

Clustering algorithms

Konstantinos Koutroumbas

Unit 3

- Missing data issue
- Categories of clustering algorithms
- Sequential clustering algorithms

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?

Proximity functions between a point and a set

Remark: Having in mind that a **cluster** is actually a set C , a **proximity function** between a point \mathbf{x} and a set C actually **quantifies** the **resemblance/relation** of \mathbf{x} with the cluster C .

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and $\mathbf{x} \in X, C \subset X$

Definitions of $\wp(\mathbf{x}, C)$:

(a) All points of C contribute to the definition of $\wp(\mathbf{x}, C)$.

- **Max** proximity function

$$\wp^{ps}_{max}(\mathbf{x}, C) = \max_{\mathbf{y} \in C} \wp(\mathbf{x}, \mathbf{y})$$

- **Min** proximity function

$$\wp^{ps}_{min}(\mathbf{x}, C) = \min_{\mathbf{y} \in C} \wp(\mathbf{x}, \mathbf{y})$$

- **Average** proximity function

$$\wp^{ps}_{avg}(\mathbf{x}, C) = \frac{1}{n_C} \sum_{\mathbf{y} \in C} \wp(\mathbf{x}, \mathbf{y})$$

n_C is the **cardinality** of C .

Proximity functions between a point and a set

Definitions of $\wp(\mathbf{x}, C)$ (cont.):

(b) A **representative** of C , r_C , **contributes** to the definition of $\wp(\mathbf{x}, C)$.

In this case $\wp(\mathbf{x}, C) = \wp(\mathbf{x}, r_C)$

Typical **point representatives** are:

- The **mean vector**

$$\mathbf{m}_p = \frac{1}{n_C} \sum_{\mathbf{y} \in C} \mathbf{y}$$

n_C is the **cardinality** of C .

- The **mean center**

$$\mathbf{m}_C \in C: \sum_{\mathbf{y} \in C} d(\mathbf{m}_C, \mathbf{y}) \leq \sum_{\mathbf{y} \in C} d(\mathbf{z}, \mathbf{y}), \forall \mathbf{z} \in C$$

- The **median center**

$$\mathbf{m}_{med} \in C: \text{med}(d(\mathbf{m}_{med}, \mathbf{y}) | \mathbf{y} \in C) \leq \text{med}(d(\mathbf{z}, \mathbf{y}) | \mathbf{y} \in C), \forall \mathbf{z} \in C$$

d : dissimilarity
measure.

Proximity functions between a point and a set

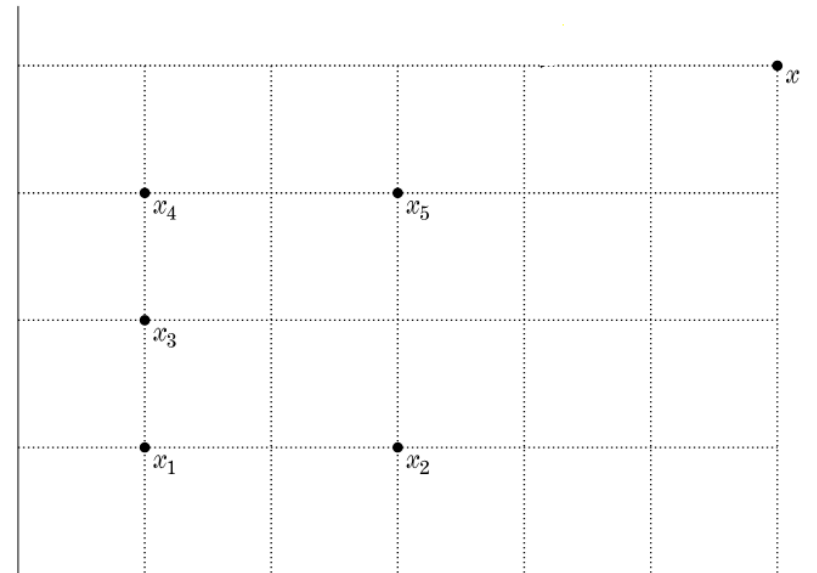
Definitions of $\wp(\mathbf{x}, C)$ (cont.):

(b) A **representative** of C , r_C , **contributes** to the definition of $\wp(\mathbf{x}, C)$.

