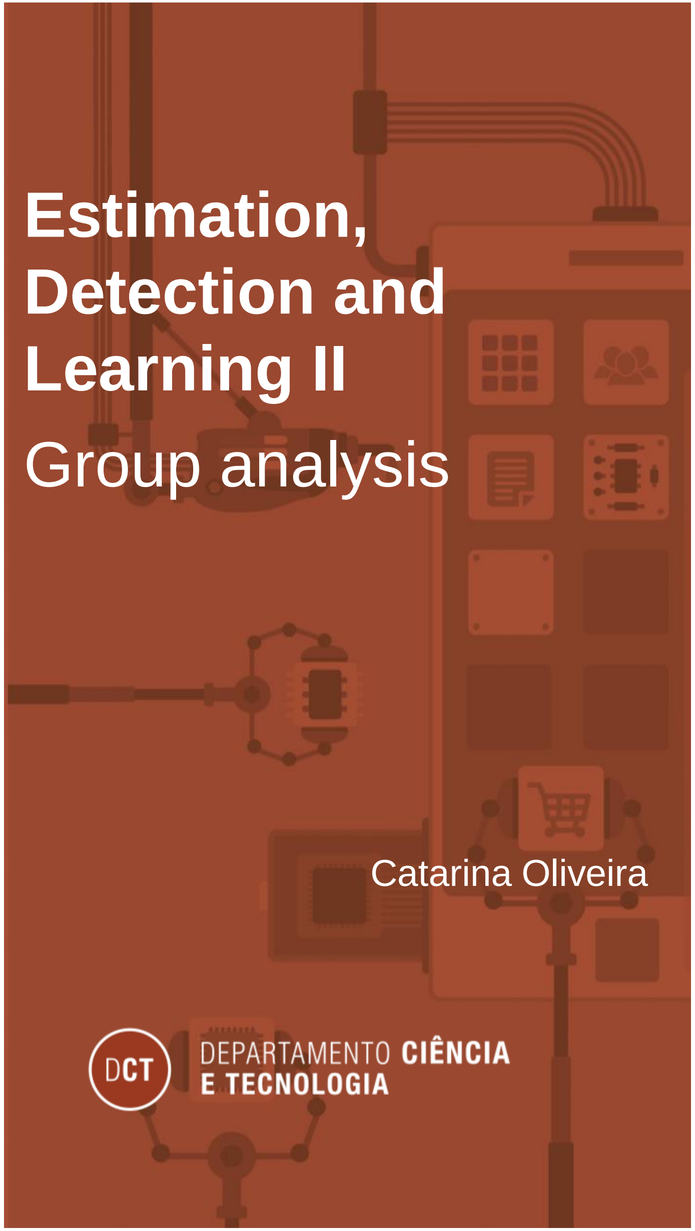


Estimation, Detection and Learning II

Group analysis

Catarina Oliveira

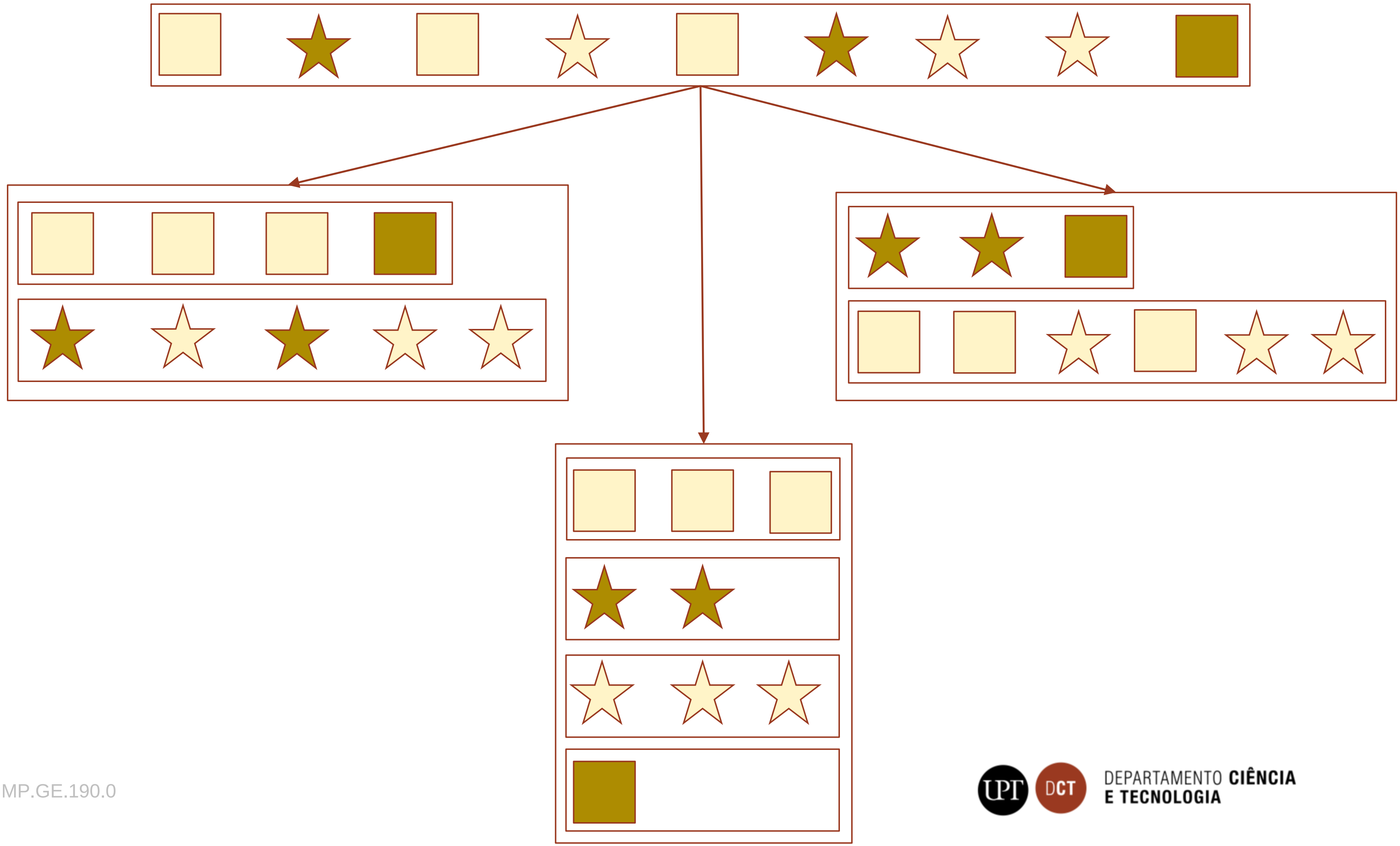
DCT DEPARTAMENTO CIÊNCIA E TECNOLOGIA

UPT UNIVERSIDADE PORTUCALENSE

CONTENT

1. partition methods
    1. k- means
    2. k- medoids
2. density-based clustering
3. hierarchical methods
    1. grid methods

Problem



clustering

Organize the data into groups such that:

- Intra- group similarity is high.
- The intergroup similarity is reduced

Clustering ≠ sorting

- **Classification** : discover the label (from a set of possible values) of each instance
- **Clustering** : discover the set of possible values for the labels and assign one to each instance

Results:

- Exclusive clusters: each item can only belong to one cluster
- Overlapping clusters: each item can belong to more than one cluster
- Probabilistic clusters: each item has a certain probability of belonging to a cluster

clustering

- Biology and science:
  - Grouping of animals / plants

- Market
  - Similar customer groups for targeted advertising
  - fraud identification

- web
  - document classification
  - Discovery of groups of similar access patterns in logs
  - recommendation systems

Types of Clustering

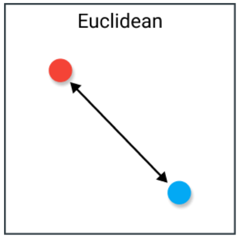
- Partition / Hierarchical
  - **Partition** : Constructs several partitions of the objects and evaluates each one using a criterion
    - Ex: k- means , k- medos
  - **Hierarchical** : Creates a hierarchical decomposition of objects based on a criterion

- Density-based / Model-based
  - **density-based**
    - Uses the notion of density (number of objects in a cluster)
    - Allows non-spherical clusters (unlike methods that use distance measures)
    - Robust to outliers
    - Ex: DBSCAN
  - **model-based**
    - Defines a template for each cluster
    - Looks for the best data fit for each model
    - Optimal number of clusters defined using statistical methods

similarity calculation

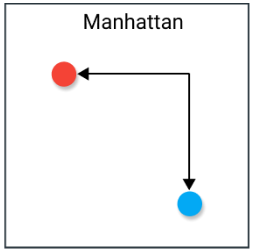
euclidean

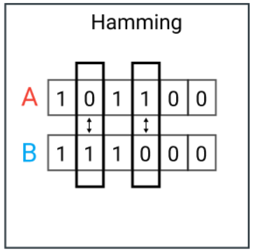
$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$



