CHAPTER 3

CLUSTERING

1. Objectives of data clustering
2. Types of clustering techniques
3. Agglomerative techniques
4. Partition techniques
5. Clustering in Matlab
6. Bibliography

3.1. Objectives and characteristics of Clustering

Clustering objectives:

- division of a big data set into clusters (groups) of similar objects

- the objects of a cluster are similar among them and dissimilar from the objects of the other clusters.

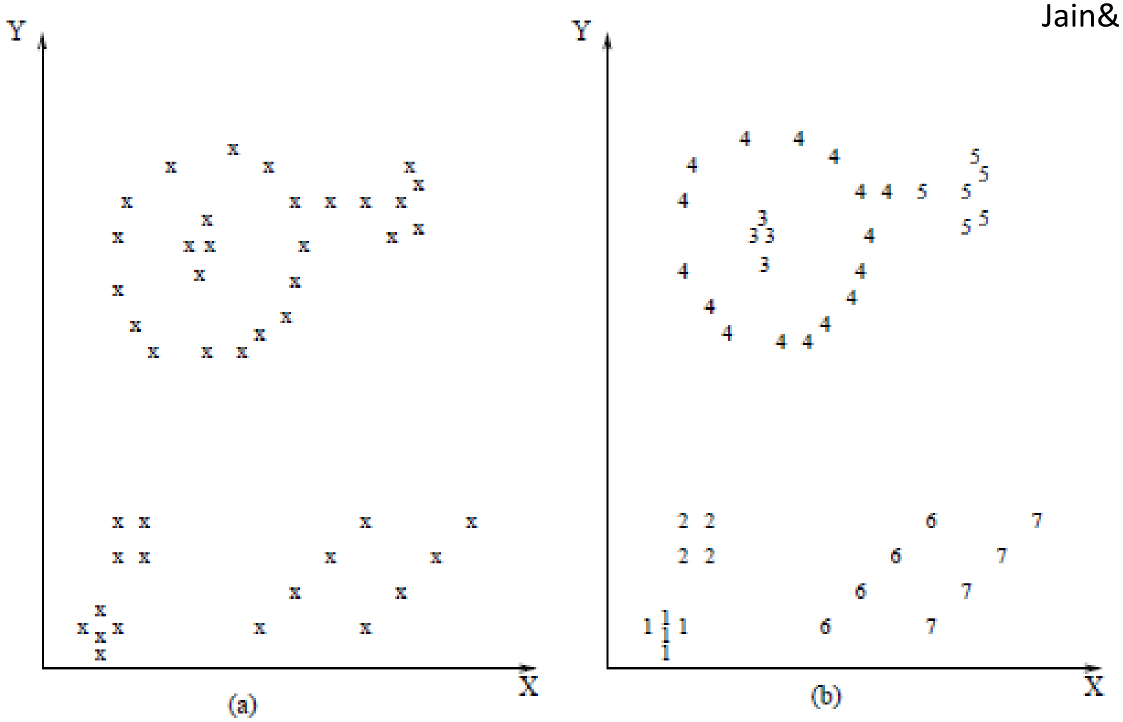
- the overall dataset can be represented by its clusters, in this way modelling the data.

- these clusters correspond to patterns hidden in the data.

- it is very used in the exploratory analysis of the data and to extract information from the data.

- it is an unsupervised learning technique of the structure embedded in the data.

Jain&Murty&Flynn



Original data

Clustering (hierarch., single-link)

Applications: image processing, diagnosis (classification of biological signals), epidemiologic studies, marketing, decision-making, etc..

Steps of a clustering study:

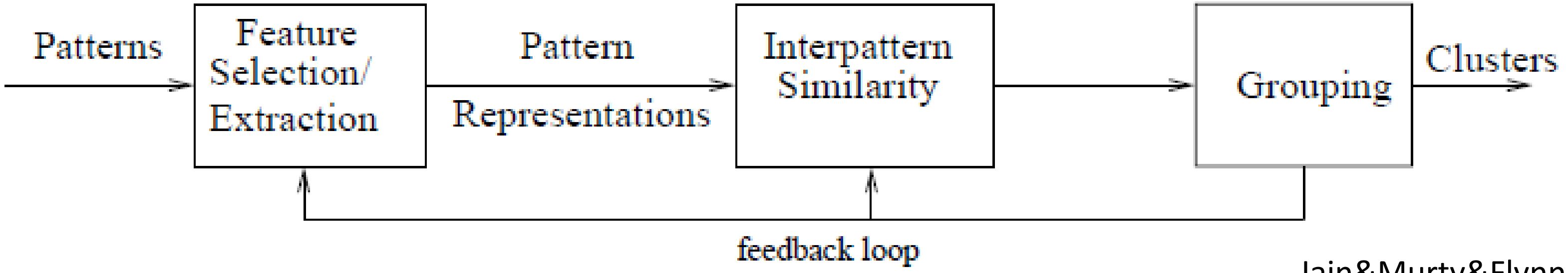
1- Features extraction and /or selection: select and extract the features from the data that define their patterns.

2- Define a similarity measure appropriate for the problem (usually a distance).

3- Cluster data with an appropriate algorithm.

4- Data abstraction: when necessary, the data are represented by their clusters.

5- Critical analysis of the results. Go to step 1 if results do not satisfy.



Jain&Murty&Flynn

Notation:

A pattern $\boldsymbol{x}$ is composed by a single data used by the clustering algorithm. $\boldsymbol{x}$ is composed by *d* measurements

$$\boldsymbol{x} = \{x_1 , x_2 , \ldots, x_d \}$$

- each scalar component x_d of $\boldsymbol{x}$ is a feature (characteristic) or attribute,
- *d* is the dimensionality of the pattern in the data space.

A pattern may measure a physical object (an apple, a table, a car, ...), a physiological state characterized by measurements of several biosignals, or an abstract concept such as writing style, painting, etc..

The characteristics or attributes can be:

quantitative : continuous (weight, height, temperature,…), discrete (number of patients) or by intervals (ex. duration of an event).

qualitative: nominal or unordered (ex. color), ordered (ex. position in a professional career).

In this course we will work with quantitative attributes.

Similarity measures (distances) between two points x_i and x_j in a multidimensional space

Euclidian distance: →

$$d(x_i - x_j) = \left[\sum_{k=1}^{d} (x_{ik} - x_{jk})^2\right]^{1/2}$$

squared Euclidian distance: →

$$d(x_i - x_j) = \left[\sum_{k=1}^{d} (x_{ik} - x_{jk})^2\right]$$

Manhattan / city-block distance: →

$$d(x_i - x_j) = \left[\sum_{k=1}^{d} | x_{ik} - x_{jk} |\right]$$

Chebychev distance: →

$$d(x_i - x_j) = \max_{k=1}^{d} \left| x_{ik} - x_{jk} \right|$$

Minkowsky distance: →

$$d(x_i - x_j) = \left[\sum_{k=1}^{d} (x_{ik} - x_{jk})^m\right]^{1/m}$$

There is no best measure in general. It depends on the dataset, the application, the objective. For an interesting comparison of measures see for ex.

https://towardsdatascience.com/log-book-guide-to-distance-measuring-approaches-for-k-means-clustering-f137807e8e21 11 September 2023

Matrix of patterns (data matrix)
$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}$$

d – number of features (dimensionality)
n – number of objects (patterns)

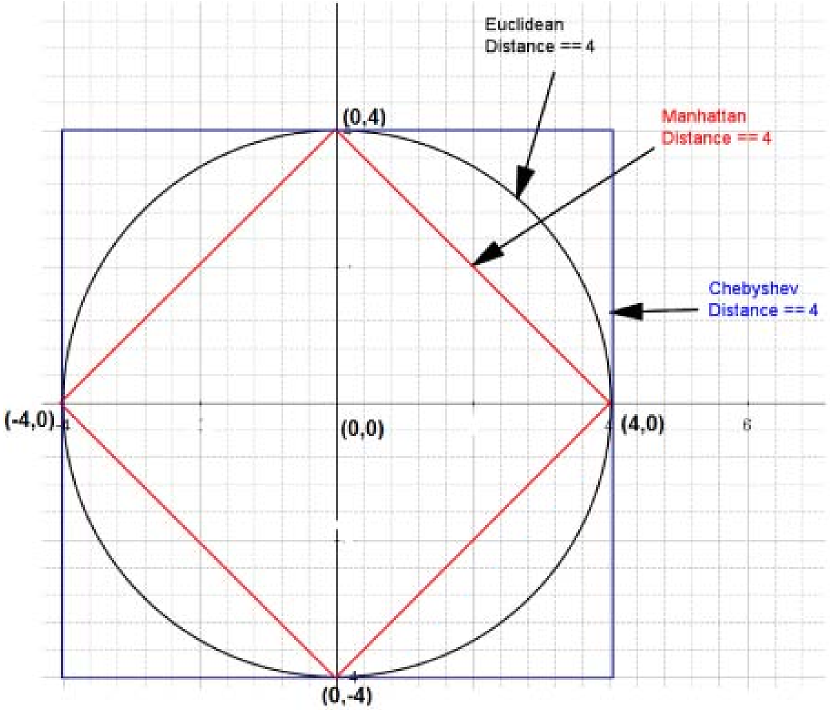
Similarity matrix, symmetric
$$\begin{bmatrix} 0 & d_{12} & d_{13} & \dots & d_{1n} \\ & 0 & d_{23} & \dots & d_{2n} \\ & & 0 & \dots & d_{3n} \\ : & : & : & : & : \\ & & & & 0 \end{bmatrix}$$

d_{ij} – distance between x_i and x_j
n – number of objects (patterns)

The shape of the clusters depends:

- on the used distance: for example, euclidian distance produce circular clusters (all points of the cluster can be put into a circle), Manhattan and Chebyshev produce shapes as in the figure.

- on the used method



https://towardsdatascience.com/log-book-guide-to-distance-measuring-approaches-for-k-means-clustering-f137807e8e21 11 Sept 2023

3.2. Types of Clustering techniques

Agglomerative:

initially each pattern defines a cluster, and the clusters are merged successively until some stopping criteria is reached.
ex. hierarchical clustering (upwards)

Divisive:

initially all the patterns belong to a single cluster, and a divisive technique is successively applied until some stopping criteria is reached.
ex. hierarchical clustering (downwards).

3.2. Types of Clustering techniques

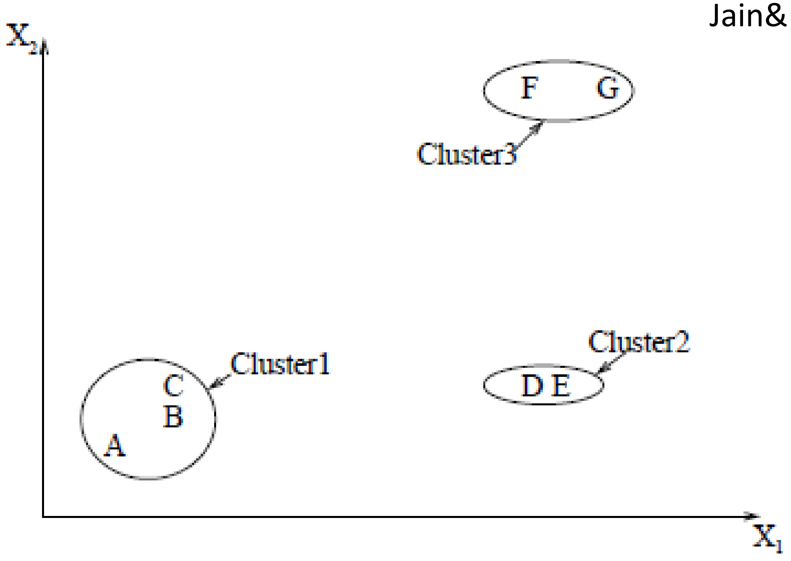
Partitional:

- **distance based**: a unique partition of the data, in a certain number of clusters, is obtained. Problem: how many clusters ?
  They result from the optimization of a local criteria (of a subset of points) or global criteria (of all points).
  Ex. k-means (or c-means), fuzzy c-means

- **density based:** the method itself finds the number of clusters by capturing the density of points in each region.
  Ex.: subtractive : based on a function of radiating potential of a point;
  DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

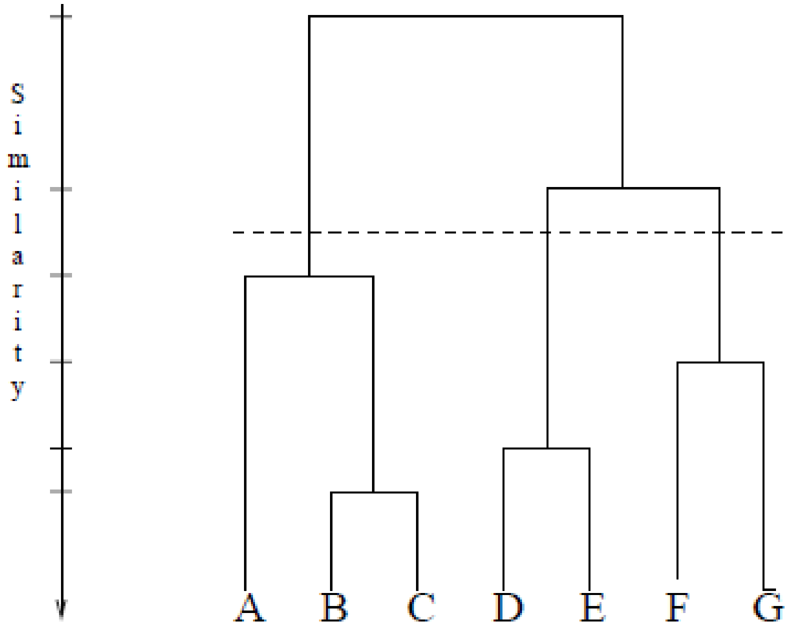
3.3. Agglomerative techniques

Hierarchical clustering



group 7 points into three clusters

Jain&Murty&Flynn



The number of clusters depends on the desired level of similarity

Dendrogram (from Greek *dendro* – tree, *gramma* - drawing) representing the chained clustering of the patterns and the similarity levels in which the groups change.

