

# Clustering algorithms

## Konstantinos Koutroumbas

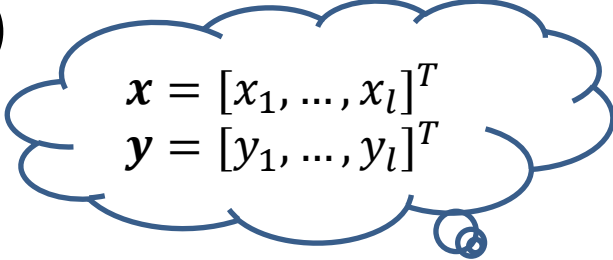
### Unit 3

- Proximity measures for mixed-valued vectors
- Fuzzy measures
- Missing data issue
- Short introduction to clustering algorithms
- Sequential clustering algorithms

# Proximity measures between vectors

## (D) Mixed-valued vectors –similarity measures (SMs)

Here **some coordinates** of the feature vectors are **real-valued**, while **others** are **discrete-valued**.


$$\mathbf{x} = [x_1, \dots, x_l]^T$$
$$\mathbf{y} = [y_1, \dots, y_l]^T$$

How to **measure** the **proximity** between  $\mathbf{x}$  and  $\mathbf{y}$ ?

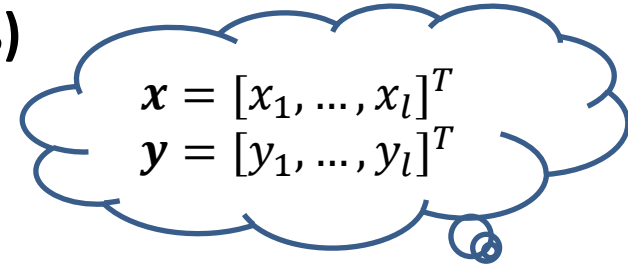
- **Adopt** a **proximity measure suitable** for **real-valued vectors** (**only** for **ordinal discrete-valued** features).
- **Convert** the **real-valued features to discrete-valued ones** (e.g., via quantization) and **employ** a **discrete proximity measure** (again, **only** for **ordinal discrete-valued** features).
- For the more general case where **nominal, ordinal, interval-scaled** and **ratio-scaled** features **co-exist**, we treat each one of them separately, as follows:

# Proximity measures between vectors

## (D) Mixed-valued vectors –similarity measures (SMs)

The similarity between  $\mathbf{x}$  and  $\mathbf{y}$  is defined as:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^l s_k(x_k, y_k)}{\sum_{k=1}^l w_k}$$



$\mathbf{x} = [x_1, \dots, x_l]^T$   
 $\mathbf{y} = [y_1, \dots, y_l]^T$

where:

- $w_k = 0$ , if **at least one** of  $x_k$  and  $y_k$  is **undefined** or (optionally) both  $x_k$  and  $y_k$  are equal to 0. Otherwise  $w_k = 1$ .
- If  $x_k$  and  $y_k$  are **binary**,  $s_k(x_k, y_k) = \begin{cases} 1, & \text{if } x_k = y_k = 1 \text{ (or } x_k = y_k) \\ 0, & \text{otherwise} \end{cases}$
- If  $x_k$  and  $y_k$  are **nominal** or **ordinal**,  $s_k(x_k, y_k) = \begin{cases} 1, & x_k = y_k \\ 0, & \text{otherwise} \end{cases}$
- If  $x_k$  and  $y_k$  are **interval** or **ratio scaled**-valued

$$s_k(x_k, y_k) = 1 - \frac{|x_k - y_k|}{r_k}$$

This is the **overlap measure**. Other options can also be used.

where  $r_k$  is the width of the interval where the  $k$ -th coordinates of the vectors of  $\mathbf{X}$  lie.

# Proximity measures between vectors

## (D) Mixed-valued vectors –similarity measures (SMs)

**Exercise 2:** Consider the data set given in the following table. Each row corresponds to a vector and each column to a feature. The first three features are ratio scaled, the 4<sup>th</sup> one is nominal and the 5<sup>th</sup> one is ordinal. Utilizing the previous similarity measure, compute the similarities between any pair of feature vectors.