Manhattan

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$$





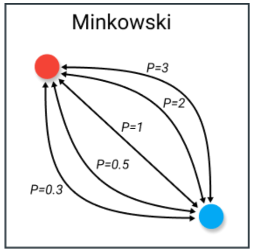
A	1	0	1	1	0	0
		↕		↕		
B	1	1	1	0	0	0

Hamming (distance between strings)

Number of different characters between strings

Minkowski

$$d(x,y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}}$$



Distance measurements: https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa

K- means

means

Input:

- O: Set of *n* objects
- K: number of clusters to create

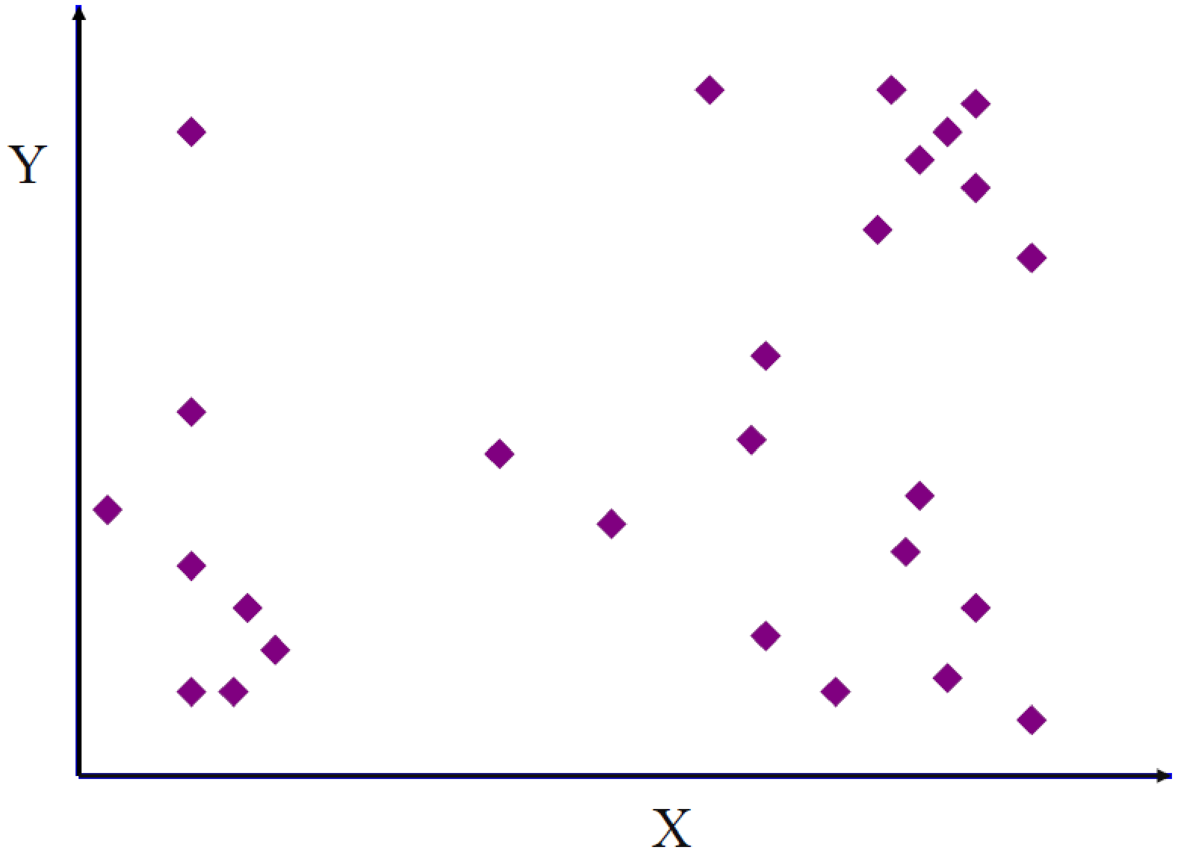
Steps:

1. Randomly choose *K* centroids
2. Repeat until no changes are made
    1. Assign each object to the cluster it is most similar to
    2. Calculate the new cluster centroid

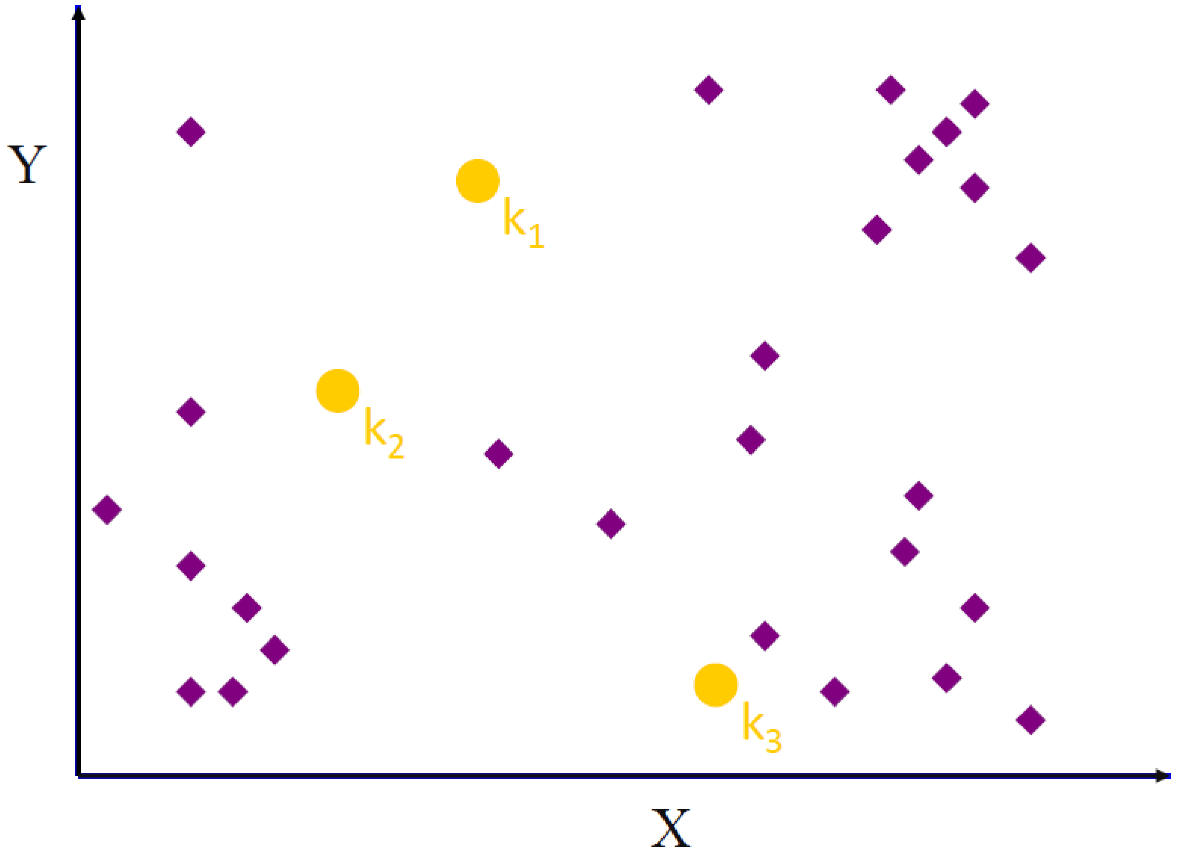
Output:

- Set of *K* clusters

K- means : example

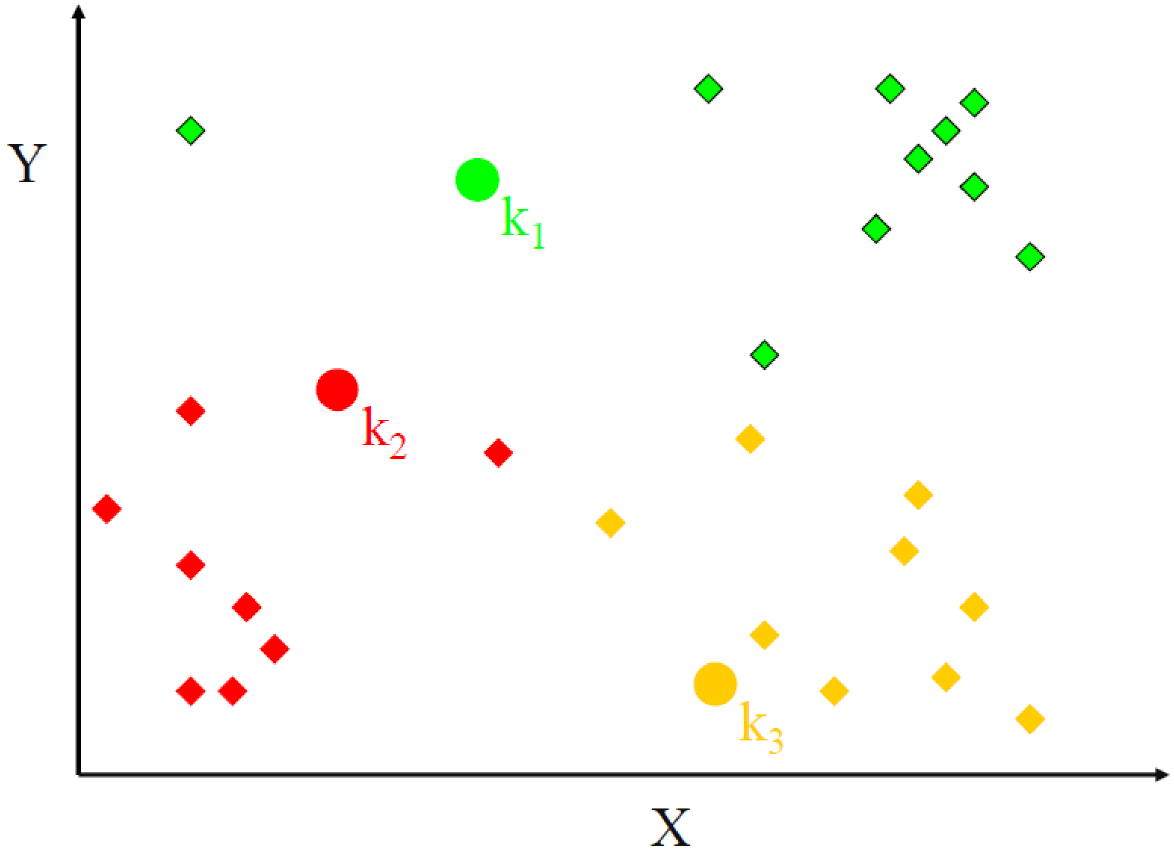


Example: step 1



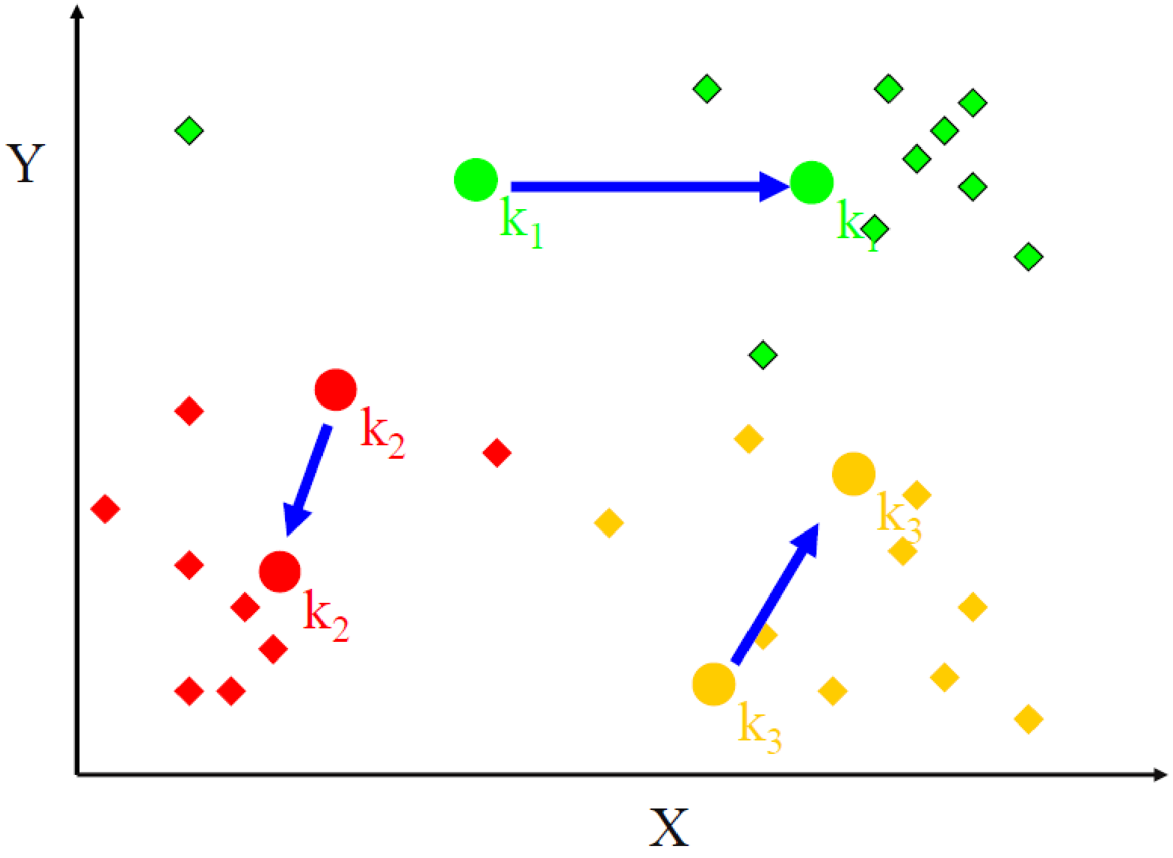
Randomly choose k centroids

Example: step 2.1 (iteration 1)



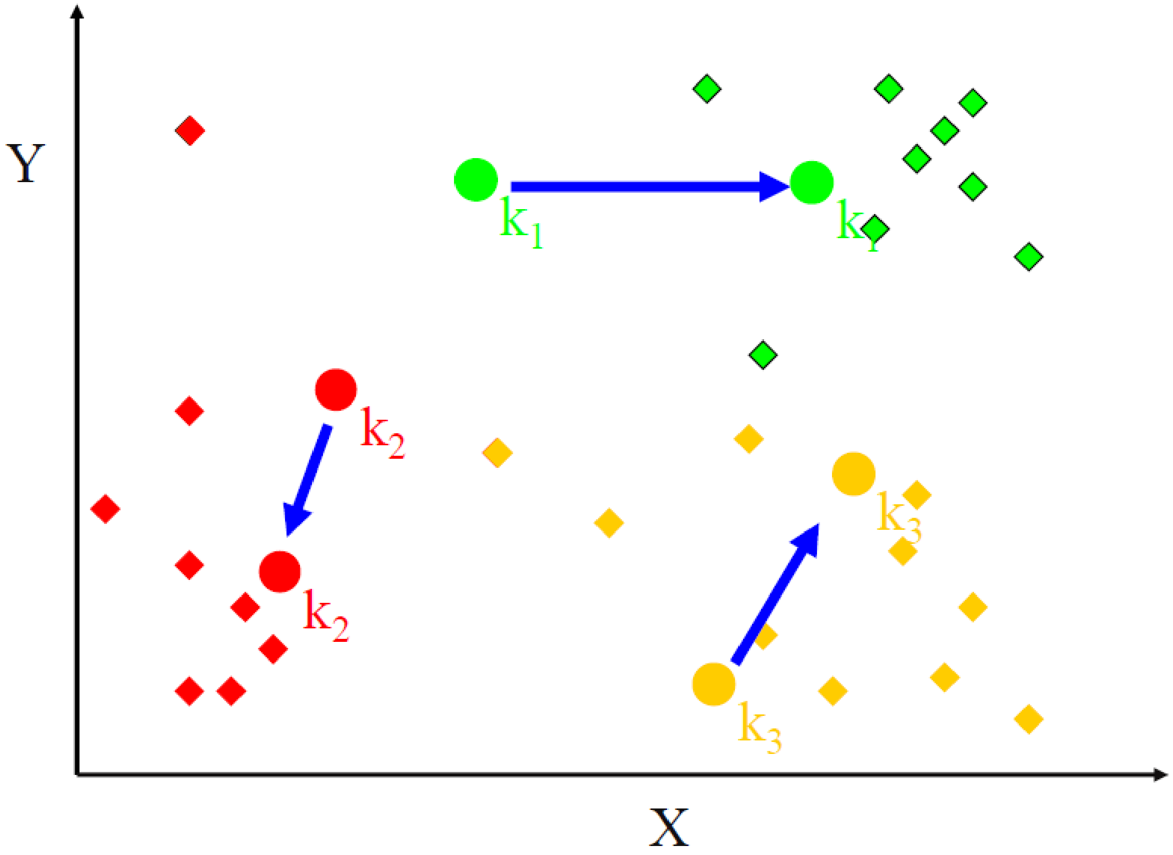
Assign each object to the cluster it is most similar to

Example: step 2.2 (iteration 1)