Algorithms for agglomerative hierarchical clustering

1. Initially make each object or pattern a cluster.
2. Compute the similarity (or proximity) matrix, containing the distances between each pair of clusters.
3. Find the most similar pair of clusters using the proximity matrix (those with shortest distance outside the diagonal). Merge these two clusters into one.
4. Update the proximity matrix resulting from this merging : one row and one column of the pre-merged matrix are deleted, and new distances between the merged cluster and each of the other clusters must be recomputed; the rest of the proximity matrix remains unchanged, its dimension decreases by one).
5. Are all objects into a single cluster , i.e., is the proximity matrix 0 ?   If yes stop. If no go to 2.

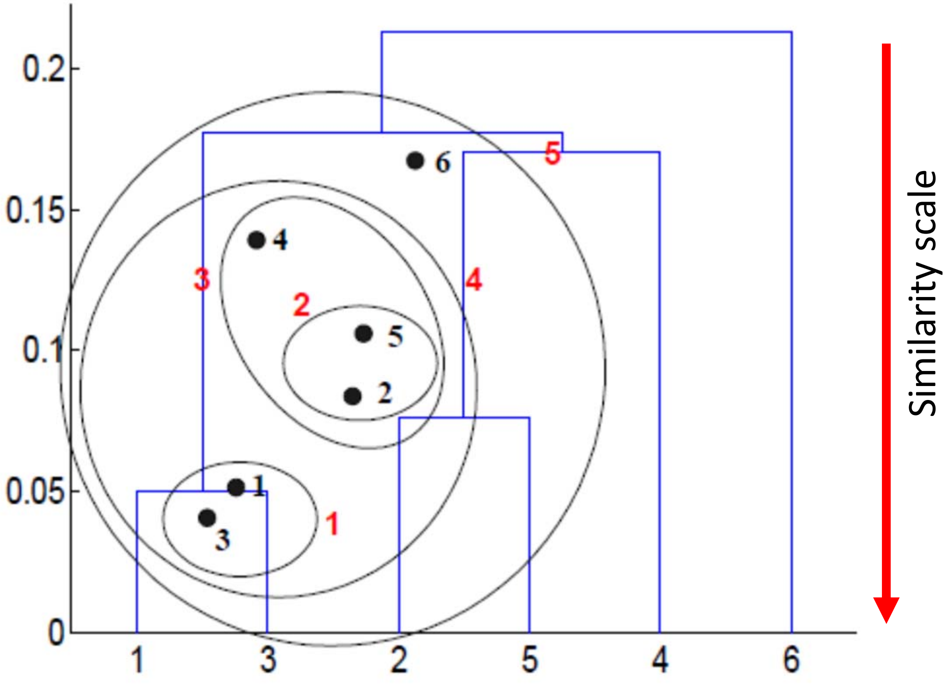
How is performed the update of the proximity matrix ? Or, equivalently, how to measure the distance between two clusters ? Each method gives an algorithm.

Advantages:

- Eases the interpretation and utilization of the results obtained.

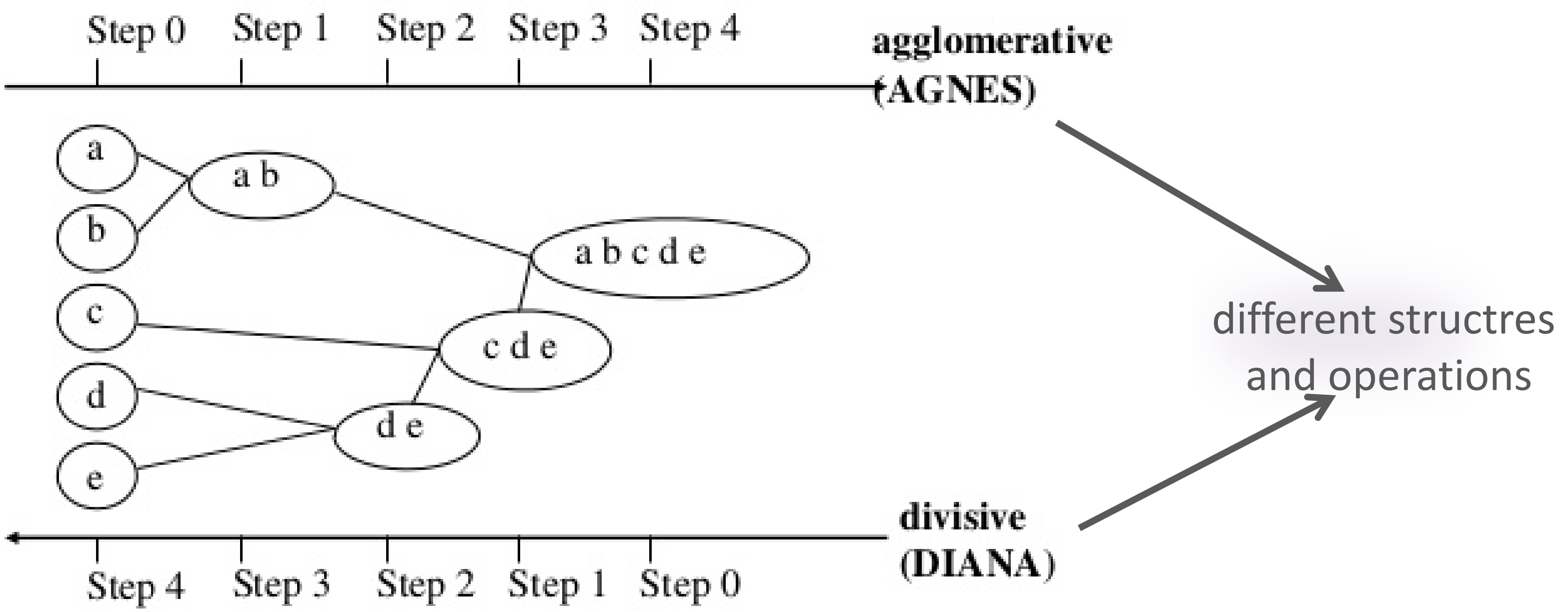
- Allows a great flexibility in the analysis of the clusters in the different levels of the dendrogram. We can stop at any number of clusters (flexibility)

- The task of affecting a concept (label) to the clusters becomes semi-automatic.



Rai P. and Singh S, A Survey of Clustering Techniques

Two approaches: bottom-up and top-down



adaptado de https://www.slideshare.net/salahecom/10-clusbasic 11 Sept 2023

- uses recursive processes

- **stopping criteria**: number *k* of wished *clusters*.

- The agglomerative method is more used than the divisive: it is computationally lighter and produces better results.

Hierarchical Clustering – AGNES AGlomerative NESting

1. At start each existent pattern composes a cluster.
2. Using an appropriate metric (ex. Euclidian), compute the distances between all pairs of clusters.
3. Find the pair of clusters more similar and merge these two into one single cluster.
4. Recalculate the distances between the new *cluster* and all the other clusters already existent.
5. Repeat step 3 until one single cluster is obtained, containing all of the patterns.

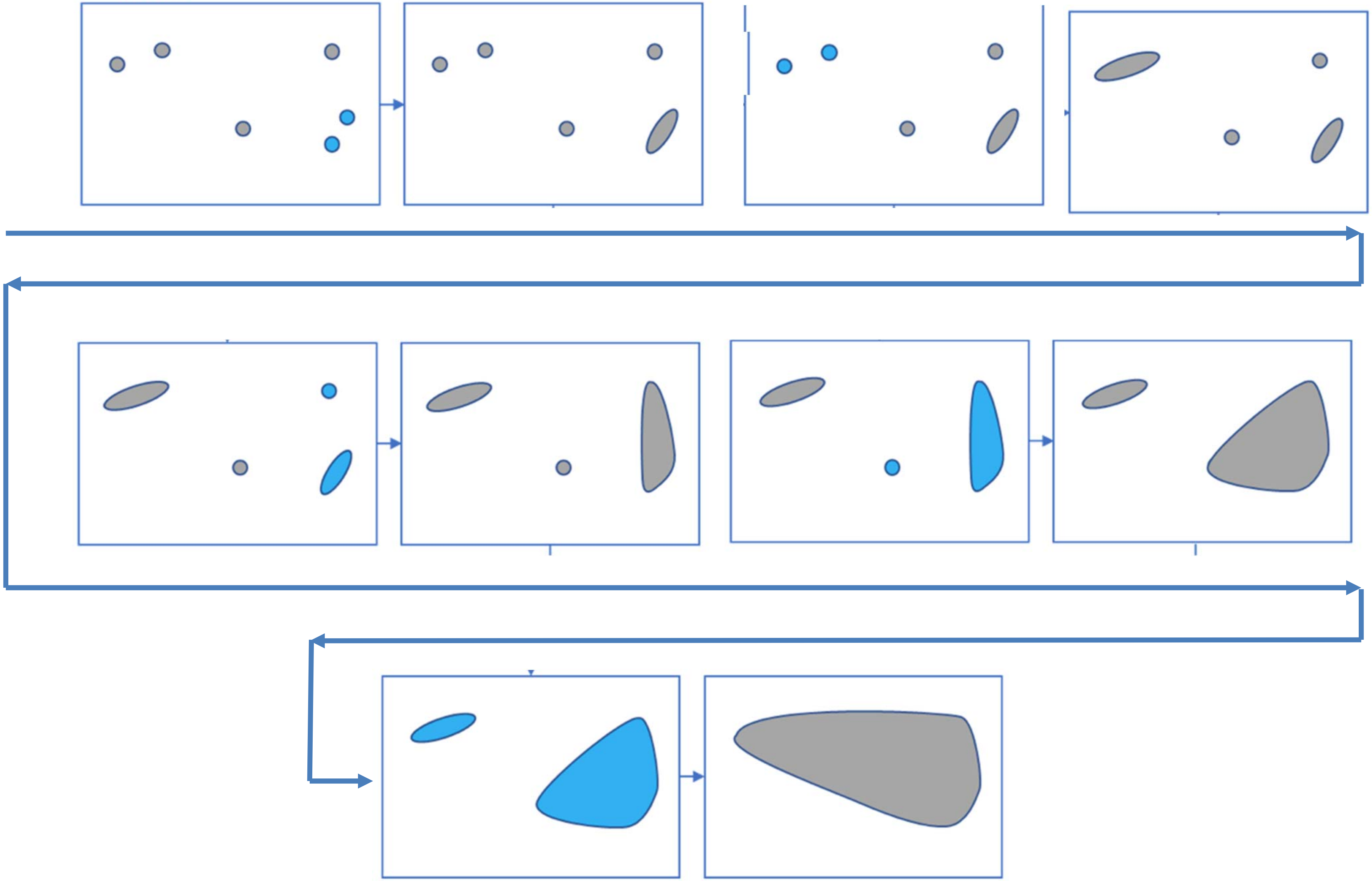
These steps repeat the development in slide 83 and are illustrated in the next slide.

AGNES

Identify the two clusters that are closest together

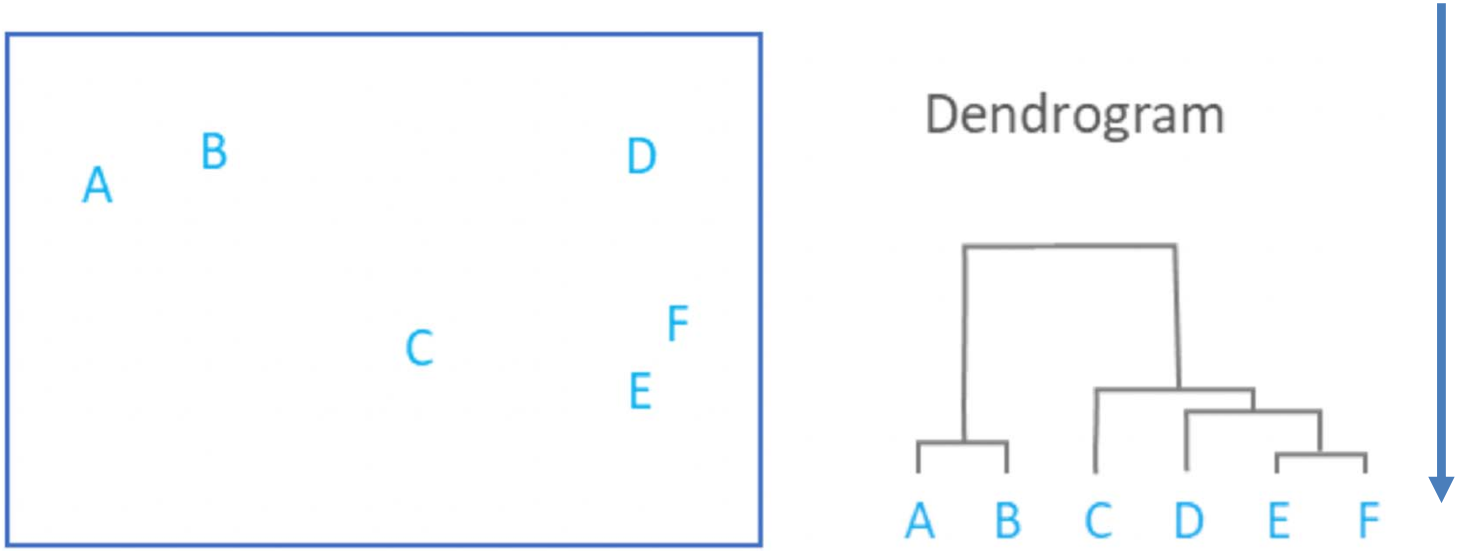
Merge the two most similar clusters

repeat ...