Company	1 <sup>st</sup> year budget	2 <sup>nd</sup> year budget	3 <sup>rd</sup> year budget	Activity abroad	Rate of services 0: not good 1: good 2: very good
1 ( $x_1$ )	1.2	1.5	1.9	0	1
2 ( $x_2$ )	0.3	0.4	0.6	0	0
3 ( $x_3$ )	10	13	15	1	2
4 ( $x_4$ )	6	6	7	1	1

# Proximity measures between vectors

## Fuzzy measures – an alternative perspective

- Let  $\mathbf{x} \in [0,1]^l$ .
- In this context,  $x_k$  is **not** the **outcome** of a **measuring device**.
- Rather, it **indicates** the **degree** to which  $\mathbf{x}$  **possesses** the  $k$ -th characteristic.
- The closer the  $x_k$  to **1** (**0**), the **more likely** is that  $\mathbf{x}$  **possesses** (**does not possess**) the  $k$ -th characteristic.
- As  $x_k$  **approaches 0.5**, the **certainty** about the **possession or not** of the  $i$ -th feature from  $\mathbf{x}$  decreases.
- Let

$$\mathbf{x} = [x_1, x_2, \dots, x_k, \dots, x_l]^T \in [0,1]^l$$

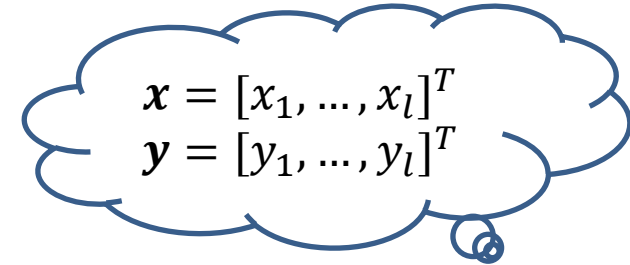
$$\mathbf{y} = [y_1, y_2, \dots, y_k, \dots, y_l]^T \in [0,1]^l$$

- A **measure of similarity between**  $x_k$  and  $y_k$  is the following

$$s(x_k, y_k) = \max(\min(1 - x_k, 1 - y_k), \min(x_k, y_k))$$

Then, as **measure of similarity between**  $\mathbf{x}$  and  $\mathbf{y}$  we can use the following

$$s^q(\mathbf{x}, \mathbf{y}) = \left( \sum_{k=1}^l s(x_k, y_k)^q \right)^{1/q}, \quad q \in [1, +\infty)$$



A thought bubble containing the definitions of vectors  $\mathbf{x}$  and  $\mathbf{y}$ .

$$\mathbf{x} = [x_1, \dots, x_l]^T$$
$$\mathbf{y} = [y_1, \dots, y_l]^T$$

# Proximity measures between vectors

## Fuzzy measures – an alternative perspective

**Exercise 3:** Let  $l = 3$  and  $q = 1$ .

(a) Consider the vectors  $\mathbf{x}_1 = [1,1,1]^T$ ,  $\mathbf{x}_2 = [0,0,1]^T$ ,  $\mathbf{x}_3 = [\frac{1}{2}, \frac{1}{3}, \frac{1}{4}]^T$ ,  
 $\mathbf{x}_4 = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]^T$ . Determine the similarities  $s^1(\mathbf{x}_i, \mathbf{x}_j)$ ,  $i, j = 1,2,3,4$ .

(b) Consider the vectors  $\mathbf{y}_1 = [\frac{3}{4}, \frac{3}{4}, \frac{3}{4}]^T$ ,  $\mathbf{y}_2 = [1,1,1]^T$ ,  $\mathbf{y}_3 = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}]^T$ ,  
 $\mathbf{y}_4 = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]^T$ . Determine the similarities  $s^1(\mathbf{y}_i, \mathbf{y}_j)$ ,  $i, j = 1,2,3,4$ ,  $i \neq j$ .

(c) Draw your conclusions.