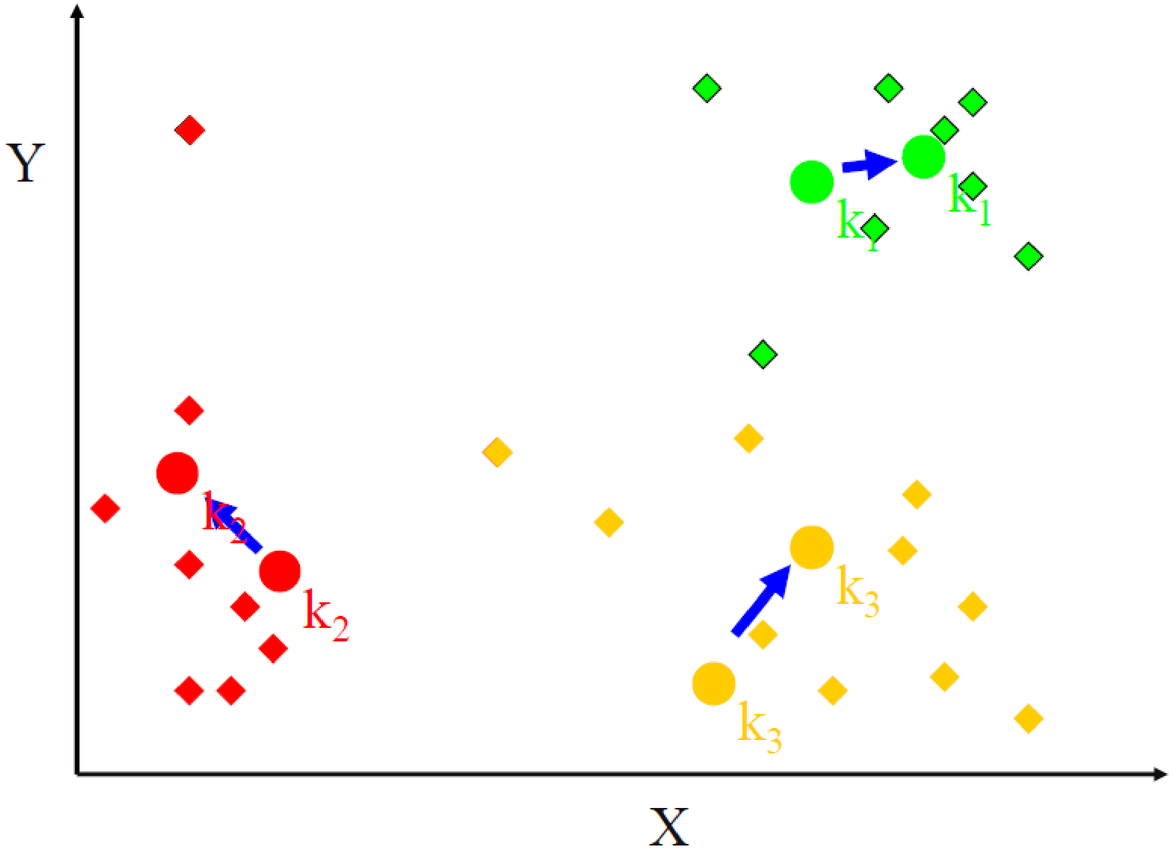
Calculate the new cluster centroid

Example: step 2.1 (iteration 2)



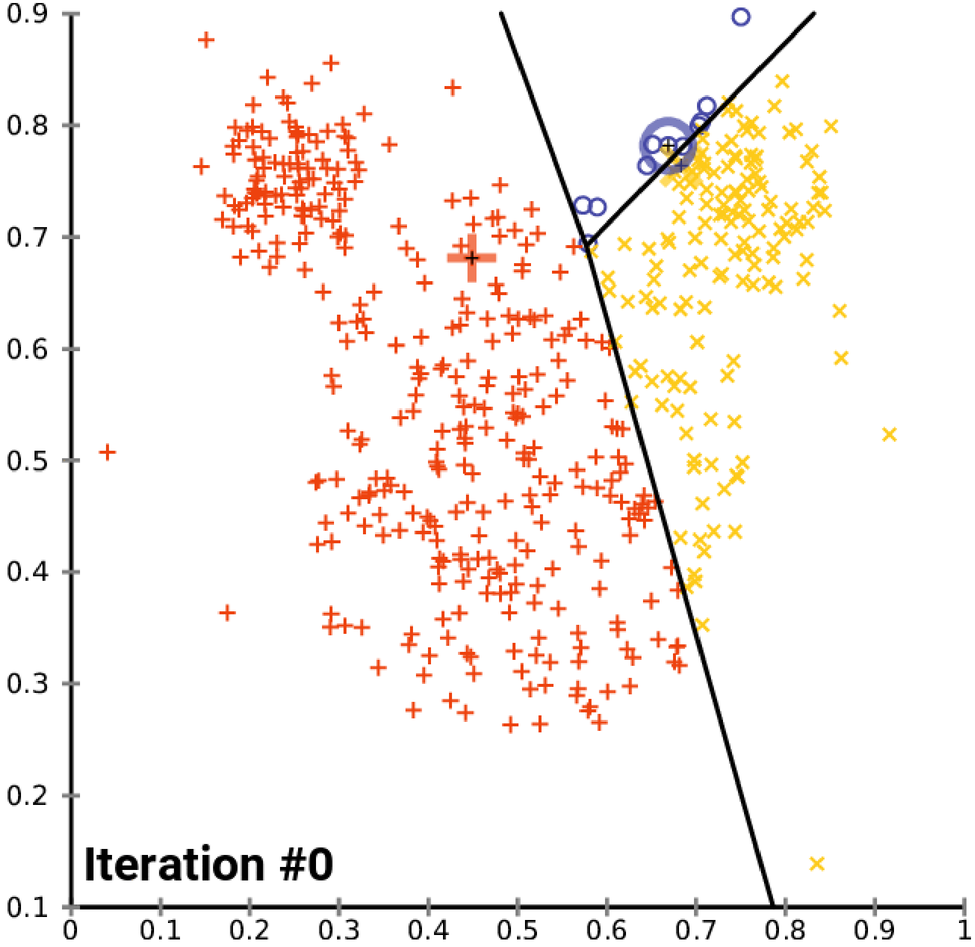
Assign each object to the cluster it is most similar to

Example: step 2.2 (iteration 2)



Calculate the new cluster centroid

How K- means works



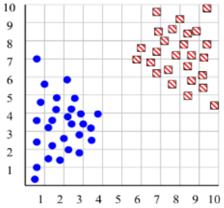
https://upload.wikimedia.org/wikipedia/commons/e/ea/K-means_convergence.gif

Advantages and disadvantages of K- means

Benefits	Disadvantages
• Available in most data analysis tools • Simple to use (requires only one parameter, *k*) • Efficient (fast and converges guaranteed) • Easily interpretable (the centroids represent the cluster profile)	• Parameterization (it is necessary to establish *k*) • Stochastic (different solutions are obtained with different initializations) • Can get “stuck” in a local optimum • Clusters are mutually exclusive (each item can only belong to one cluster) • Only allows numeric variables • Difficulties dealing with noise and *outliers* • Identifies spherical clusters

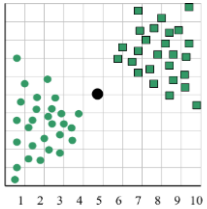
K determination

Given a given dataset :

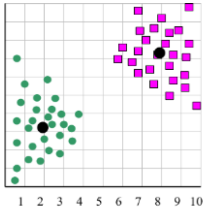


How many clusters to use?

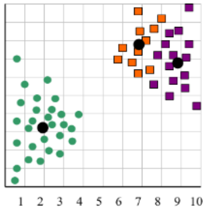
- 1?



- two?

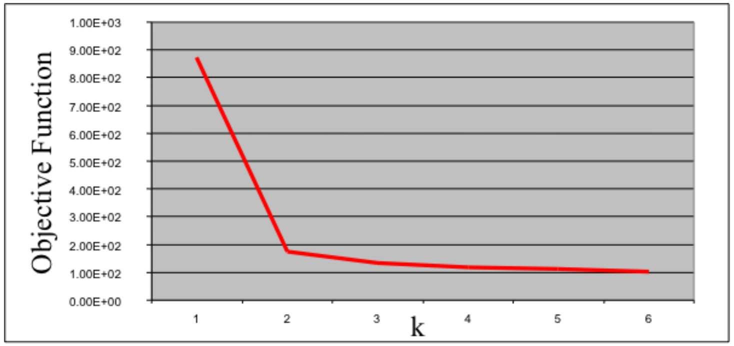


- 3?