(adapted from https://www.displayr.com/what-is-hierarchical-clustering/ 11 Sept 2023)

Naming the points of previous example as A, B , C, D, E, F, we can build the dendrogram, a tree replicating the agglomerative operations of the previous slide.



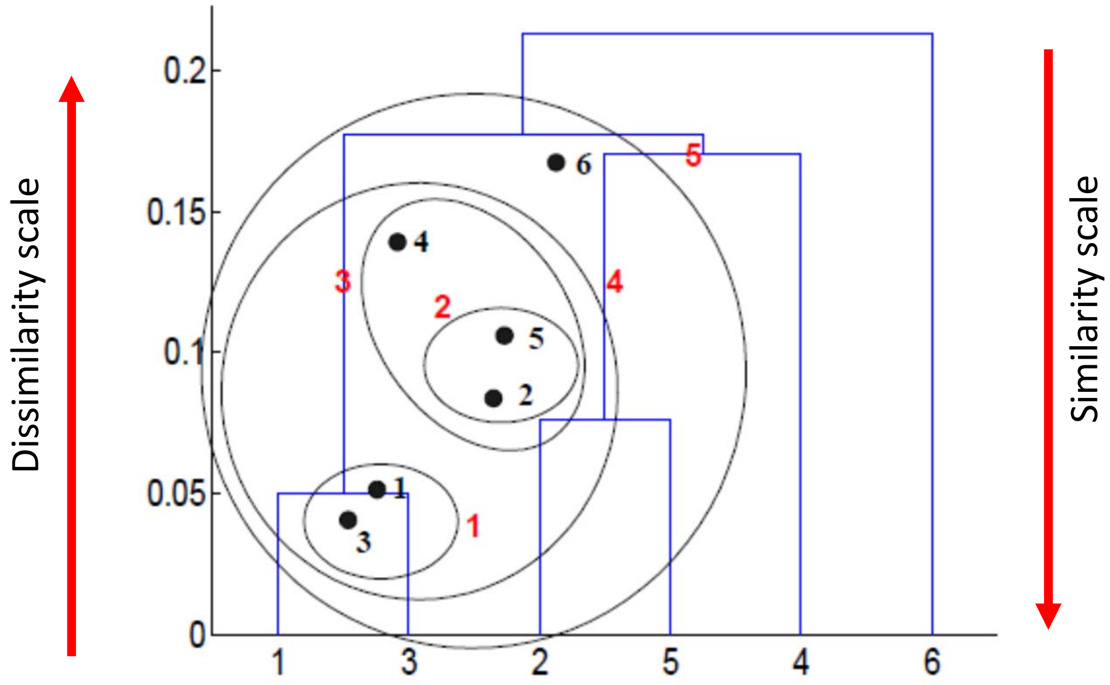
similarity level

(https://www.displayr.com/what-is-hierarchical-clustering/ 11 Sept 2023

The horizontal connection of the points follows the smallest distance in the previous figure. The similarity level of the horizontal connection is inverse to the minimum distance in last figure (smaller distance higher similarity)

The dendrogram has a hierarchical structure that accompanies the agglomerations done in the example. The level of the hierarchy is related to the level of similarity among clusters At each level, we can see how many clusters do we have.

Another example showing the agglomerative clustering and the building up pf the dendrogram.
Each node in the tree represents a cluster. Its height is proportional to the dissimilarities of its daughters (a daughter can be a single point or a cluster).



Dissimilarity scale

Similarity scale

Rai P. and Singh S, A Survey of Clustering Techniques

Cutting the tree at some height gives the number of cluster. For example, at 0.15 we have 4 clusters (same fig. as in the slide 84)..

Hierarchical Clustering – AGNES, agglomerative

How to compute the distance between two clusters ?

single-link

- Considers the **minimum** of the distances between all pairs of points (each element of the pair from each cluster).
- Produces clusters geometrically elongated: the *chaining effect*.

complete-link

- Considers the **maximum** of the distances between all pairs of points (each element of the pair from each cluster).
- Produces small and compact clusters with well defined frontiers.

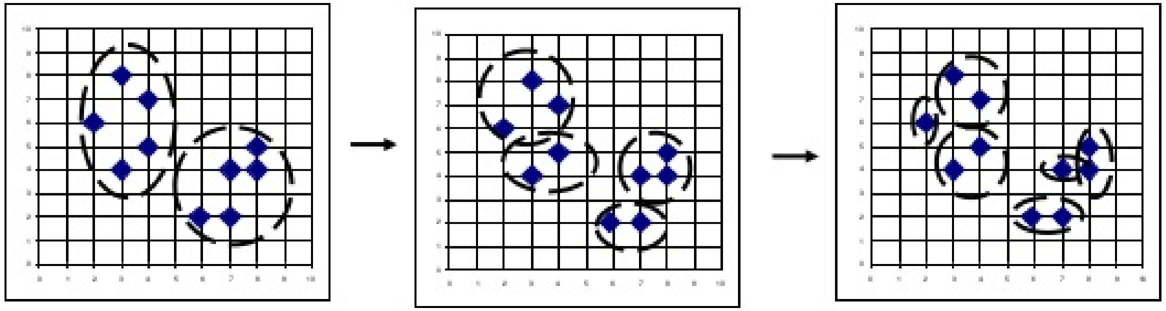
average-link**:** compromise between the *single-link* and the *complete-link*.

- Compute the **mean** of all the distances between all pairs of points (each element of the pair from each cluster).
- Produces quite acceptable results.

centroid-link: considers the distance between the centroids of the clusters.

See also https://www.stat.cmu.edu/~ryantibs/datamining/lectures/06-clus3.pdf 11/09/2027)

Hierarchical clustering – DIANA DIvisive ANAlysis



https://www.slideshare.net/salahecom/10-clusbasic 11/09/2023

1. All of the points are agglomerated into a cluster.
2. Divide that cluster in two new clusters.

-polytetic methods: All of the features of the data are used for the division.

-monotetic methods: Only one feature of the data is used for the division.

The process repeats. Find the most heterogeneous cluster to be divided into two: higher number of samples, higher variance, higher mean squared distance, ...

This heterogeneous cluster is divided.

While the number of clusters is not equal to the number of patterns in the sample, step 2 is repeated.

To divide a cluster use for example a cutting algorithm based on optimization ("Cut-based optimization", with similar results as the k- means). See more in https://www.cs.princeton.edu/courses/archive/spr08/cos435/Class_notes/clustering4.pdf 11/09/2023)

3. 4. Partitional techniques

- Do not produce a hierarchical structure of the data.
- Only one level of clusters is created.
- The result is the centers of the clusters and the final membership of each pattern to its cluster.

advantage:

- They are very useful when one needs to work with very big datasets for which the construction of a dendrogram is computationally complex.

disadvantage:

- It is necessary to chose, *a priori*, the number of desired clusters, for some of the methods.
- They do not produce the same result in each execution (because of different initializations).
- The final results depends on that choice and on the initialization.

algorithms: **k-means (or c-means), k-medoids ,** distance based
subtractive , DBSCAN, density based
and more ...

3.4.1. c (ou k) -means clustering

It organizes the data into *c* clusters, *c* being fixed a-priori.

Each cluster has at least one element.

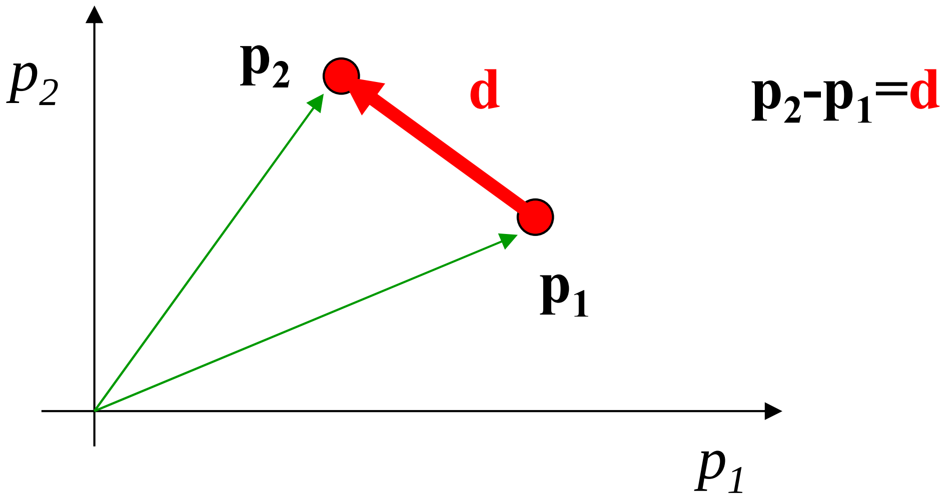
Each element belongs to one single cluster.

Each element is affected to the center that is closer.

The centers of the clusters are iteratively moved, from iteration to iteration, in order to decrease the distances to the elements of its cluster.

At the end, when the algorithm converges, the sum of the distances of the points of a cluster to its center is minimized and the distance between the centers is maximized.

Distance: euclidian



Criterion:

Minimize the sum of the squared distances between the elements and its centers, taken for all clusters..

The algorithm also maximizes the distances among the centers

Mathematical formulation :

$\mathbf{P} = \{\mathbf{p_1} \quad \mathbf{p_2} \quad \dots \quad \mathbf{p_Q}\}$ set of Q data

$\mathbf{p}_i = \{p_{i1} \quad p_{i2} \quad \dots \quad p_{iR}\}$ the R characteristics of each data point

$\mathbf{v}_i = [v_{i1} \quad v_{i2} \quad \dots \quad v_{iR}] \triangleq$ center of the cluster i

$$d(\mathbf{p}_k - \mathbf{v}_i) = \|\mathbf{p}_k - \mathbf{v}_i\| = \sqrt{\sum_{j=1}^{R} (p_{kj} - v_{ij})^2} = d_{ik},$$ euclidian distance

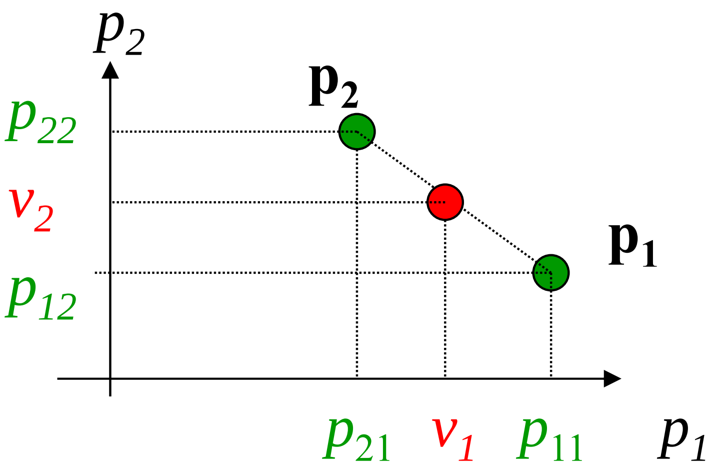
between each data point $\mathbf{p}_k$ and the center $\mathbf{v}_i$ of the cluster i

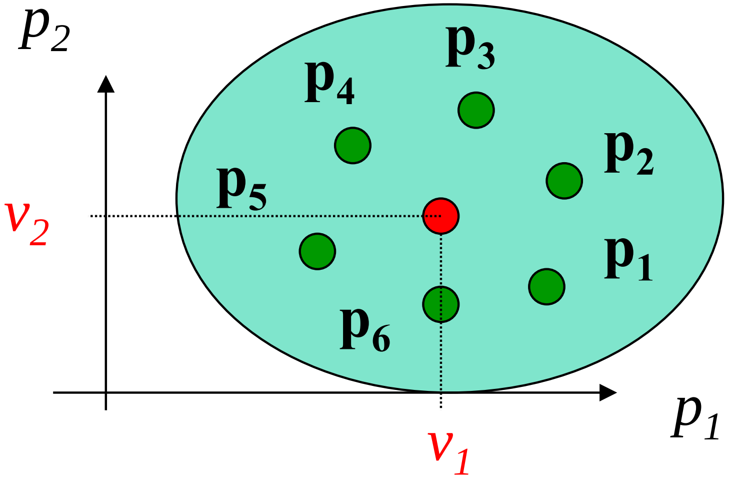
$$\psi_{ik} = \begin{cases} 1, & \text{if } \mathbf{p}_k \text{ belongs to the cluster } i \\ 0, & \text{if } \mathbf{p}_k \text{ does not belong to the cluster } i \end{cases}$$