In this case $\wp(\mathbf{x}, C) = \wp(\mathbf{x}, r_C)$

Exercise 5: Let $C = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$, where $\mathbf{x}_1 = [1,1]^T$, $\mathbf{x}_2 = [3,1]^T$, $\mathbf{x}_3 = [1,2]^T$, $\mathbf{x}_4 = [1,3]^T$, $\mathbf{x}_5 = [3,3]^T$. All points lie in the discrete space $\{0,1,2, \dots, 6\}^2$. Use the Euclidean distance to measure the dissimilarity between two vectors in C .

- (a) Determine the **mean vector**, the **mean center** and the **median center** of C .
- (b) Compute the distance of point $\mathbf{x} = [6,4]^T$ from C using the above defined representatives (where it is valid).



Proximity functions between a point and a set

Definitions of $\wp(\mathbf{x}, C)$ (cont.):

(b) A **representative** of C , r_C , **contributes** to the definition of $\wp(\mathbf{x}, C)$.

In this case $\wp(\mathbf{x}, C) = \wp(\mathbf{x}, r_C)$

Linear-shaped clusters:

- Such clusters occur e.g., in computer vision applications.
- In this case, a **hyperplane** is a **better representative** of such clusters
- Equation of a hyperplane H :

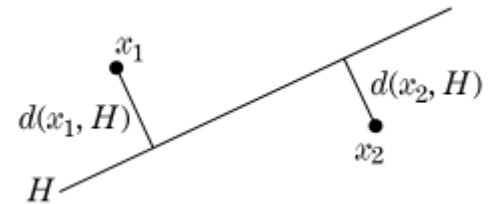
$$\sum_{j=1}^l a_j x_j + a_0 = \mathbf{a}^T \mathbf{x} + a_0 = 0$$

where $\mathbf{x} = [x_1, x_2, \dots, x_l]^T$, $\mathbf{a} = [a_1, a_2, \dots, a_l]^T$ is the **direction vector** of H and a_0 is its **offset**.

- **Distance** of a point \mathbf{x} from H : $d(\mathbf{x}, H) = \min_{\mathbf{z} \in H} d(\mathbf{x}, \mathbf{z})$
- If $d(\mathbf{x}, \mathbf{z})$ is the **Euclidean distance**, it is

$$d(\mathbf{x}, H) = \frac{|\mathbf{a}^T \mathbf{x} + a_0|}{\|\mathbf{a}\|}$$

$$\|\mathbf{a}\| = \sqrt{\sum_{j=1}^l a_j^2}$$



Proximity functions between a point and a set

Definitions of $\wp(\mathbf{x}, C)$ (cont.):

(b) A **representative** of C , r_C , **contributes** to the definition of $\wp(\mathbf{x}, C)$.

In this case $\wp(\mathbf{x}, C) = \wp(\mathbf{x}, r_C)$

Hyperspherical clusters:

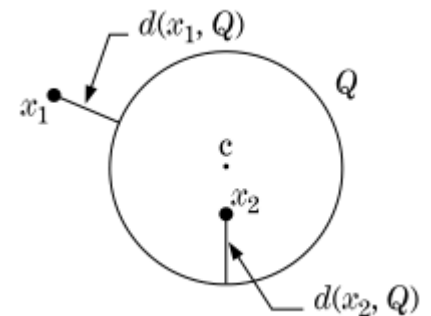
- Such clusters occur e.g., in computer vision applications.
- In this case, a **hypersphere** is a **better representative** of such clusters
- Equation of a hypersphere Q :

$$(\mathbf{x} - \mathbf{c})^T (\mathbf{x} - \mathbf{c}) = r^2$$

where $\mathbf{x} = [x_1, x_2, \dots, x_l]^T$, $\mathbf{c} = [c_1, c_2, \dots, c_l]^T$ is the **center** of Q and r is its **radius**.

• **Distance** of a point \mathbf{x} from Q : $d(\mathbf{x}, Q) = \min_{\mathbf{z} \in Q} d(\mathbf{x}, \mathbf{z})$

• For **Euclidean distance** between two points, $d(\mathbf{x}, Q)$ has a **geometric insight**.



• However, other **non-geometric** alternatives have also been proposed.