“Elbow/knee method”

- Run the algorithm for several k (1, 2, 3, …)
- For each k, calculate the value of the objective function
  - Ex: In k- means , the distance of the points to the centroids
- Graphically view the result
- Choosing ok which represents an abrupt change



DEPARTAMENTO CIÊNCIA E TECNOLOGIA

K- medoids

Differences compared to k- means

K- means is very sensitive to outliers

Example:

$média(1,3,5,7,9) = 5$

$média(1,3,5,7,9,1009) = 172$

K- medoids uses as a “centroid” (here, it is called a medoid) the central point of the cluster (median) instead of the mean

$mediana(1,3,5,7,9) = 5$

$mediana(1,3,5,7,9,1009) = 6$

medoids

Input:

- O: Set of *n* objects
- K: number of clusters to create

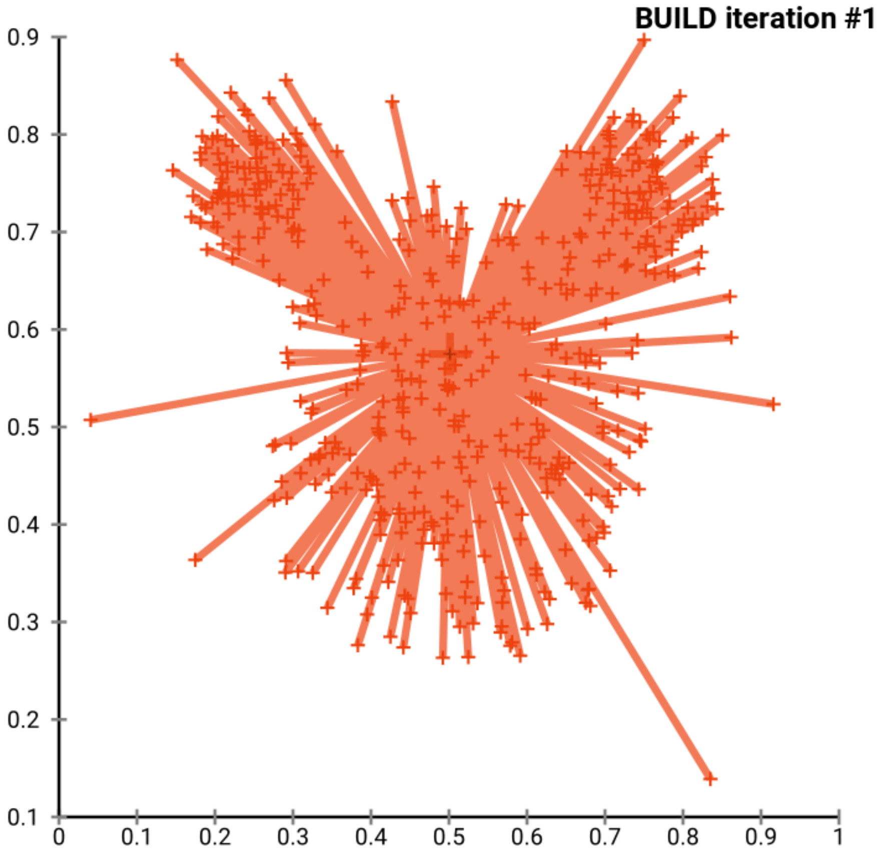
Steps:

1. Randomly choose *K* medoids m_k
2. Repeat until no changes are made:
    1. Assign each object to the medoid $m_k$closer
    2. Calculate the distortion D(sum of the " *dissimilarities* " of all points to the nearest medoids)
    3. For every non- medoid point x:
        1. Swap $m_k$with $x$and calculate the objective function
        2. Select the configuration with the lowest cost

Output:

- Set of *K* clusters

How K- means works



https://commons.wikimedia.org/wiki/File:K-Medoids_Clustering.gif

Density based clustering

Density based clustering

Clusters can be defined based on density-connected points

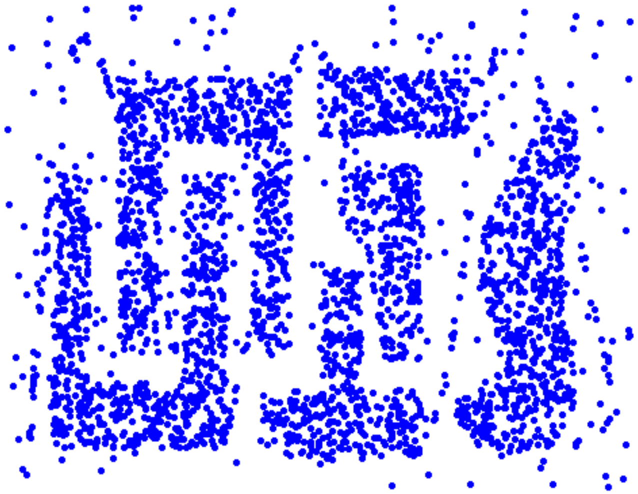
Allows discovering clusters with arbitrary shapes

Handles noise better

Requires density parameters as terminal condition

Examples:

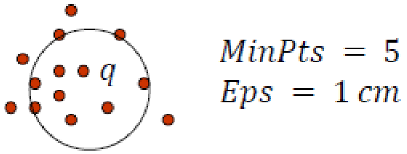
- DBSCAN
- OPTICS
- DENCLUE
- CLICK



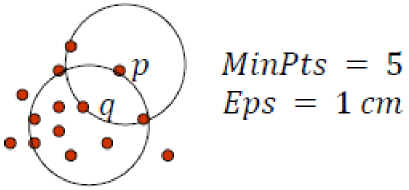
Concepts

The neighborhood of radius ε of an object is called ε *-neighborhood* . If it contains at least *MinPts* objects, the object is a *core object*

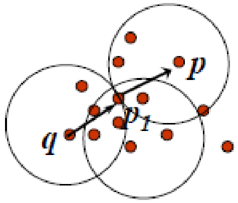
- ***Eps :*** maximum radius of the neighborhood
- ***MinPts :*** minimum number of points in the *Eps -neighborhood* of this point



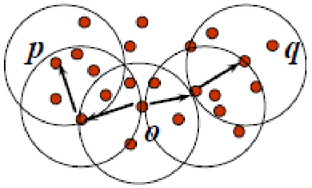
An object p is directly *density-reachable* from an object q if it p is within the ε *-neighborhood* of q and q is a *core object*



An object p is *density-reachable* from an object q with respect to (*Eps* , *MinPts*) if there is a chain of points $p_1, \dots, p_n$, with $p_1 = q$ It is $p_n = p$ such that p_{i+1} it is directly *density-reachable* from p_i



An object p is *density-connected* to object q with respect to (*Eps* , *MinPts*) if there exists an object o such that p and q are both density-reachable from o relative to (*Eps* , *MinPts*)



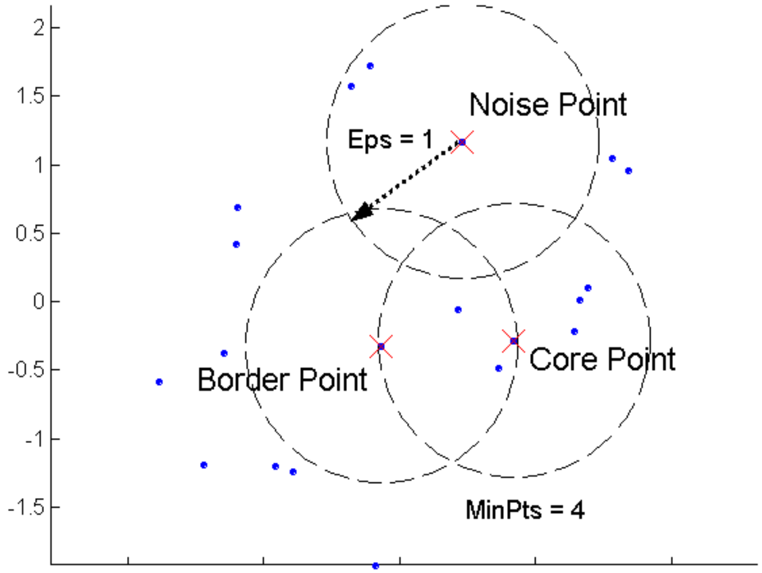
DBSCAN

DBSCAN

Extracts clusters as a set of *density-connected objects*

Concepts:

- ***Density-based cluster*** *:* set of *density-connected objects* which is maximum (cannot be expanded)
- ***Border point :*** has less *MinPts* with *Eps* , but is in the vicinity of a *core point*
- ***Noise point :*** any point that is neither *a core* nor *a border point point*



DBSCAN Algorithm

Input:

- O: Set of *n* objects
- Eps : maximum radius of the neighborhood
- MinPts : minimum number of points in *Eps -neighborhood*

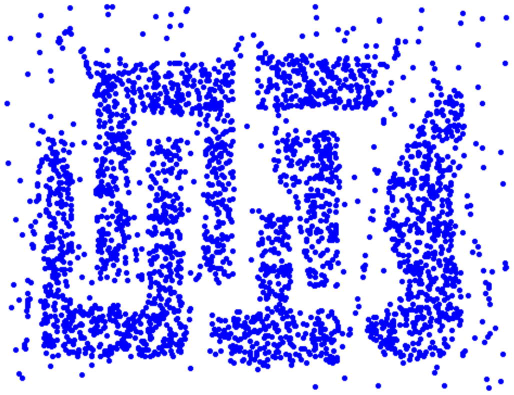
Steps:

1. Classify all points as core, border , or noise:
    1. Repeat until all points are sorted:
        1. Randomly select a point$p$
        2. density points reachable from $p$data $Eps$and$MinPts$
            1. If it $p$is a core point , a cluster is formed
            2. If it p's a border point , there are no density points reachable from p, visit next point
2. Eliminate noise points
3. Place an “edge” between all core points that are less than Eps
4. Turn each group of connected core points into a cluster
5. Assign each border point to one of your core point clusters