Criterion to be minimized

$$J(\mathbf{P},\mathbf{v})=\sum_{k=1}^{Q}\sum_{i=1}^{C}\psi_{ik}(d_{ik})^{2}$$

How to compute the coordinates of a center ?





$$v_2 = \frac{p_{12} + p_{22}}{2}$$

$$v_1 = \frac{p_{11} + p_{21}}{2}$$

$$v_2 = \frac{p_{12} + p_{22} + p_{32} + p_{42} + p_{52} + p_{62}}{6}$$

$$v_1 = \frac{p_{11} + p_{21} + p_{31} + p_{41} + p_{51} + p_{61}}{6}$$

$$v_{ij} = \frac{\sum_{k=1}^{Q} \psi_{ik} p_{kj}}{\sum_{k=1}^{Q} \psi_{ik}}$$

Average of the coordinates, one by one, of the points belonging to the cluster *i*

Let $\{\mathbf{A}_i, i=1,2,\ldots,c\}$ be a family of sets, the *c*-partition of $\mathbf{P}$. The following properties apply:

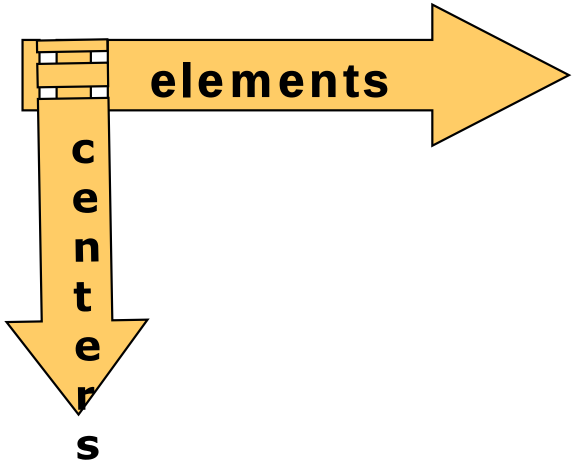
$$\bigcup_{i=1}^{c} \mathbf{A}_i = \mathbf{P} \qquad \bigvee_{i=1}^{c} \chi_{\mathbf{A}_i}(\mathbf{p}_k) = 1, \forall k$$

$$\mathbf{A}_i \cap \mathbf{A}_j = \varnothing, \forall i \neq j \qquad \chi_{\mathbf{A}_i}(\mathbf{p}_k) \wedge \chi_{\mathbf{A}_j}(\mathbf{p}_k) = 0, \forall k$$

$$\varnothing \subset \mathbf{A}_i \subset \mathbf{P}, \forall i \qquad 0 < \sum_{k=1}^{Q} \chi_{\mathbf{A}_i}(\mathbf{p}_k) < Q, \forall i$$

$$2 \leq c < Q \qquad \chi_{\mathbf{A}_i}(\mathbf{p}_k) = \psi_{ik} = \begin{cases} 1, \mathbf{p}_k \in \mathbf{A}_i \\ 0, \mathbf{p}_k \notin \mathbf{A} \end{cases}$$

$$\mathbf{U} = \begin{bmatrix} \psi_{11} & \psi_{12} & \psi_{1Q} \\ \psi_{21} & \psi_{22} & \psi_{2Q} \\ & & \\ \psi_{c1} & \psi_{c2} & \psi_{cQ} \end{bmatrix}_{c \times Q}$$



Matrix of *c*-partion of **P:** any matrix $\mathbf{M_c}$ such that:

$$\mathbf{M}_c = \{\mathbf{U} \mid \psi_{ij} \in \{0,1\}, \quad \sum_{i=1}^{c} \psi_{ik} = 1, \quad 0 < \sum_{k=1}^{Q} \psi_{ik} < Q$$

Sum of the elements of one column of **U**

Sum of the elements of one row of **U**

Any matrix with such structure defines a partition (division into clusters) of **P**. The possible number of partitions is (Ross):

$$\eta_{\mathbf{M}_c} = \left(\frac{1}{c!}\right)\left[\sum_{i=1}^{c}\binom{c}{i}(-1)^{c-i}.i^{Q}\right]$$

For Q=25, c=10 $\eta_{\mathbf{M}_c} = 10^{18}.$

NP-hard problem even for two clusters !!

Iterative optimization !

The batch c-means algorithm

1st Chose initial values of the c centers by a good sense criterion (or randomly).

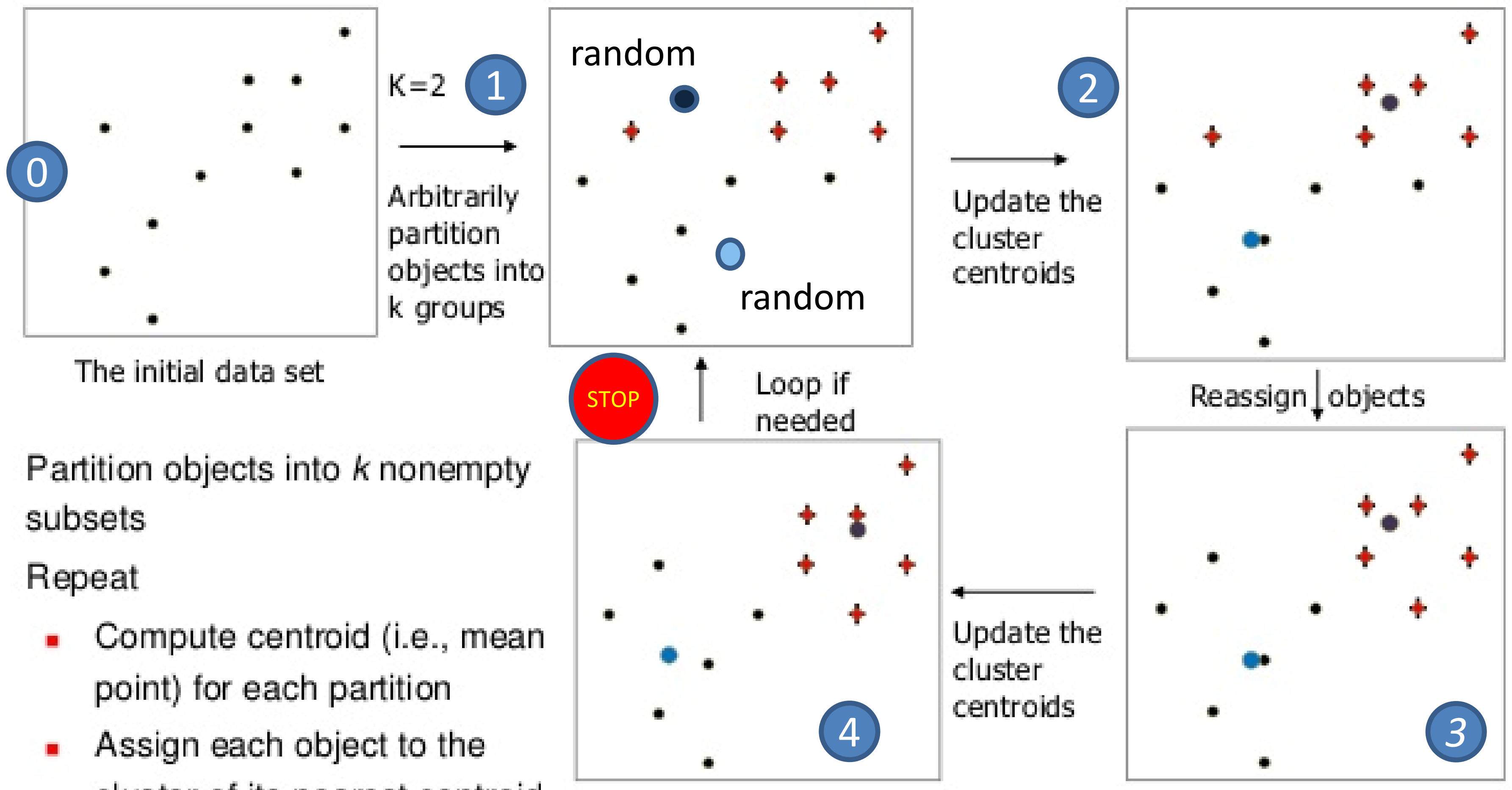
2nd Affect each point, from 1 to Q, to the center that is closest to it (Euclidian distance).

3rd Recalculate the coordinates of each center (averages of the coordinates of the points affected to it).

4th If the centers do not move, stop. End.

5th Otherwise go to 2nd.

The batch c-means algorithm


0
The initial data set
K=2
1
Arbitrarily partition objects into k groups
random
random
2
Update the cluster centroids
Reassign objects
3
Update the cluster centroids
4
Loop if needed
STOP

- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change , in this example after 4.

17

adapted from https://www.slideshare.net/salahecom/10-clusbasic 11/09/2023

The batch c-means algorithm pseudo code

```
Make initial (random) estimations of the centers $v_1, v_2, ...v_c$
While there is some change in any center do
    Use the actual centers to cluster the data (i.e., affect each point to the closest center according to a distance)
    For $i$ from 1 to $c$
        Replace $\mathbf{v}_i$ by the average of all points in the cluster $i$
    End_for
End_while
```

c-means sequential algorithm (in real time)

(https://www.cs.princeton.edu/courses/archive/fall08/cos436/Duda/C/sk_means.htm 11/09/2023)

Update the centers whenever a new point appears

1st Chose initial values of the c centers by a good sense criterion (or randomly).

2nd Acquire the next point **p** and update the center $\mathbf{v}_\mathbf{i}$ that is closer to it according to the number of points that already appeared in that region. For that, one needs to count them; if they are already n_i then

$$\boldsymbol{v}_i^{(k+1)}=\boldsymbol{v}_i^{(k)}+(1/n_i).(\boldsymbol{p}-\boldsymbol{v}_i^{(k)})$$

Each center needs a counter associated to it.

3º Continue until there are no more points to read.

c-means sequential algorithm (in real time) pseudo-code (Duda)

```
Make initial guesses for the centers $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_c$

Initialize the counters $n_1, n_2, \ldots, n_c$ to zero

Until the end do

    Acquire the next point $\boldsymbol{p}$

        If $\boldsymbol{v}_\mathrm{i}$ is the closest to $\boldsymbol{p}$

            increment $n_i$

            replace $\boldsymbol{v}_i$ by $\boldsymbol{v}_i + (1/n_\mathrm{i}).(\boldsymbol{p}-\boldsymbol{v}_i)$

        End_if

End_until
```

The resulting centers, $\boldsymbol{v}_\mathrm{i}$, are the averages of all points $\boldsymbol{p}$ that, when acquired, were closest to $\boldsymbol{v}_\mathrm{i}$

c-means sequential algorithm (in real time) pseudo code (Duda)

(without counters associated to the centers)

```
Make initial guesses for the centers $v_1, v_2, ...v_c$
 Until the end do
        Acquire the next point $\mathbf{p}$
            If $\mathbf{v}_\mathbf{i}$ is the closest to $\mathbf{p}$
                   replace $\mathbf{v}_i$ by $\mathbf{v}_i$ + $\alpha$ .$(\mathbf{p}-\mathbf{v}_\mathbf{i})$
            End_if
 End_until
```

The centers have the following evolution:

$$\mathbf{v}_i(k) = (1-\alpha)^k \mathbf{v}_i(0) + \alpha \sum_{l=1}^{k} (1-\alpha)^{k-l} \mathbf{p}(l) \quad *$$

$$\mathbf{v}_i(k) = \begin{cases} \mathbf{v}_i(0), & if \quad \alpha = 0 \\ *, & if \quad 0 < \alpha < 1 \\ \mathbf{p}(k), & if \quad \alpha = 1 \end{cases}$$

Variable forgetting factor effect

Advantages of the *c*-means:

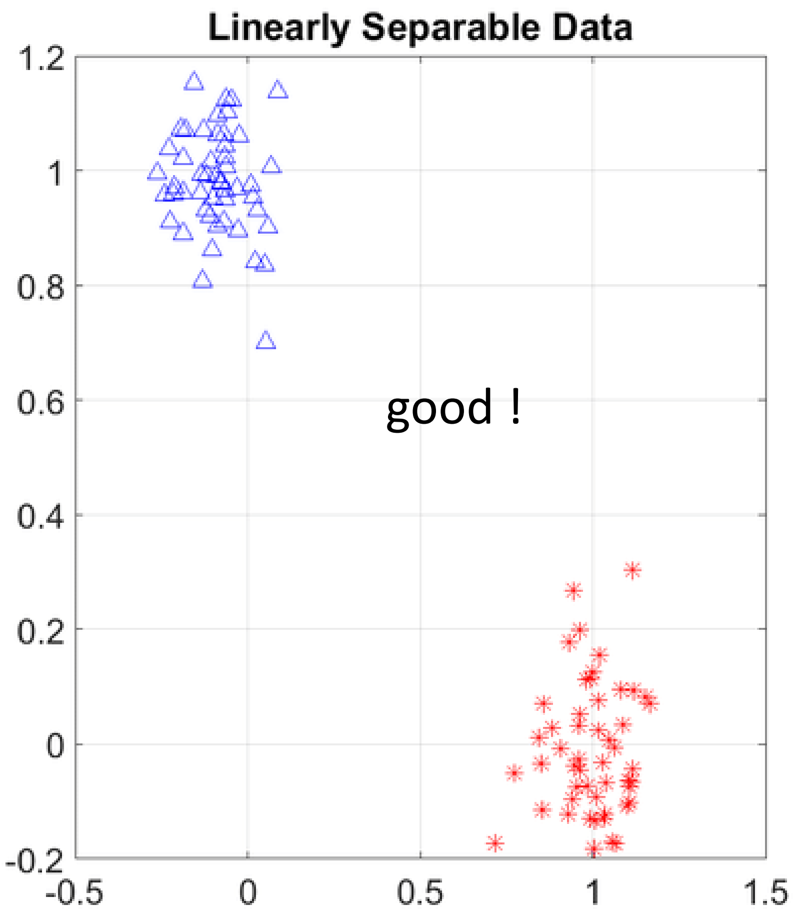
- simple implementation
- allows to fix a-priori the number of clusters

Disadvantages:

- requires some sensitivity to fix *c* ; otherwise, is by trial and error
- the result depends on the initialization of the centers. A different initialization will produce a different sequence of affecting points to a center and recalculating the new center, which may end up with different centers and different clusters, particularly in big datasets.
- it is sensitive to outliers. An outlier will introduce a big distance that may push too much a center to it.