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:

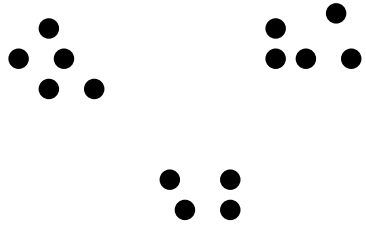
- $S(15,3) = 2,375,101$
- $S(20,4) = 45,232,115,901$
- $S(25,8) = 690,223,721,118,368,580$
- $S(100,5) \approx 10^{68}!!$

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

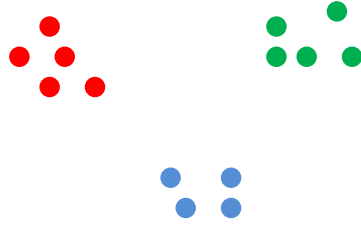
⇒

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

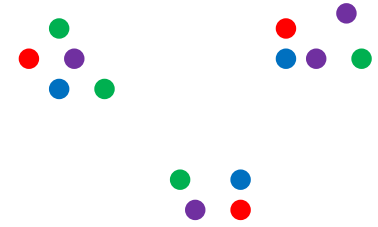
Clustering algorithms



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 θ associated with the distance.

Sequential clustering algorithms

Basic Sequential Clustering Algorithm (BSAS)

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

- $C_m = \{\mathbf{x}_1\}$

- **For** $i = 2$ to N

- **Find** C_k : $d(\mathbf{x}_i, C_k) = \min_{1 \leq j \leq m} d(\mathbf{x}_i, C_j)$

- **If** $(d(\mathbf{x}_i, C_k) > \theta)$ AND $(m < q)$ then

- $m = m + 1$

- $C_m = \{\mathbf{x}_i\}$

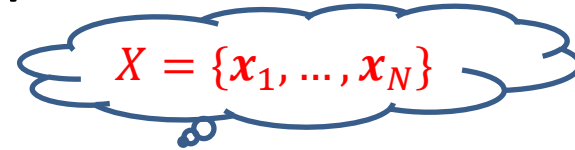
- **Else**

- $C_k = C_k \cup \{\mathbf{x}_i\}$

- Where necessary, update representatives (*)

- **End** {if}

- **End** {for}


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

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

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

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 θ .
- 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.

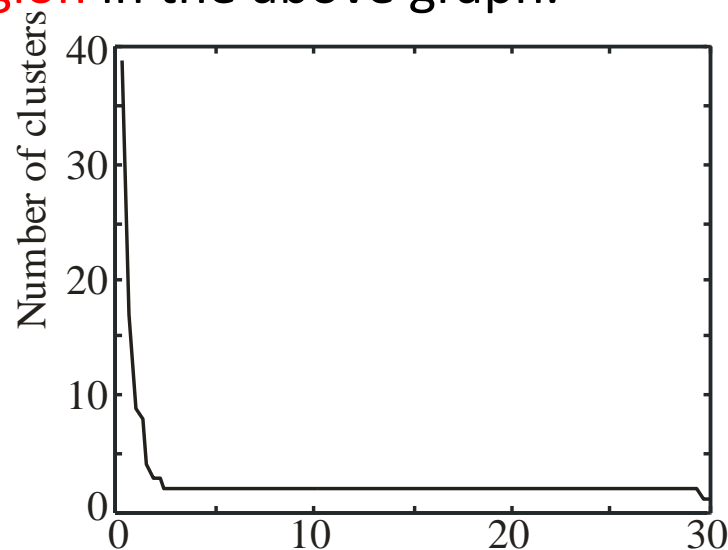
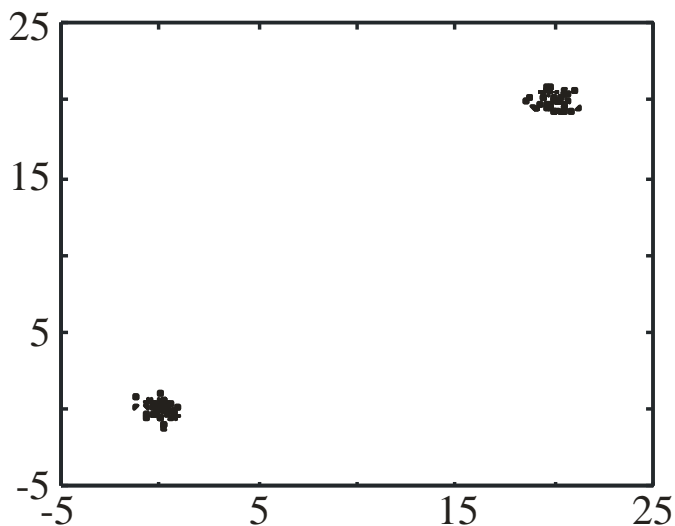
Sequential clustering algorithms

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 θ .