# Proximity measures between vectors

## Dynamic similarity measures

- These are useful for cases where **the two vectors** to be compared have **different lengths**.
- Such a situation may arise e.g., when **comparing two strings** stemming **from two different texts**.
- A simple example: The **Edit distance**.

# Proximity measures between vectors – Missing data

## Missing data

- For **some vectors** of the data set  $X$ , **some features values** are **unknown**.
- This issue arises **very often** in **practice**.
- It may be caused by a measurement device failure, inability to take measure due to specific physical conditions etc.
- Ways to deal with this situation:
  - ✓ **Discard** all **vectors** with **missing values** (not recommended for small data sets).
  - ✓ **Find** the mean value  $m_k$  of the **available  $k$ -th feature values** over that data set and **substitute** the **missing  $k$ -th feature values** with  $m_k$ .



# Proximity measures between vectors – Missing data

## Missing data

- Ways to deal with this situation:

- ✓ Define  $b_k = 0$ , if **both** the  $k$ -th features  $x_k, y_k$  are **available** and **1 otherwise**. Then

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

where  $\phi(x_k, y_k)$  denotes the **proximity measure** between two scalars  $x_k, y_k$ .

**NOTE:** The **proximity** is **based only on the features** for which both  $x_k, y_k$  are **available**.

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

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

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

# Proximity measures between vectors – Missing data

## Missing data

**Exercise 4:** Consider the data set  $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$ , with  $\mathbf{x}_1 = [0,0]^T$ ,  $\mathbf{x}_2 = [1,*]^T$ ,  $\mathbf{x}_3 = [0,*]^T$ ,  $\mathbf{x}_4 = [2,2]^T$ ,  $\mathbf{x}_5 = [3,1]^T$  (“\*” stands for **missing values**).

- (a) Compute the  $l_1$  distances between all pairs of vectors, using all the four techniques for dealing with missing data.
- (b) In which of these techniques, the computed distances are dependent on the specific data set?

# Clustering algorithms

## Number of possible clusterings

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

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

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

## Examples:

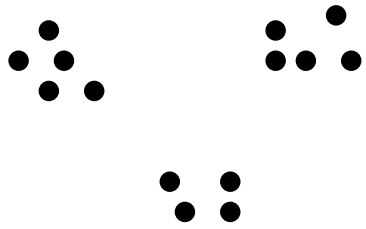
- $S(15,3) = 2,375,101$
- $S(20,4) = 45,232,115,901$
- $S(25,8) = 690,223,721,118,368,580$
- $S(100,5) \approx 10^{68}!!$

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

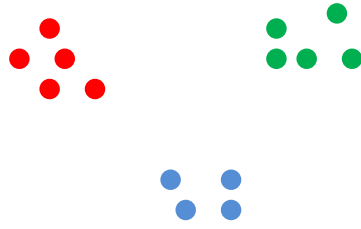
⇒

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

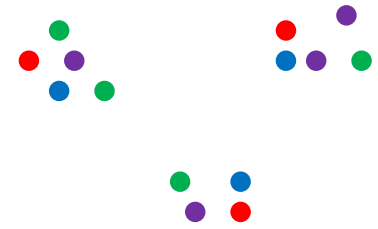
# Clustering algorithms



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 (BSAS)

- $m = 1$  \{number of clusters\}

- $C_m = \{x_1\}$

- For  $i = 2$  to  $N$

- Find  $C_k$ :  $d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$

- If  $(d(x_i, C_k) > \theta)$  AND  $(m < q)$  then

- $m = m + 1$

- $C_m = \{x_i\}$

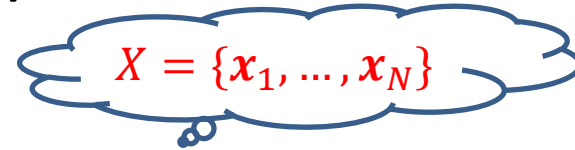
- Else

- $C_k = C_k \cup \{x_i\}$

- Where necessary, update representatives (\*)

- End {if}

- End {for}


$$X = \{x_1, \dots, x_N\}$$

-----

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

$$m_C^{new} = (n_C m_C^{old} + x) / (n_C + 1)$$

# Sequential clustering algorithms

## Basic Sequential Clustering Algorithm (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  $x$  is **reached prior** to the **final cluster formation**.
- **BSAS** perform a **single pass** on the data. Its complexity is  $O(N)$  (when point representatives are used).
- If clusters are represented by **point representatives**, **compact clusters** are favored.