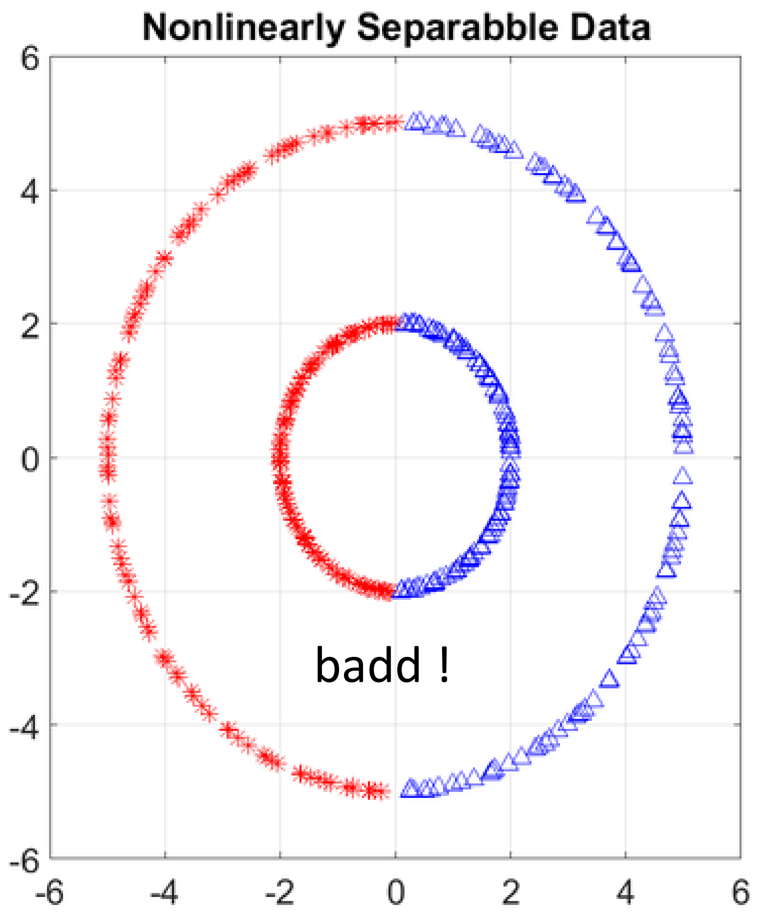
Limitations of standard c-means

If the points are not linearly separable, it may not work as we want:


Linearly Separable Data

good !

(a)


Nonlinearly Separabble Data

badd !

(b)

https://www.researchgate.net/figure/Results-of-k-means-clustering-algorithm-on-a-linearly-separable-input-data-and-b_fig1_323650119 , Mayank Baranwal, 11/Sept/2023

3.4.2 c (or k)-medoids

Kaufman, L. and Rousseeuw, P.J. (1987), Clustering by means of Medoids, in Statistical Data Analysis Based on the –Norm and Related Methods, edited by Y. Dodge, North-Holland, 405–416.

Algorithm very similar to c-means

The centers (medoids) are patterns (the objects more central in the cluster).

Minimizes the sum of the distances of all points (of the cluster) to the medoid.

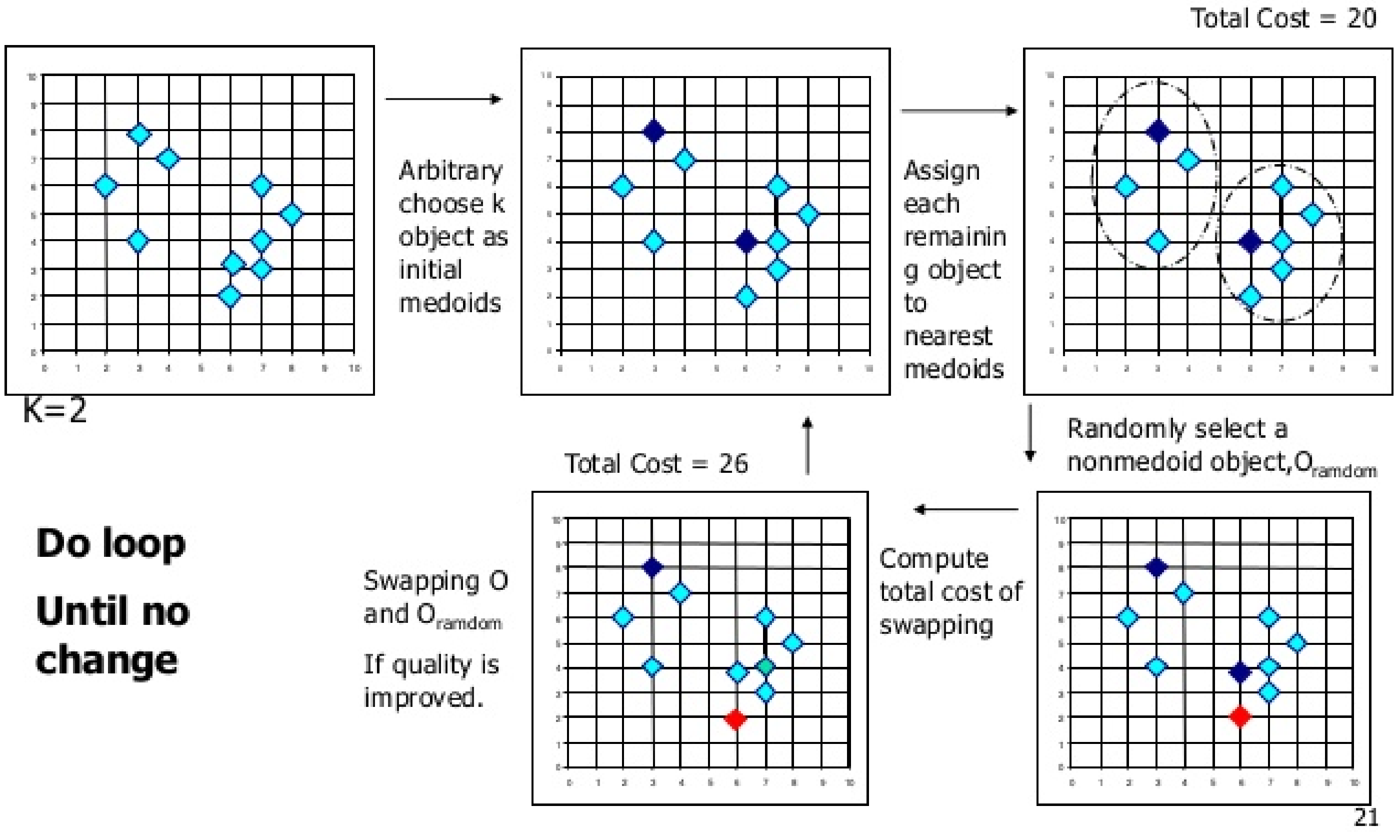
advantage:

- It is more robust that the c-means in the presence of noise and outliers.

disadvantage:

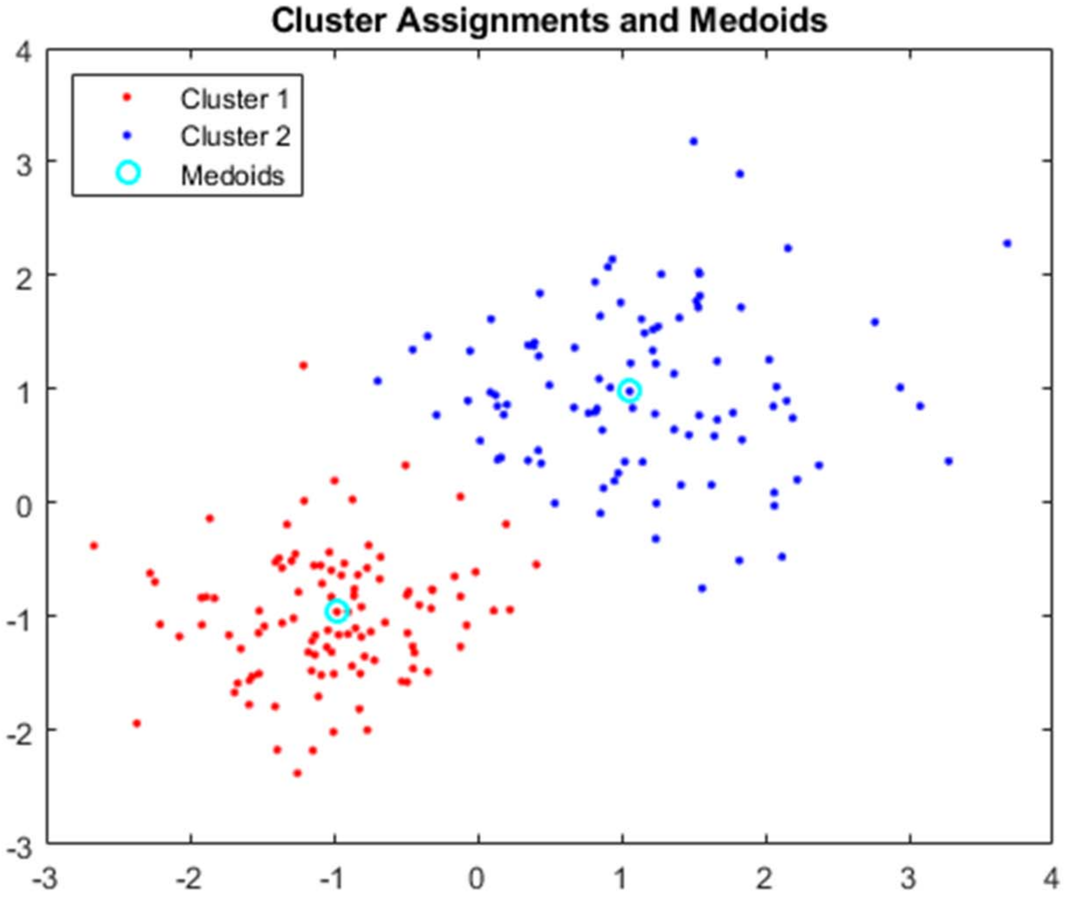
- It does not work well for big datasets. As the dimension increases it is NP hard.

k-medoids


Total Cost = 20
K=2
Arbitrary choose k object as initial medoids
Assign each remaining object to nearest medoids
Randomly select a nonmedoid object,O_{ramdom}
Compute total cost of swapping
Total Cost = 26
Swapping O and O_{ramdom}
If quality is improved.
Do loop
Until no change
21

https://www.slideshare.net/salahecom/10-clusbasic 11/09/2023

1. Define an initial set of medoids.
2. Replace iteratively one of the medoids by one of the non-medoids, if that improves the total distance of the resulting clusters. Hard to compute.



https://www.mathworks.com/help/stats/kmedoids.html 11/09/2023

3.4.3 The fuzzy c-means clustering

$X \triangleq$ universe of points to group (classify)

$\left\{ \underset{\sim}{A}_i, i = 1, 2, \ldots, c \right\}$ a fuzzy c-partition in X

Each element in X belongs to each of the partitions $\underset{\sim}{A}_i$, with some membership value.

In the limit, one point may belong to all partitions (with sum of membership functions equal to 1).

Let $\mu_{ik} = \mu_{\underset{\sim i}{A}}(x_k),\ \mu_{ik} \in [0,1]$

with the constrain

$$\sum_{i=1}^{c} \mu_{ik} = 1, \quad k = 1, 2, \ldots, n$$

One class cannot be empty, and one class cannot have all points with membership 1. So

$$0 < \sum_{k=1}^{n} \mu_{ik} < n,\ \ i = 1, 2, \ldots, c$$

It can happen that $\mu_{ik} \wedge \mu_{jk} \neq 0$, since the k point may belong to both classes i and j.

