\mu}) = NC - \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{x}_i-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})$$

# Maximum likelihood method

**Example 1** (cont.):

Setting the gradient of $L(\boldsymbol{\mu})$ wrt $\boldsymbol{\mu}$ equal to $\mathbf{0}$ we have

$$\frac{\partial L(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = \frac{\partial}{\partial \boldsymbol{\mu}}\left(NC - \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\right) = \mathbf{0} \Leftrightarrow$$

$$\sum_{i=1}^{N}\Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) = \mathbf{0} \Leftrightarrow \sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}) = \mathbf{0} \Leftrightarrow \sum_{i=1}^{N}\boldsymbol{x}_i - N\boldsymbol{\mu} = \mathbf{0}$$

$$\boldsymbol{\mu}_{ML} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$$

**Remark:** The ML estimate for the covariance matrix is

$$\Sigma_{ML} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$