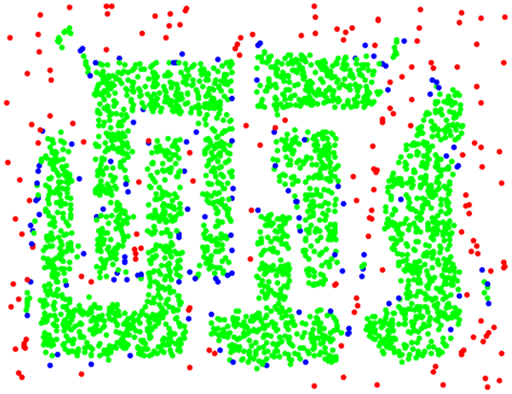
Output:

- cluster set

DBSCAN: example



original data

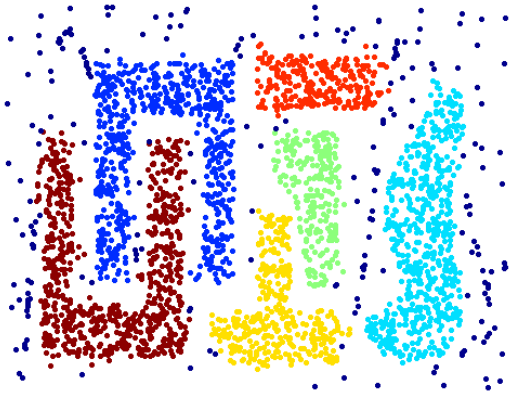


Points:

core (green)

border (blue)

noise (red)



Clusters

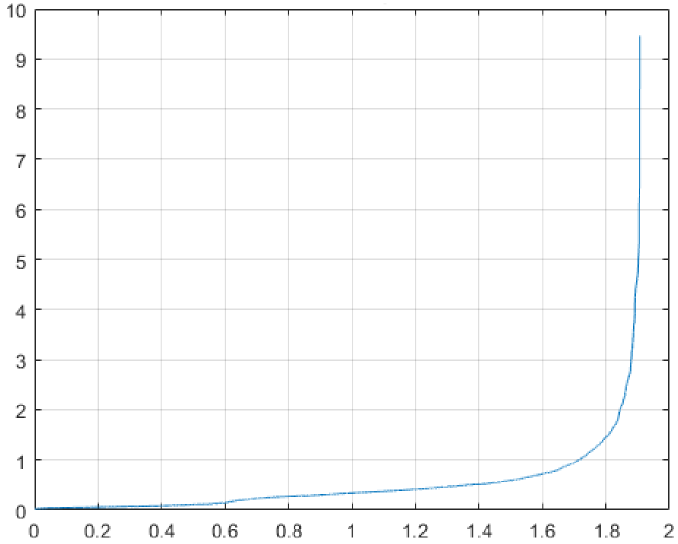
Choice of *Eps* and *MinPts*

MinPts is chosen:

- As a rule, based on domain knowledge
- If there is no domain knowledge, the rule of thumb is that $MinPts \geq D$, $D$is the number of dimensions of the data
    - 2D→ MinPts = 4
    - *D→ MinPts = 2 × D

Eps is chosen based on the distance behavior of the $k$nearest neighbors, $k = MinPts$:

- Select$\mathrm{MinPts} = \ k$
- Calculate distances from each point to its kºnearest neighbor
- Sort distances and visualize them graphically
- Follow the “elbow/knee rule”



The change in behavior is seen at approximately y=2:

Choose Eps = 2

Advantages and Disadvantages of DBSCAN

Benefits	Disadvantages
• Generate clusters of arbitrary shapes • (Almost) deterministic • Robust to outliers	• computational complexity • Need to establish parameters • Difficulties in interpretation

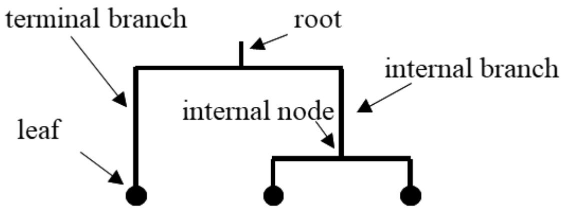
Hierarchical Clustering

Hierarchical

Purpose: create a decomposition of objects according to a certain criteria

Create a dendogram :

binary tree to evaluate/discriminate examples from a dataset



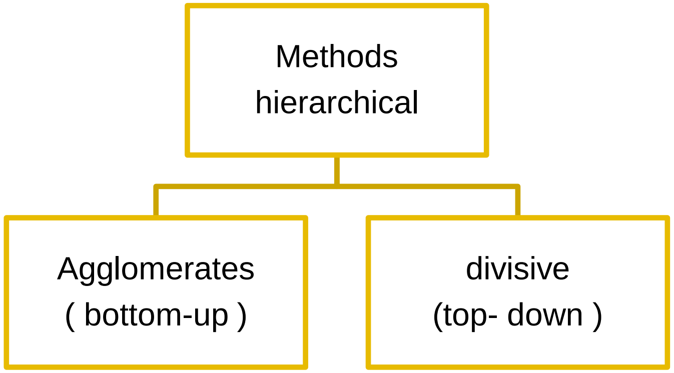
Types of Hierarchical

Number of different dendograms possible with n items: *(2n -3)!/[(2(n -2)) (n -2)!]*

n	Dendrograms
two	1
3	3
4	15
…	…
10	34,459,425

Problem: It is not possible to test all alternatives.

Solution:



Methods hierarchical

Agglomerates (bottom-up)

divisive (top- down)

1. Starts with n clusters (1 item in each cluster)
2. Finds the best pair of objects to group together
3. Repeat until all objects are grouped

1. Starts with all items in a cluster
2. Considers all partitions that divide the cluster into 2
3. choose the best
4. Apply the same process recursively to both partitions

Bottom-up

Input:

- O: Set of *n* objects

Steps:

1. Start with *n* items and a metric (eg Euclidean distance) of all pairs $\binom{n}{2} = \frac{n(n-1)}{2}$. Treat each item as a cluster
2. repeat for$i = n, n-1, n-2, \dots, 2$:
    1. Examine all distances between pairs of inter-cluster items in the *i* clusters and identify the pair of clusters that are least different (most similar). Merge the two clusters. The difference between the clusters indicates, on the dendrogram , the height at which the fusion should be placed.
    2. Recalculate the distances between the $i - 1$ remaining clusters

Output:

- cluster set

Calculation of the difference (*dissimilarity*)

Single linkage :

Minimal intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the smallest distance

Complete linkage :

Maximal intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the largest distance

Average linkage :

mean intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the average of the distances

centroid linkage :

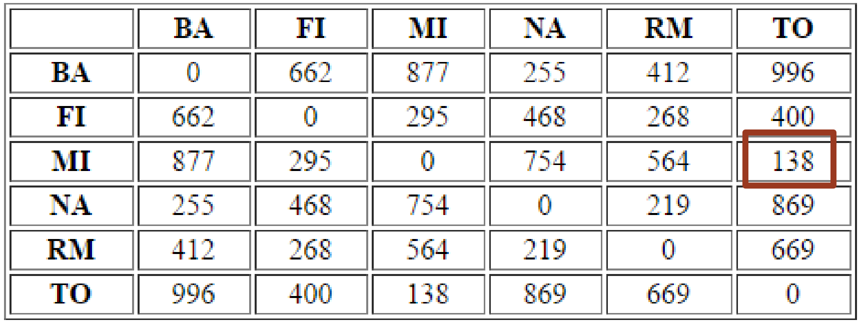
Difference between the centroid of cluster A and the centroid of cluster B

Single linkage (step 1)

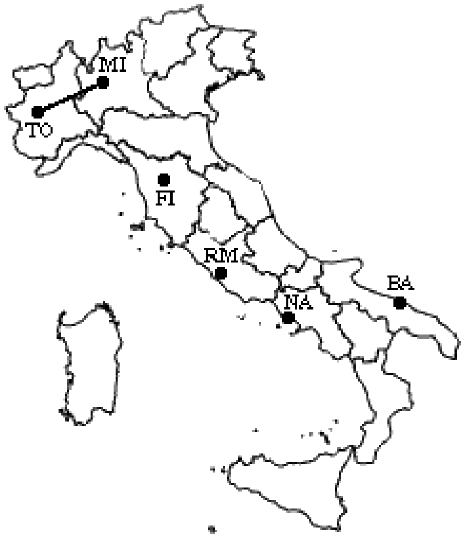
Minimal intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the smallest distance