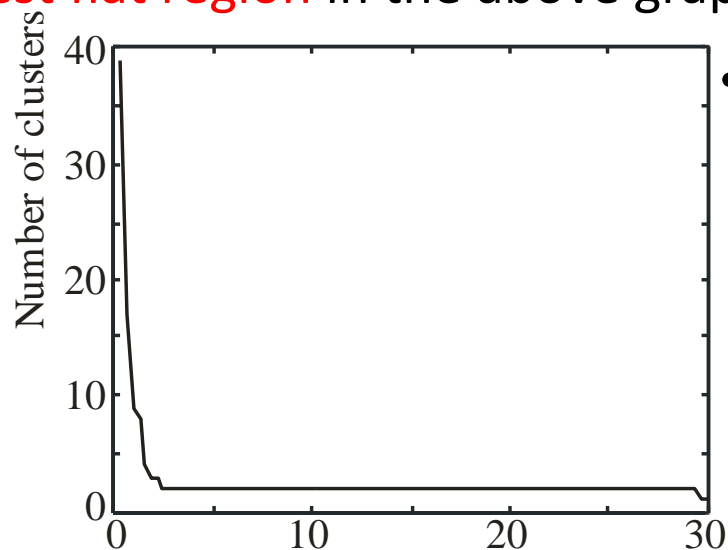
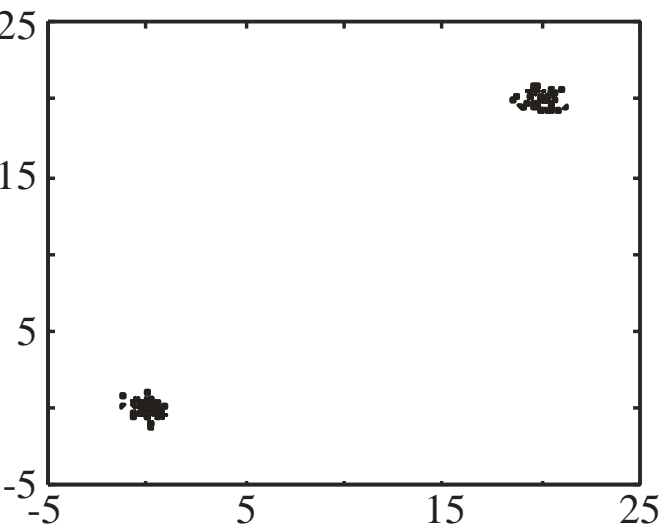
# Sequential clustering algorithms

## Basic Sequential Clustering Algorithm (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  $x$  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  $x$  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  $d(C_i, C_j) = \min_{k,r=1,\dots,m,k \neq r} d(C_k, C_r)$

**If**  $d(C_i, C_j) \leq M_1$  then  $\{ M_1 \text{ is a user-defined threshold} \}$

–**Merge**  $C_i, C_j$  to  $C_i$  and eliminate  $C_j$ .

–If necessary, update the cluster representative of  $C_i$ .

–Rename the clusters  $C_{j+1}, \dots, C_m$  to  $C_j, \dots, C_{m-1}$ , 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  $\mathbf{x}$  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  $\mathbf{x}$ ”