We have

$\bigcup_{i=1}^{c} \mu_{\underset{\sim}{A}_i}(x_k) = 1,$ the sum of memberships of each x_k, for all k

$0 < \sum_{k=1}^{n} \mu_{\underset{\sim}{A}_i}(x_k) < n,$ for all i (no class has all points with membership 1)

Fuzzy c-means clustering algorithm

n points

c clusters, with centers in $v_1, v_2, \ldots, v_c$

m' , fuzziness degree

Crisp case)

Objective function

$$J_m(\underset{\sim}{U}, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^{m'} . (d_{ik})^2$$

$$J(U, \mathbf{v}) = \sum_{k=1}^{n} \sum_{i=1}^{c} \psi_{ik} (d_{ik})^2$$

$$d_{ik} = d(x_k - v_i) = \sqrt{\sum_{j=1}^{m} (x_{kj} - v_{ij})^2}$$

μ_{ik} : value of membership of point k to the class i

$m' \in [1, \infty)$: ponderation coefficient, measures the fuzziness degree of the classification

The m coordinates of each center v_i are

$$v_{ij} = \frac{\sum_{k=1}^{n} \mu_{ik}^{m'} . x_{kj}}{\sum_{k=1}^{n} \mu_{ik}^{m'}}, \; j = 1, 2, \ldots, m$$

Crisp case (Chap. 6)

$$v_{ij} = \frac{\sum_{k=1}^{n} \psi_{ik} x_{ik}}{\sum_{k=1}^{n} \psi_{ik}}$$

The c-fuzzy optimal partition will minimize J_m

A global minimum cannot be guaranteed, but only the best solution under a pre-specified accuracy.

For two classes $\underset{\sim}{A}_i$ and $\underset{\sim}{A}_j$ $\quad \underset{\sim}{A}_i \cap \underset{\sim}{A}_j \neq \varnothing$

$$\varnothing \subset \underset{\sim}{A}_i \subset X$$

A family of fuzzy partition matrices , M_{fc}, can be defined, to classify n points into c-classes,

$$M_{fc} = \{\underset{\sim}{U} \mid \mu_{ik} \in [0,1]; \sum_{i=1}^{c} \mu_{ik} = 1; 0 < \sum_{k=1}^{n} \mu_{ik} < n\}$$

$$i = 1, 2, \ldots, c \qquad k = 1, 2, \ldots, n$$

Any $\underset{\sim}{U} \in M_{fc}$ is a fuzzy partition.

M_{fc} has infinite cardinality.

Iterative procedure (Ross, 352)

1th Fix c $(2 \leq c < n)$

Chose a value for m'

Initialize the partition matrix $\underset{\sim}{U}^{(0)}$ (ex. randomly)

For r=1,2,..., do:

2nd Compute the centers $\{v_i^{(r)}, i=1,2,...,c\}$

3th Update the partition matrix at iteration r, $\underset{\sim}{U}^{(r)}$

$$\mu_{ik}^{(r+1)} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right)^{\frac{2}{m'-1}} \right]^{-1}$$ for $I_k \neq \varnothing$ (for the classes whose indexes belong to I_k)

or

$\mu_{ik}^{(r+1)} = 0,$ for all the classes i in which $i \in \tilde{I}_k$

$I_k = \{ i \mid 2 \le i < c \ : \ d_{ik}^{(r)} = 0 \}$ (centers whose distance to point k is null point k will be a center, so membership 1)

$\tilde{I}_k = \{2, \ldots, c\} - I_k$ (centers whose distance to the point k is non null)

$$\sum_{i \in I_k} \mu_{ik}^{(r+1)} = 1$$

4th If $\left\| U^{(r+1)} - U^{(r)} \right\| < \varepsilon_L$ stop.

If not, $r = r+1$

go to 2nd

J_m: criterion of minimum of squared distances.

The squared distances are weighted by the membership values $(\mu_{ik})^{m'}$.

J_m: minimizes the squared distances of the points to their centers

maximizes the distances between the centers of the clusters.

Which is the good value for *m'* ?

$$m' = 1, \quad \frac{2}{m'-1} = \frac{2}{0} = \infty$$

and then

$$\mu_{ik}^{(r+1)} = \begin{cases} 1, \text{ if for all } j, d_{ik}^{(r)} < d_{jk}^{(r)}, i \neq j, \left(\dfrac{d_{ik}}{d_{jk}}\right)^{\infty} = 0, \left(\dfrac{d_{ik}}{d_{ik}}\right)^{\infty} = 1 \\ 0, \text{ if for some } j, d_{ik}^{(r)} > d_{jk}^{(r)}, j \neq i, \left(\dfrac{d_{ik}}{d_{jk}}\right)^{\infty} = \infty, \infty^{-1} = 0 \end{cases}$$

(crisp case)

$$m' = \infty, (\mu_{ik})^{m'} = 0 \Rightarrow J_m(U, v) = 0$$

(completely fuzzy)

The greater m’, the stronger is the fuzziness of the membership to the clusters; m´ controls this fuzzy character of membership to the clusters.

m’ increases, J_m decreases (keeping constant all the other parameters), slower is the convergence.

There is no theoretical optimum for m’; generally it is chosen between 1.25 and 2 (Ross).In Matlab default is 2.

Measures for the fuzzy classification

-which is the uncertainty (fuzziness) degree of a classification ?

-“How fuzzy is a fuzzy c-partition?” (Ross).

- which is the level of superposition of the defined classes ?

Let x_k be a classified element of the Universe

$\mu_i(x_k)$- value of membership of x_k to the class i

$\mu_j(x_k)$- value of membership of x_k to the class j

$\mu_i\,\mu_j$, algebraic product, depending directly from the relative superposition between non-empty clusters. This is a good measure for the uncertainty of the classification.

Coefficient of the fuzzy partition

$$F_C(\underset{\sim}{U}) = \frac{tr(\underset{\sim}{U}\,\underset{\sim}{U}^T)}{n}, \quad \underset{\sim}{U} = [\mu_{ik}]:$$ matriz of fuzzy partition

Interprets the results of the fuzzy partition.

Properties:

$F_C(\underset{\sim}{U}) = 1$ if the partitions are crisp

$F_C(\underset{\sim}{U}) = \frac{1}{c}$ if $\mu_i = \frac{1}{c}$, $i=1, 2, ..., c$ (total ambiguity)

$\frac{1}{c} \leq F_c(\underset{\sim}{U}) \leq 1$ in any case

The elements of the diagonal of $\underset{\sim}{U}\,\underset{\sim}{U}^T$ are proportional to the quantities of non-shared memberships in the fuzzy clusters.

The elements out of the diagonal of $\underset{\sim}{U}\,\underset{\sim}{U}^T$ represent the quantity of shared partition between pairs of fuzzy clusters. If they are null, the partition is crisp.

As the fuzzy partition coefficient approaches 1, the fuzzy uncertainty is minimized in the overlapping of the clusters. The greater $F_C(\underset{\sim}{U})$, the better succeeded is the partition of the dataset into clusters.

Remarks:

1. If x_k belongs with memberships $\mu_i(k)$ and $\mu_j(k)$ to the classes i and j , then $min(\mu_i(k), \mu_j(k))$ gives the quantity of membership claimed both by i and by j, and so not shared. It is a good measure of self - superposition.

2. The elements of the diagonal of $\underset{\sim}{U}\,\underset{\sim}{U}^T$ are the sum of the square of the elements of each row of $\underset{\sim}{U}$

Defuzzification of a fuzzy partition

Method of maximum membership

- the highest element of each columns of U passes to 1, and all the others to zero.

$$\mu_{ik} = \underset{i \in c}{máx}\left\{u_{ik}\right\} \Rightarrow \mu_{ik} = 1 \wedge \mu_{jk} = 0,\ j \neq i$$

$$i, j = 1, 2, \ldots, c \qquad k = 1, 2, \ldots n$$

Classification by the nearest center

- each point is affected to the cluster whose center is closest to it,

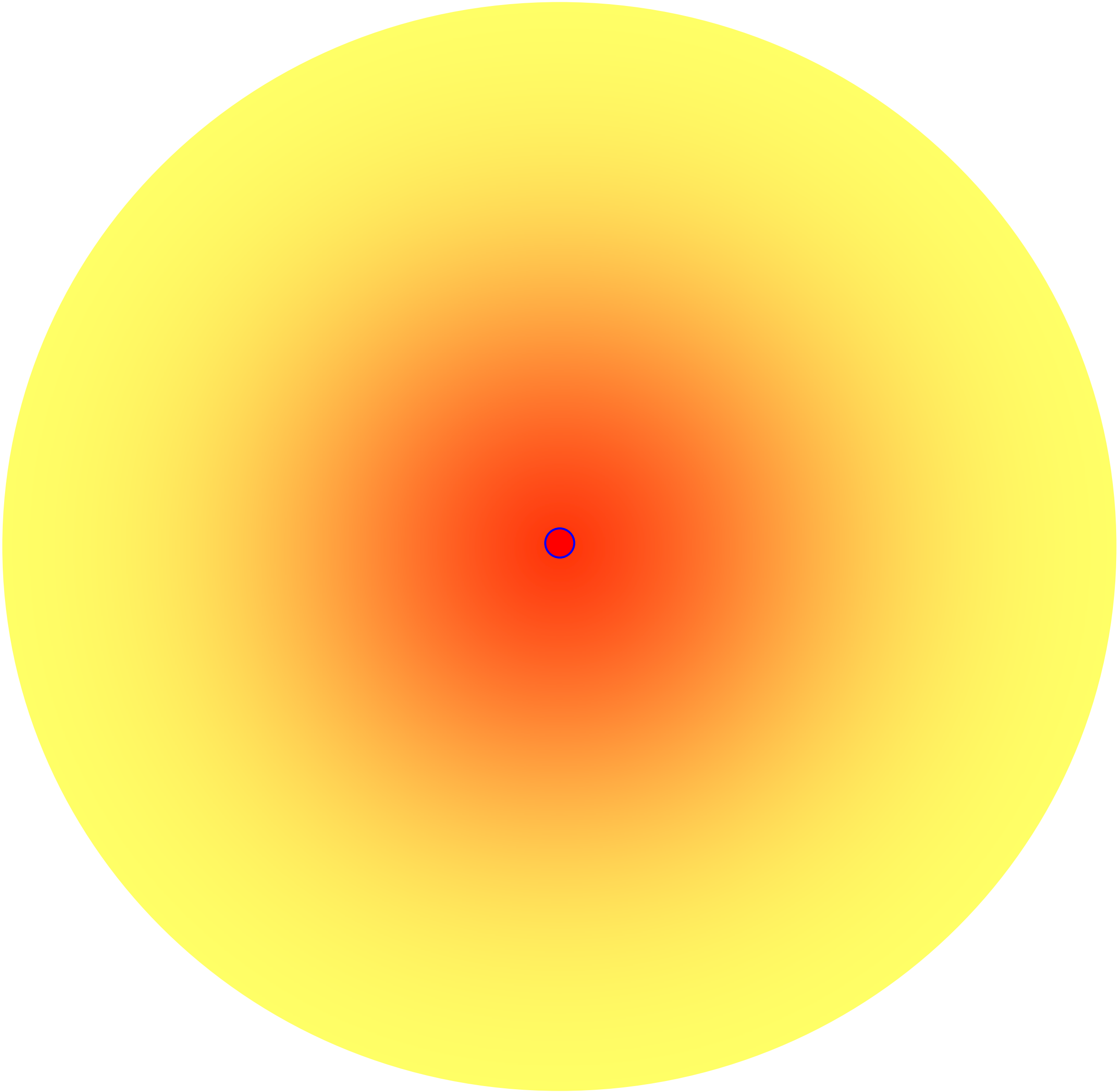
$$d_{ik} = \underset{i \in c}{mín}\left\{d_{ik}\right\} \Rightarrow \mu_{ik} = 1 \wedge \mu_{ij} = 0,\ j \neq i$$

$$d_{jk} = \left\| x_k - v_j \right\|$$

In Fuzzy Logic Toolbox: *fcm*

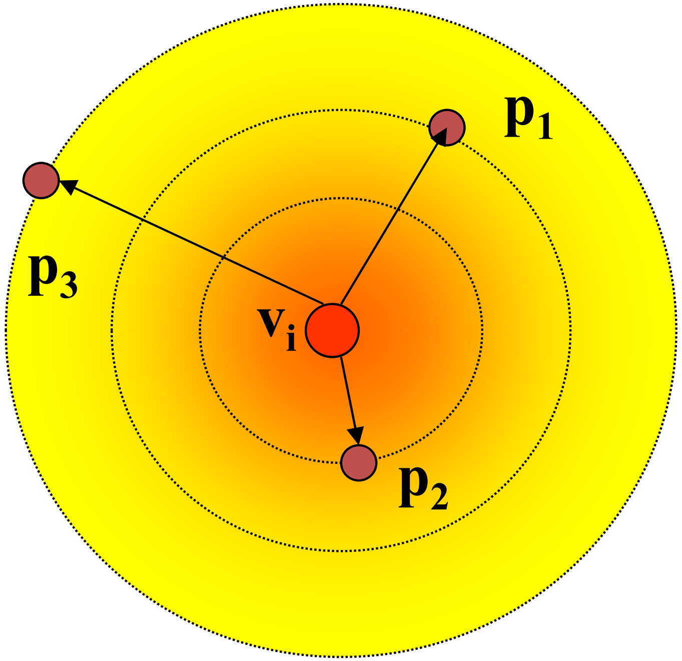
GUI for clustering: *findcluster* in command line.

3. 4.4. Subtractive clustering



Methods based on a function of potential

A set of points $\mathbf{v}_i$ are defined as possible centers (by a grid or by some heuristic). Each center is then considered as radiating energy and one measures the energy received by each point in its neighborhood. The energy received in a point (ex. $\mathbf{p}_1$) is a measure of the potential P of the center $\mathbf{v}_i$ in that point $\mathbf{p}_1$; this energy decreases exponentially with the distance:



$$P(\mathbf{v}_i,\mathbf{p}_j)=e^{-\alpha\left\|\mathbf{v}_i-\mathbf{p}_j\right\|^2}$$

The constant α fixes the speed of decay of the potential, i.e., fixes the area of influence of each center. The potential of $\mathbf{v}_i$ is the sum of its potentials in all the points.