MI
TO
FI
RM
NA
BA



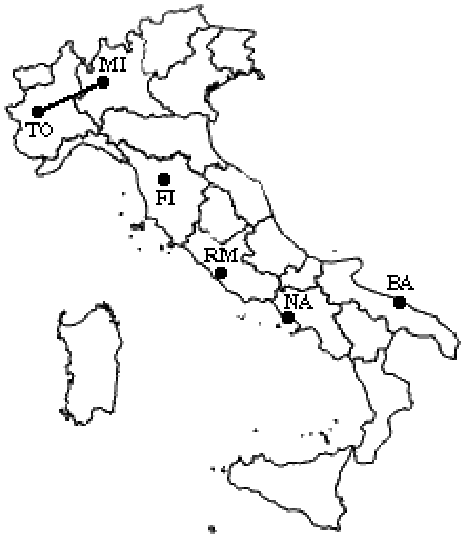
	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

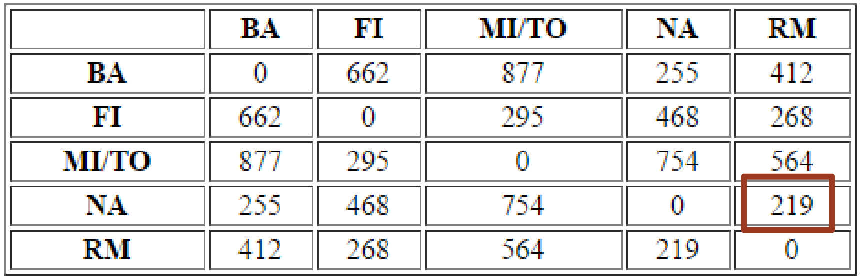


Single linkage (step 2)

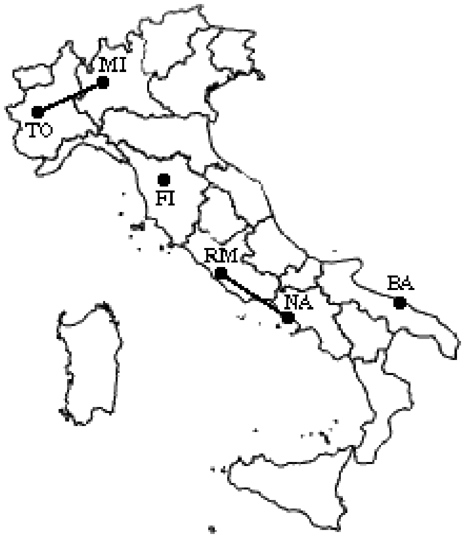
Minimal intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the smallest distance





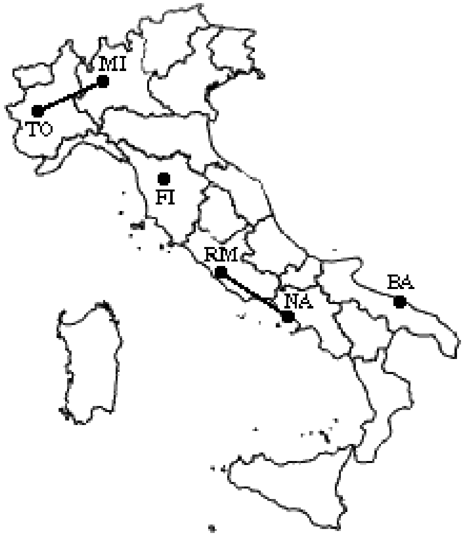
	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0



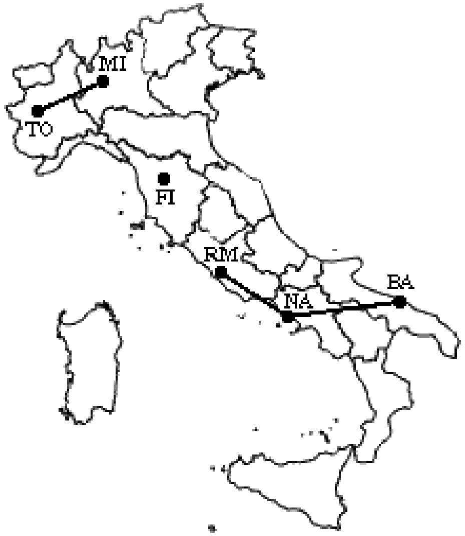
Single linkage (step 3)

Minimal intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the smallest distance



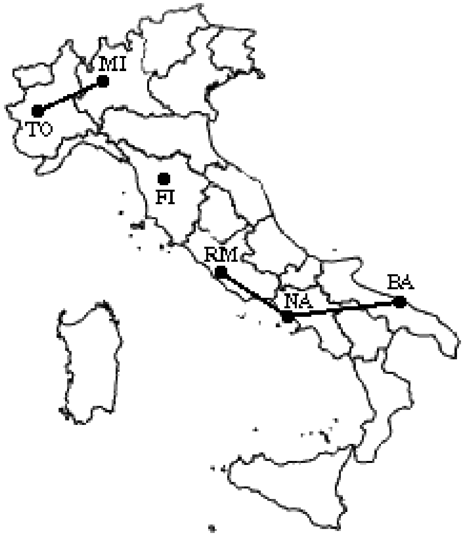
	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

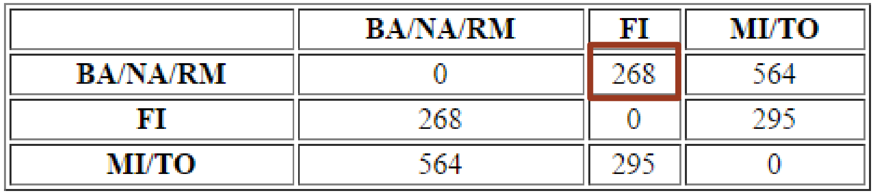


Single linkage (step 4)

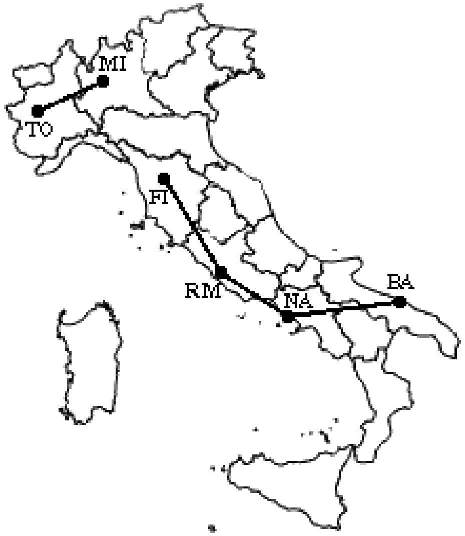
Minimal intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the smallest distance





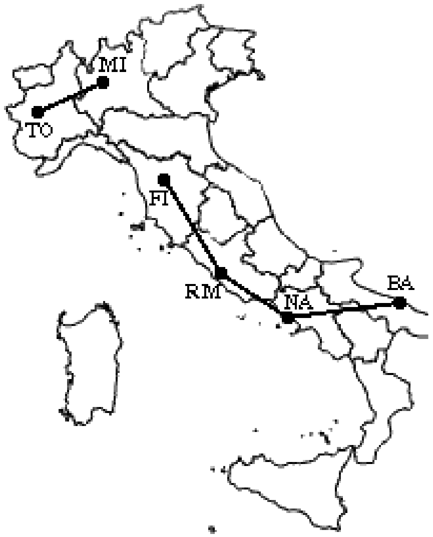
	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

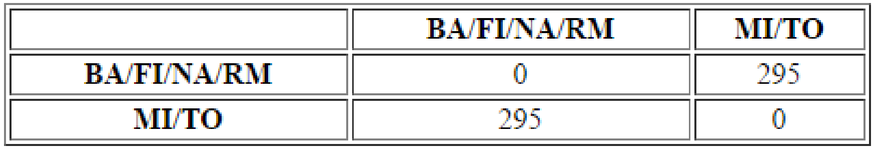


Single linkage (step 5)

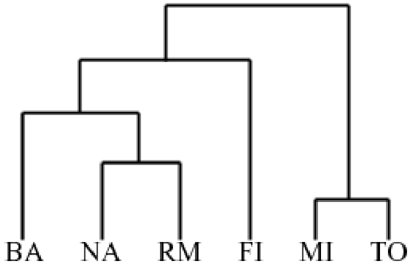
Minimal intercluster dissimilarity

Calculate the distances between items in clusters A and B (pairwise) and save the smallest distance