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



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 $\{M_1$ 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 \mathbf{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

A two-threshold sequential scheme (TTSAS)

- The formation of the clusters, as well as the assignment of vectors to clusters, is carried out concurrently (like BSAS and unlike MBSAS)
- **Two thresholds** θ_1 and θ_2 ($\theta_1 < \theta_2$) are **employed**.
- The **general idea** is the following:

If the distance $d(x, C)$ of x from its closest cluster, C , is **greater** than θ_2 then:

–A **new cluster** represented by 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 $x \in X - W$ determine $d_x = \min_{z \in W} d(x, z)$
- Determine y : $d_y = \max_{x \in X - W} d_x$
- If d_y is greater than a prespecified threshold (θ) then
 - 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:

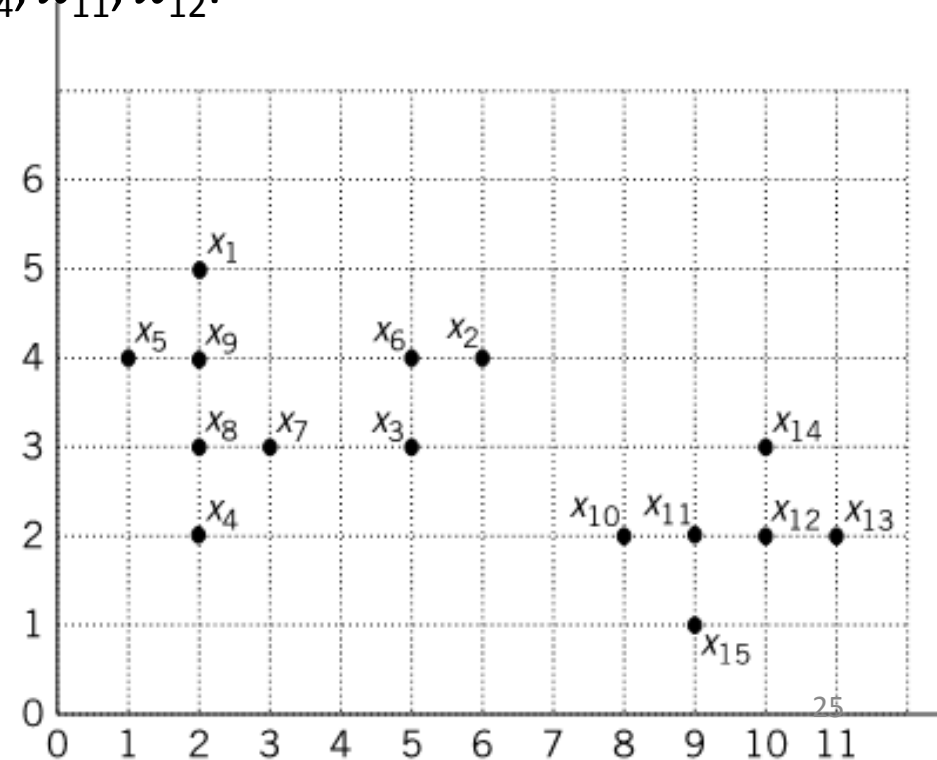
- The maxmin algorithm is more computationally demanding than MBSAS.
- However, it is expected to produce better clustering results than MBSAS.
- Its performance may be degraded in the presence of noise.

Sequential clustering algorithms

Example in MATLAB 1:

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

1. Apply the BSAS algorithm on X , presenting its elements in the order $x_8, x_6, x_{11}, x_1, x_5, x_2, x_3, x_4, x_7, x_{10}, x_9, x_{12}, x_{13}, x_{14}, x_{15}$, 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:

- Determine the maximum, d_{max} , and the minimum, d_{min} , distances between any two points in the data set.
- Determine the values of Θ 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_Θ is the number of successive values of Θ 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 Θ , run the BSAS algorithm $n_{times} = 10$, so that the data vectors are presented with different ordering to BSAS in each run. From the n_{times} estimates of the number of clusters, select the most frequently met value, m_{Θ} , as the most accurate. Let \mathbf{m}_{tot} be the n_{Θ} -dimensional vector, which contains the m_{Θ} values.
- d. Plot m_{Θ} versus Θ . Determine the widest flat region, r , of Θ 's (excluding the one that corresponds to the single-cluster case) and let n_r be the number of Θ 's in $\{\Theta_{min}, \Theta_{min} + s, \dots, \Theta_{max}\}$ that also lie in r . If n_r is "significant" (e.g., greater than 10% of n_{Θ}), the corresponding number of clusters, m_{best} , is selected as the best estimate and the mean of the values of Θ in r is chosen as the corresponding best value for Θ (Θ_{best}). Otherwise, the single-cluster clustering is adopted.

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

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