### A simple reassignment procedure

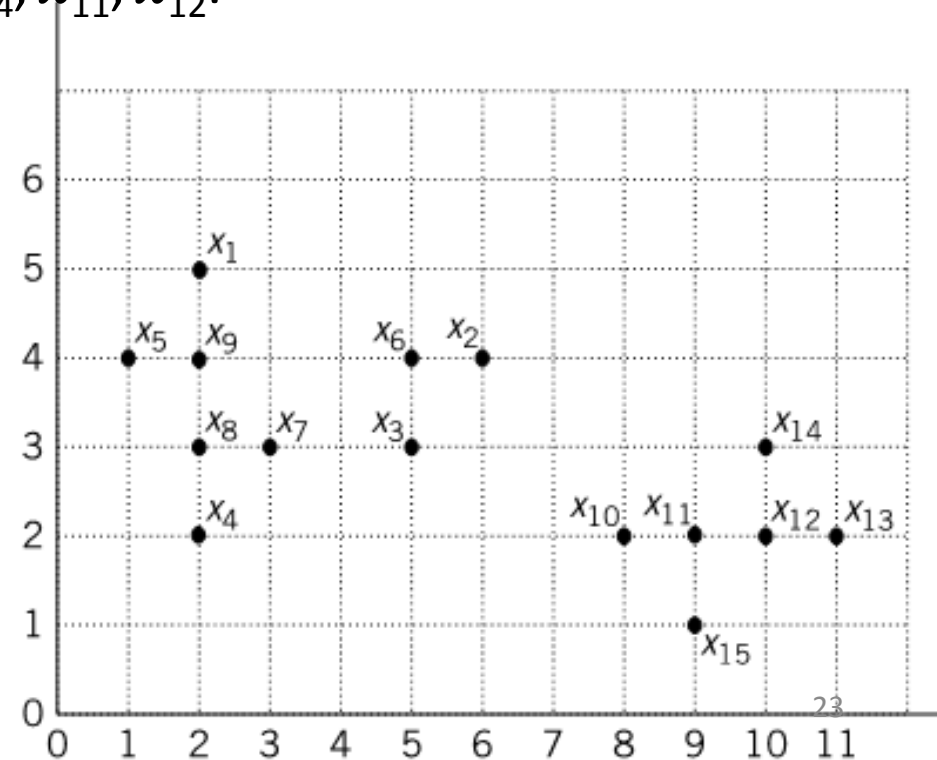
- **For**  $i = 1$  to  $N$ 
  - **Find**  $C_j$  such that  $d(\mathbf{x}_i, C_j) = \min_{k=1, \dots, m} d(\mathbf{x}_i, C_k)$
  - **Set**  $b(i) = j$  \{  $b(i)$  is the index of the cluster that lies closest to  $\underline{x}_i$  \}
- **End** {for}
  
- **For**  $j = 1$  to  $m$ 
  - **Set**  $C_j = \{\mathbf{x}_i \in X: b(i) = j\}$
  - If necessary, update representatives
- **End** {for}

# 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  $x_8, x_6, x_{11}, x_1, x_5, x_2, x_3, x_4, x_7, x_{10}, x_9, x_{12}, x_{13}, x_{14}, x_{15}$ , for  $\theta = 2.5$  and  $q = 15$ .
2. Repeat step 1, now with the order of presentation to the algorithm as  $x_7, x_3, x_1, x_5, x_9, x_6, x_8, x_4, x_2, x_{10}, x_{15}, x_{13}, x_{14}, x_{11}, x_{12}$ .
3. Repeat step 1, now with  $\theta = 1.4$ .
4. Repeat step 1, now with  $q = 2$ .



# Sequential clustering algorithms

## Example in MATLAB 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**  $\mathbf{m}_1 = [0, 0]^T$ ,  $\mathbf{m}_2 = [4, 0]^T$ ,  $\mathbf{m}_3 = [0, 4]^T$ ,  $\mathbf{m}_4 = [5, 4]^T$ , respectively, and respective **covariance matrices**  $S_1 = I$ ,  $S_2 = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1.5 \end{bmatrix}$ ,  $S_3 = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1.1 \end{bmatrix}$ ,  $S_4 = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.5 \end{bmatrix}$ . Then do the following:

1. Determine the number of clusters formed in  $X_1$  by doing the following:
  - a. Determine the maximum,  $d_{max}$ , and the minimum,  $d_{min}$ , distances between any two points in the data set.
  - b. Determine the values of  $\Theta$  for which the BSAS will run. These may be defined as  $\Theta_{min}, \Theta_{min} + s, \Theta_{min} + 2s, \dots, \Theta_{max}$ , 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  $\mathbf{m}_{tot}$  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.