The subtractive clustering method

S. L. Chiu, "Fuzzy model identification based on cluster estimation". Journal of Intelligent and Fuzzy Systems, Vol. 2, No. 3, 1994.

Consider Q points $\mathbf{P} = \{\mathbf{p}_1 \quad \mathbf{p}_2 \quad \dots \quad \mathbf{p}_\mathbf{Q}\}$ in the R-dimensional space

Normalize the data to [0,1] or [-1, 1].

1st - Each sample $\mathbf{p}_\mathbf{i}$ defines a possible center

2nd - The potential P_i associated to $\mathbf{p}_\mathbf{i}$ is the sum of the energies (irradiated by $\mathbf{p}_\mathbf{i}$) received in each point, from 1 to Q:

$$P_i^{(0)}(\mathbf{p}_i, \mathbf{P}) = \sum_{j=1}^{Q} e^{-\alpha \left\| \mathbf{p}_i - \mathbf{p}_j \right\|^2}, i = 1, \dots, Q$$

$\alpha = \dfrac{4}{r_a^2}, r_a > 0,$ radius of the neighborhood of each point

3rd – Chose the first center $\mathbf{v}_1$ as the point P_1^* with the highest potential

4th – Reduction of the potential of all the points

$$P_i^{(1)} = P_i^{(0)} - P_1^* e^{-\beta\left\|\mathbf{p}_i - \mathbf{v}_1\right\|^2} \qquad \beta = 4/r_b^2$$

r_b : radius of the neighborhood with a significant reduction of potential. If is next to $\mathbf{v}_1$ its potential is strongly reduced, preventing the concentration of the centers ($\mathbf{p}_\mathbf{i}$ is “emptied” and will never be “filled” again; the same for its neighbors)

5th – Chose the second center $\mathbf{v}_2$ as the point P_2^* that remains with the highest potential.

6th – Reduce again the potential of all points

$$P_i^{(2)} = P_i^{(1)} - P_2^* e^{-\beta \left\| \mathbf{p}_i - \mathbf{v}_2 \right\|^2}$$

7th – Chose the 3rd center $\mathbf{v}_3$ as the point P_3^* remaining with the highest potential.

8th – Reduce again the potential of all the points

$$P_i^{(3)} = P_i^{(2)} - P_3^* e^{-\beta \left\| \mathbf{p}_i - \mathbf{v}_3 \right\|^2}$$

And so on , until the potential of all points is residual, under a fixed threshold.

Advantages:

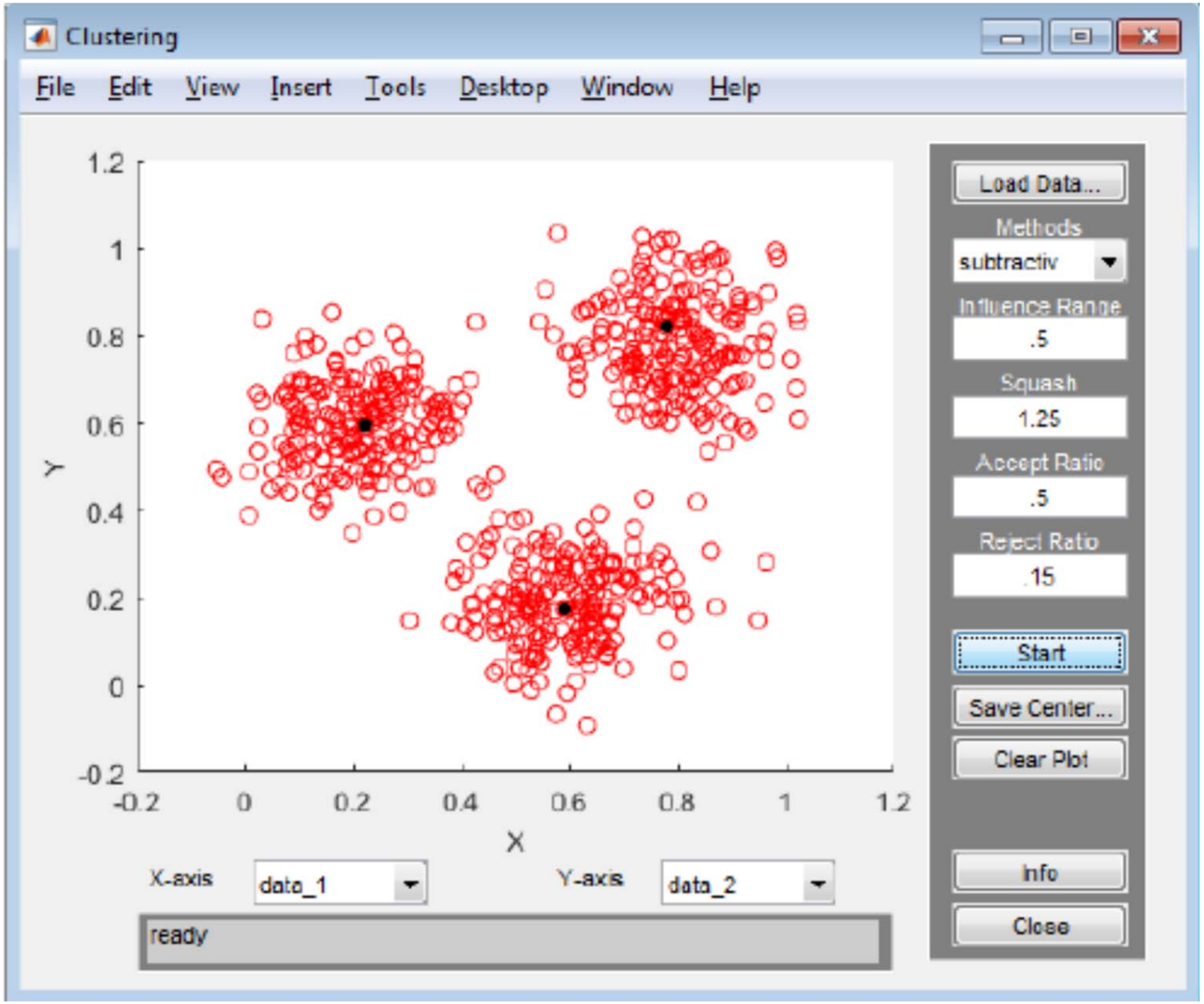
It surpasses the *curse of dimensionality* problem: the obtained number of clusters reflects the variation of the density of the points in the data space.

The number of obtained clusters depends on α and β, the two degrees of freedom of the method.

It is also very used in fuzzy and neuro-fuzzy systems.

Fuzzy c-means and subtractive clustering GUI

> findcluster, Clustering GUI , Matlab Fuzzy Logic Toolbox, 2022a, pp 4-31



Methods: subtractiv

Influence Range: .5

Squash: 1.25

Accept Ratio: .5

Reject Ratio: .15

X-axis: data_1

Y-axis: data_2

ready

3.4.5. DBSCAN Density-Based Spatial Clustering of Applications with Noise

- it is a density-oriented approach, proposed in 1996 by [Ester, Kriegel, Sander and Xu]. It is probably one of the most used clustering methods.
- if in a region there exists a high density of points, then there exists a cluster.

The DBSCAN algorithm uses two parameters:

minPts: The minimum number of points huddled together for a region to be considered dense. It is a threshold.

eps (ε): A distance measure that will be used to find the number of points in the neighborhood **ε** of any point and compare it with MinPts.

It is based on two concepts: Density Reachability and Connectivity

- **Density Reachability** : establishes a point *p* to be reachable from another point *q* if it lies within a particular distance (**ε**) from it **and** minPts of *q* is greater than the threshold. It is not (always) a symmetric relation.

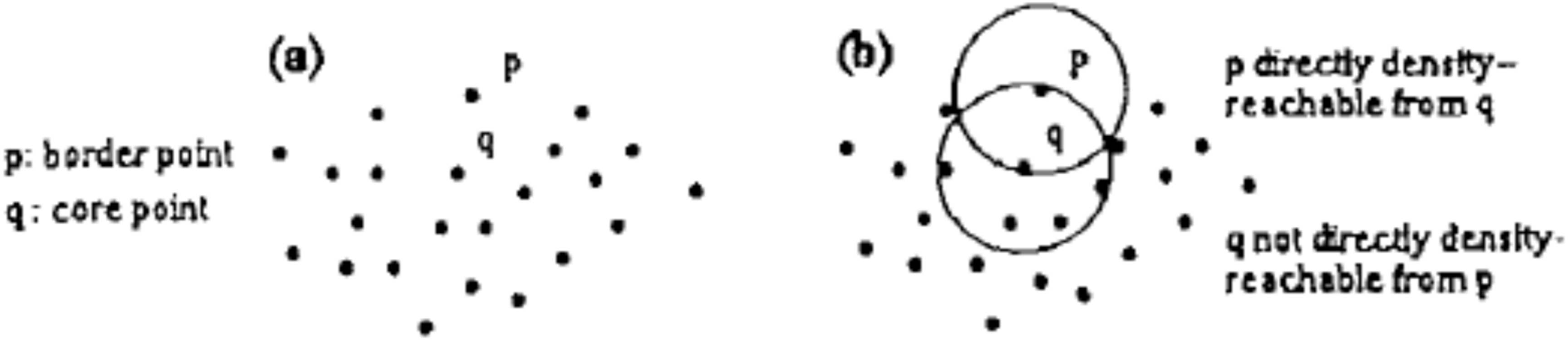

(a)
p
q
p: border point
q : core point
(b)
p
q
p directly density-reachable from q
q not directly density-reachable from p

figure 2: core points and border points
(Ester, Kriegel, Sander and Xu)

Note that in this figure, from the original work, counting, minPts (*p*)=2, minPts(*q*)=6. Let the threshold be fixed at 5.
minPts of *p* does not respect the threshold, so *q* is not directly density – reachable from *p*, while, by contrary, *p* is directly density-reachable from *q* because minPts of *q* respects the threshold. This means that *q* is in a dense region, while *p* is not in a dense region. The outliers are never in a dense regions, so they will not be included in any cluster with DBSCAN method.

- **Connectivity**, involves a transitivity based chaining-approach to determine whether points are located in a particular cluster. For example, *p* and *q* points are connected if

$$p \rightarrow r \rightarrow s \rightarrow t \rightarrow q$$

where $a \rightarrow b$ means *b* is in the **ε** neighborhood of *a* .

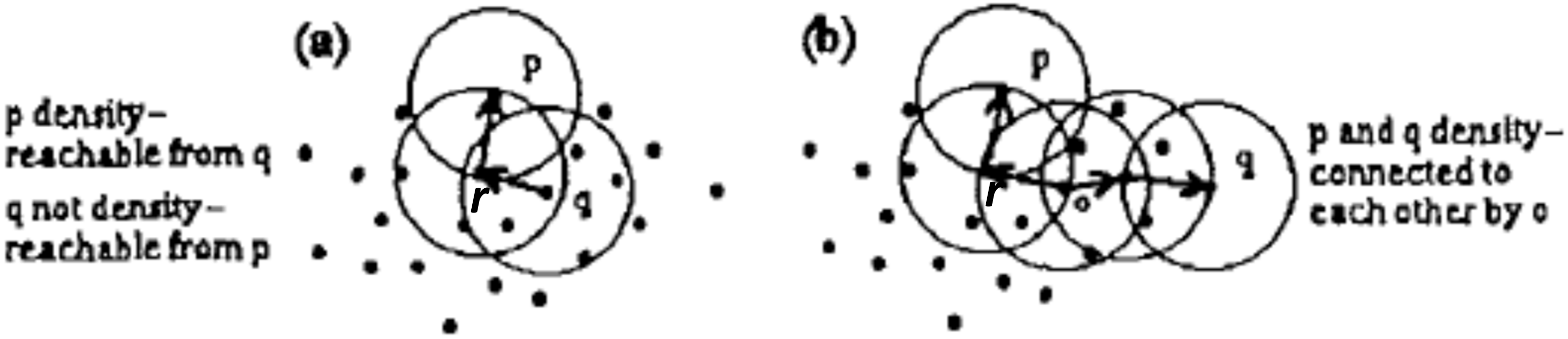

(a)
p density-reachable from q
q not density-reachable from p
p
r
q
(b)
p
r
o
q
p and q density-connected to each other by o

figure 3: density-reachability and density-connectivity

(Ester, Kriegel, Sander and Xu)

In (a), from *q* to *p*, we pass by *r.* So *p* is density reachable from *q,* because minPts of both *q* and *r* are greater than the threshold. But *q* is not density-reachable from *p* because minPts(*p*) is lower than the threshold.

In (b), from *p* to *q* we pass by successive points such that one is in the neighborhood of the next until we reach *q*.

See also in https://pt.slideshare.net/ssakpi/density-based-clustering a nice presentation.

The algorithm works as follows: (https://iq.opengenus.org/dbscan-clustering-algorithm/ with an animated demonstration, 11/Sept/2023)

1. Pick an arbitrary data point **p** as your first point.
2. Mark **p** as visited.
3. Find all points present in its neighborhood (up to **ε** distance from the point), and call it the set np**.**

4**.** If np >= minPts, then
   a. Consider **p** as the first point of a **new cluster** and mark it as visited.
   b. Find all points within **ε** distance (members of np) as other points in this cluster and mark them as visited.
   c. Repeat step b for all points in np.
5. Else if np < minPts label **p** as noise and mark it as visited.
6. Repeat steps 1-5 till the entire dataset has been visited (labeled), i.e. the clustering is complete.

After the algorithm is executed, we should ideally have a dataset separated into a number of clusters, and eventually some points labeled as noise which do not belong to any cluster. It is this particularity that give the name to the method.
The method is particularly suitable for nonlinearly separated clusters.

Advantages and disadvantages of DBSCAN

Advantages:

- it does not require a pre-set number of clusters
- it also identifies outliers as noise, unlike others
- it is able to find arbitrarily sized and arbitrarily shaped clusters quite well.

Disadvantages:

- it doesn’t perform as well as others when the clusters are of varying density. This is because the setting of the distance threshold **ε** and minPts for identifying the neighborhood points will vary from cluster to cluster when the density varies. This drawback also occurs with very high-dimensional data since again the distance threshold **ε** becomes challenging to estimate in each dimension (it may be different among the dimensions).

in MATLAB (2020b):

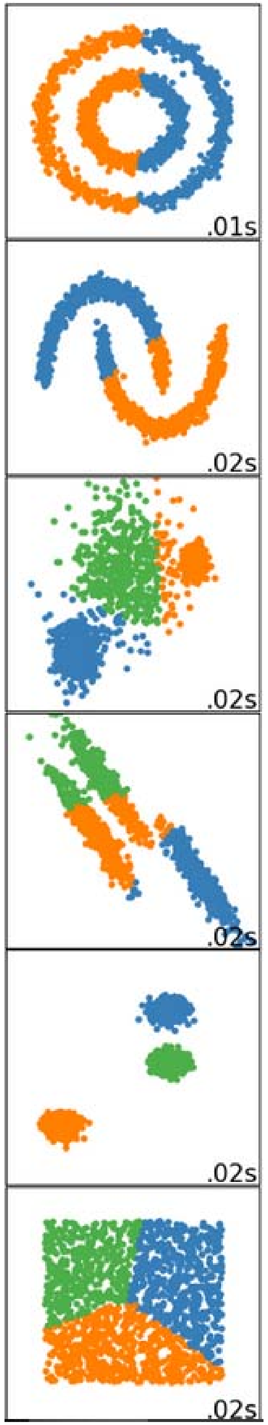
```
idx = dbscan(X,epsilon,minpts)
idx = dbscan(X,epsilon,minpts,Name,Value)
idx = dbscan(D,epsilon,minpts,'Distance','precomputed’)
```

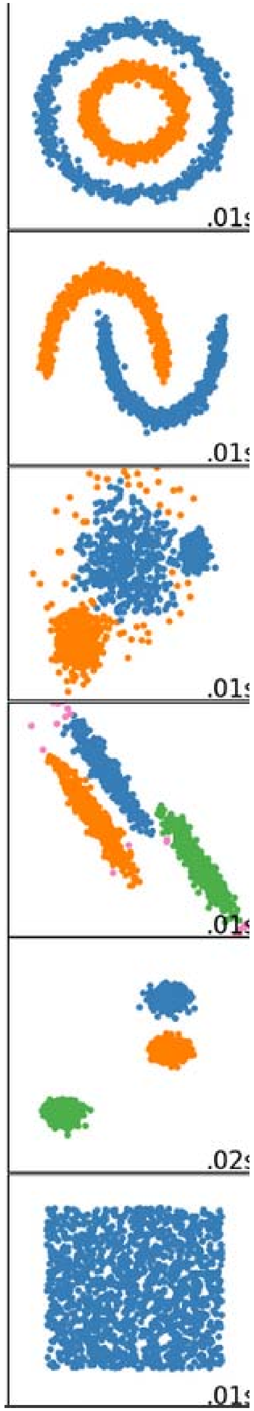
see a visualization at

https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 11/09/2023

Comparison of standard c-means with DBSCAN for several types of datasets

c-means





DBSCAN

Comparison of the more used five methods of this chapter

Method	Type	Need a-priori number of clusters	Computational complexity	Parameters to control number of clusters	Shape of clusters
AGNES	agglomerative hierarch	no	O(n^3)	height of the dendrogram	any
c-means	partitional distance	yes	O(n)	trial and error	hyper spherical
c-medoides	partitional distance	yes	O(n^2)	trial and error	hyper spherical
fuzzy c-means	partitional, distance	yes	O(n)	trial and error	hyper spherical
subtractive	partitional, density	no	O(n)	2 parameters	any
DBSCAN	particional, density	no	O(n^2)	2 parameters	any

Other clustering methods

There are several other methods. One interesting, that surpasses some drawbacks of k-means, is the **Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)**, that can be seen at https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 11/09/2023, and can also be implemented in Matlab Statistics and Machine Learning Toolbox 2022b, pp. 16-54. https://www.mathworks.com/help/stats/clustering-using-gaussian-mixture-models.html .

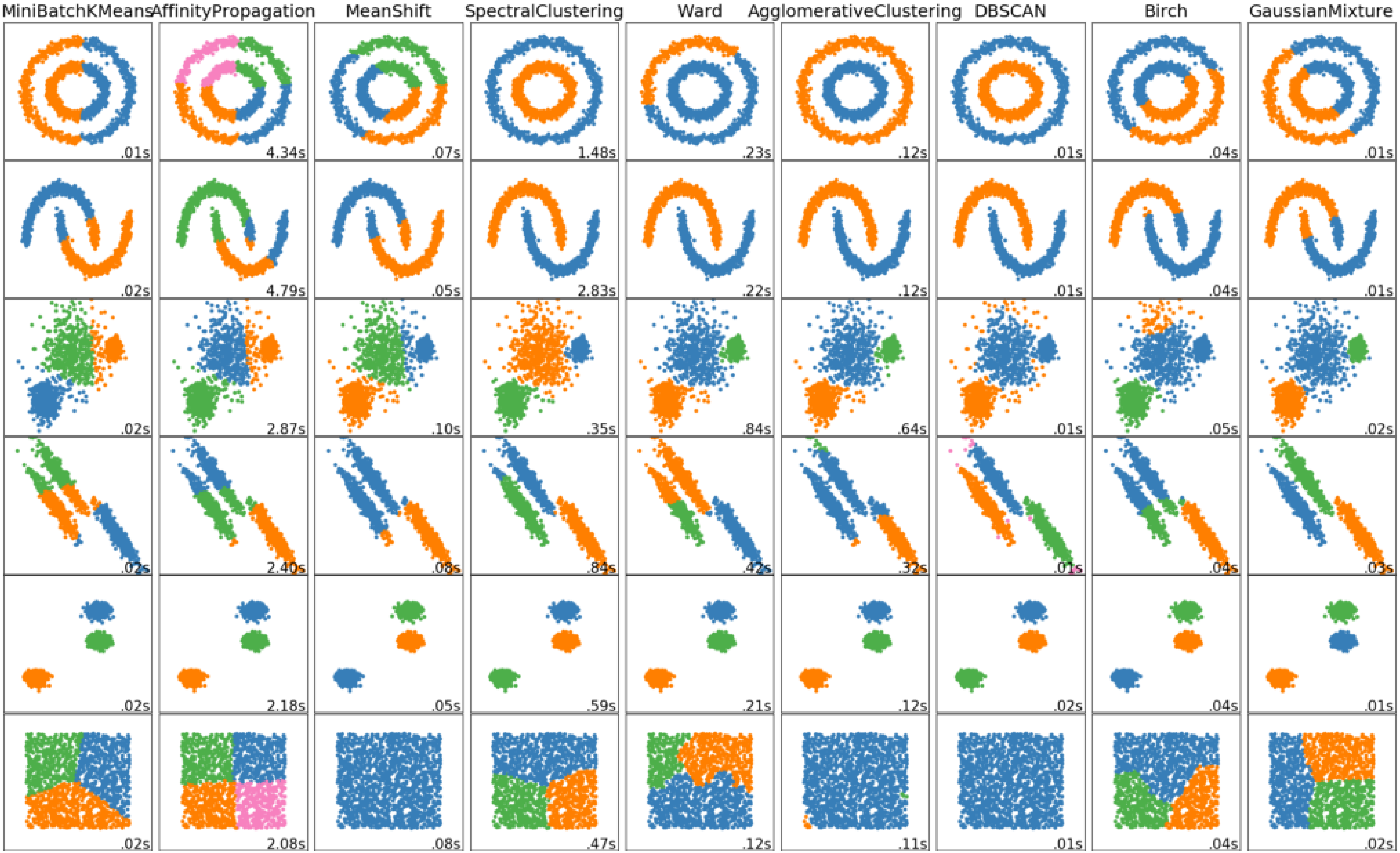
Mean-Shift Clustering, is a sliding-window-based algorithm that attempts to find dense areas of data points. In contrast to K-means clustering there is no need to select the number of clusters as mean-shift automatically discovers it. See also in https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 , 11/09/2023.

Spectral clustering is a graph-based . Very often outperforms traditional algorithms such as the k-means algorithm. See https://towardsdatascience.com/spectral-clustering-for-beginners-d08b7d25b4d8 , 11/09/2023.

Other method that is frequently used in classification is the **k-nearest neighbors**, see https://www.mathworks.com/help/stats/nearest-neighbors-1.html?s_tid=CRUX_lftnav , 11/09/2023)

However, the six methods included in this chapter give a broad idea of the clustering problem, as an unsupervised machine learning technique, its particularities and difficulties. The figure in the next slide is very illustrative of the challenges of a good clustering.

Comparison of (9) clustering algorithms for different types of data



from https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68
11/09/20223

3.4.6. Clustering in Matlab

Statistics and Machine Learning Toolbox 2020b, Chapter 16

Hierarchical Clustering: T = cluster(Z,'Cutoff',C), pp 16-6

K-means: IDX = kmeans(X, K) pp 16-33
partitions the points in the N-by-P data matrix X into K clusters,
by default uses the squared euclidian distance
distances: *pdist*

K-medoids IDX = kmedoids(X, K) partitions the points in the N-by-P data matrix X into K clusters

Spectral clustering: spectralcluster function, 16-26

DBSCAN: IDX = dbscan(X, EPSILON, MINPTS) partitions the points in the N-by-P data matrix X into clusters based on parameters EPSILON and MINPTS.
pp. 16-19.

Fuzzy Logic Toolbox

Fuzzy c-means: [centers,U] = fcm(data,Nc,options) The membership function matrix U contains the grade of membership of each DATA point in each cluster pp 4-12

Subtractive : centers = subclust(data,clusterInfluenceRange) clusters input data using subtractive clustering with the specified cluster influence range, and returns the computed cluster centers. pp 4-2.

7. Bibliography

Handbook of Cluster Analysis, Christian Hennig, Marina Meila, Fionn Murtagh, Roberto Rocci,, 2015, Chapman and Hall/CRC.

Comparing the performance of biomedical clustering methods, Christian Wiwie, Jan Baumbach, Richard Röttger, Nature Methods , 12, 1021-1031 (2015)   DOI: 10.1038/nnmeth.3623.

Data Clustering: A Review, Jain, A.K., M.N. Murty, and P.J. Flynn, ACM Computing Surveys, Vol. 31, No. 3, September 1999.

Clustering Algorithms in Biomedical Research: A Review, Rui Xu,Donald C. Wunsch II,, IEEE Reviews in Biomedical Engineering (Volume: 3) , p 120-154, Oct 2010, https://ieeexplore.ieee.org/document/5594620/ , 11 Sept 2023

Survey of Clustering Data Mining Techniques, Pavel Berkhin ,, Accrue Software https://www.cc.gatech.edu/~isbell/reading/papers/berkhin02survey.pdf 11 Sept 2023

Matlab Statistics and Machine Learning Toolbox User's Guide, 2020b , Chapter 16, Mathworks Inc.
Matlab Fuzzy Logic Toolbox, 2020b, Chapter 4, Mathworks Inc.

Fuzzy Logic With Engineering Applications, Timothy Ross, 4th Ed.,Wiley, 2016.

7. Bibliography (cont.)

A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu, KDD-96 Proceedings. Copyright © 1996, AAAI https://aaai.org/papers/037-a-density-based-algorithm-for-discovering-clusters-in-large-spatial-databases-with-noise/ 11 Sept 2023

https://www.slideshare.net/salahecom/10-clusbasic Jiawei Han, Micheline Kamber, and Jian Pei, University of Illinois Urbana-Campaingn & Simon Fraser University
slides after chapt 10 of the book
Data Mining: Concepts and Techniques (3rd Ed.) , Jiawei Han, Micheline Kamber, and Jian Pei, University of Illinois Urbana-Campaingn & Simon Fraser University, Morgan Kaufmann**,** 2013.
A Survey of Clustering Technique, Rai P. and Singh S, International Journal of Computer Applications, Vol 7, nº 12, October 2010, doi : 10.5120/1326-1808. s

Fuzzy model identification based on cluster estimation, S. L. Chiu, Journal of Intelligent and Fuzzy Systems, Vol. 2, No. 3, 1994. (subtractive clustering) DOI: 10.3233/IFS-1994-2306

A Tutorial on Clustering Algorithms, with interactive demonstrations, Matteo Matteucci , https://matteucci.faculty.polimi.it/Clustering/tutorial_html/index.html 11 Sept 2023

https://sites.google.com/site/dataclusteringalgorithms/hierarchical-clustering-algorithm
11 Sept 2023