	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0



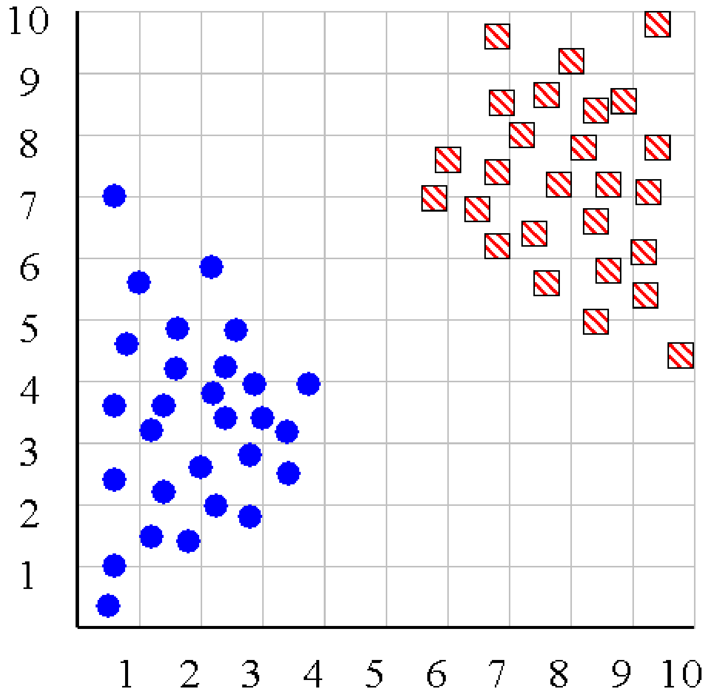
Advantages and disadvantages of *single* and *complete linkage*

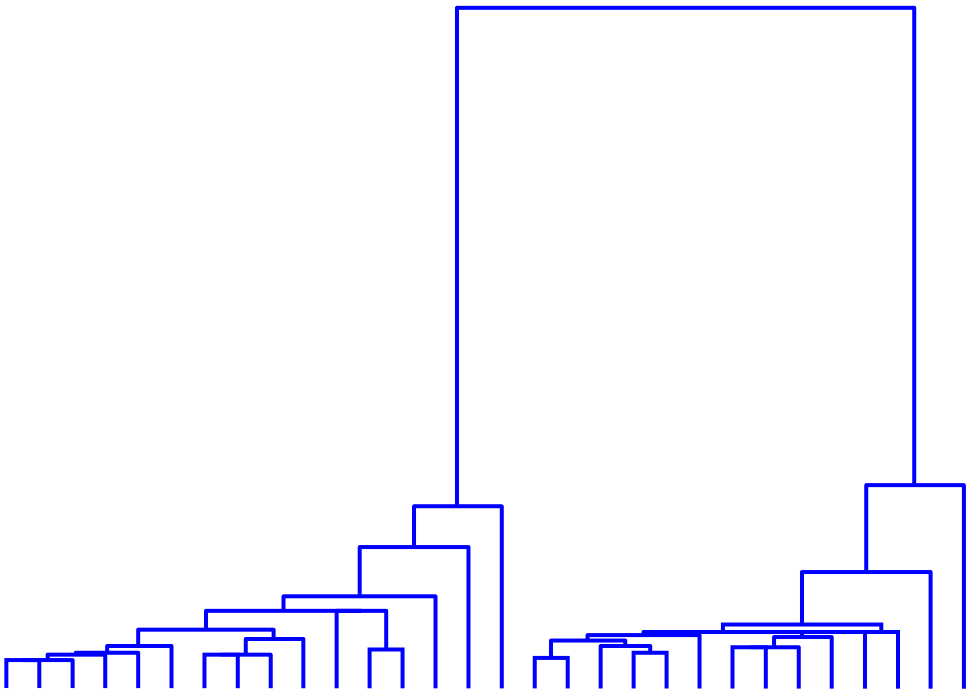
	Benefits	**Disadvantages**
Single Linkage	• Can handle non-elliptical clusters	• Sensitive to noise and outliers
Complete Linkage	• Robust to noise and *outliers*	• Part large clusters • Biased for spherical clusters

Determining the number of clusters

The “ideal” number of clusters is determined based on the dendrogram

Example: two very separated sub-trees suggest the existence of two clusters





Advantages and disadvantages of hierarchical methods

Benefits	Disadvantages
• It is not necessary to establish *k* • Simplicity of interpretation of hierarchies	• Computational complexity (runtime greatly increases with increasing number of items) • Can get “stuck” in a local optimum • Interpretation can be subjective

Grid clustering

grid algorithm clustering

Input:

- O: Set of *n* objects dispersed in space

Steps:

1. Creating the Grid Structure: Partitioning Space into a Finite Set of Cells
2. Calculate the density of each cell
3. Sort cells according to densities
4. Identify cluster centers
5. Consider the neighborhood of the centers as the cluster points

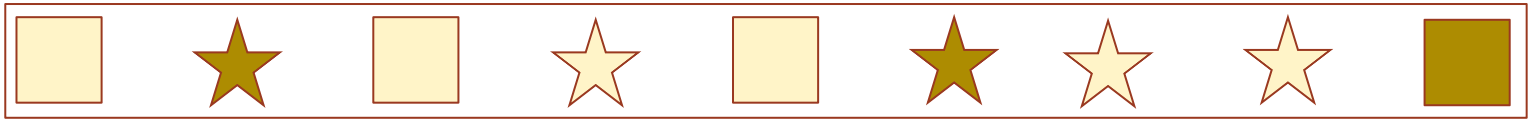
Output:

- cluster set

Other questions

Algorithm selection

Given a given dataset :



What algorithm to use?

- partition
    - K- means
    - K- medoids
    - …
- hierarchical
    - Aggregation + single linkage
    - Aggregation + complete linkage
    - …
- density-based
    - …
- model-based
    - …

Clustering

- Scalability
- Ability to handle different types of data
- Usability
- Ability to handle noise and *outliers*
- Sensitivity to the order of representation of items
- Possibility to incorporate user-defined constraints
- Interpretability of the result
- Availability in the tool used

Clustering

- Analyze intra- cluster homogeneity

- Analyze inter- cluster homogeneity

- Analyze the sensitivity of clusters
  - Ex: run several times k- means with different initializations and check the result
  - Ex: run several times k- means with slightly different samples and check the result

- Evaluate the “quality” of the resulting clusters:
  - To determine the trend of a data set (distinguish whether non-random structures exist in the data)
  - clustering results with known external results
  - To assess how well the clustering results fit the data without reference to external information
  - To compare the results of two different clusterings
  - To determine the “correct” number of clusters

Clustering evaluation : methods

- Calculate the correlation between **Similarity Matrix** and the **incidence Matrix** (1, if the points belong to the same cluster; 0, otherwise)
    - High correlation when points belonging to the same cluster are close
    - Not a very good metric for density-based clusters

- **Dunn's index** :$DI = \frac{\min(inter-cluster\ distance)}{\max(cluster\ size)}$
    - *DI* (better clustering) when:
        - Inter-cluster distances are high (better separation)
        - Small clusters (more compact)

Clustering Assessment : Metrics

- ***Internal Indexes:*** used to measure the quality of a given cluster, without external information
- ***External Indexes:*** used to measure the extent to which *labels* determined by *clustering* resemble given *labels*
- ***Relative Indexes:*** used to compare two *clustering algorithms*

Internal Indexes

- ***Sum of squared Errors*** (relative to the centroid)
  - Good for comparing two *clusterings* , or two clusters (*average* SSE)
  - Can be used for the “elbow method”

- ***Cohesion*** **:** measures the affinity between objects in the same cluster
  - *Within Cluster SSE* (WSS)

$$WSS = \sum_{k} \sum_{x_n \in c_k} (x_n - c_k)^2$$

- ***Separation*** **:** measures how well separated a cluster is from others
  - *Between cluster* SSE (BSS), where $|c_k|$is the cluster size $k$and $\bar{c}$is the average of all centroids

$$BSS = \sum_{k} |c_k|(\bar{c} - c_k)^2$$

- ***silhouette coefficient*** (of each item):$s = \frac{b-a}{\max(a,b)}$,
  - a: average distance of the item to the other items of the same cluster
  - b: average distance from the item to the items in the closest cluster

  - Values between 0 and 1. The closer to 1, the better
  - *average* can be calculated *silhouette* of an algorithm

External Indexes (knowing classes) (1)

For each cluster k, relative to the class $j$and having

- $N$is the total number of elements to be grouped
- $N_k$is the number of cluster elements$k$
- $N_j$is the number of elements in the class$j$
- $N_{kj}$is the number of cluster elements $k$belonging to the class$j$

$$precision(k,j) = p_{kj} = \frac{N_{kj}}{N_k}$$

$$recall(k,j) = r_{kj} = \frac{N_{kj}}{j}$$

$$F = \frac{2}{\frac{1}{p} + \frac{1}{r}} = \frac{2pr}{p + r}$$

$$purity, \qquad p_k = \max_j p_{kj}$$

$$purity, \qquad p = \sum_{k=1}^{K} \frac{N_k}{N} p_k$$

External Indexes (knowing classes) (2)

Entropy : measures the extent to which a cluster contains elements of the same class

- Let $p_{kj} = \frac{N_{kj}}{N_k}$the probability that a cluster member $k$belongs to the class$j$
- be $L$the number of classes

Entropy of a cluster:

$$e_k = -\sum_{j=1}^{L} p_{kj} \log_2 p_{kj}$$

Total clustering entropy is the size-weighted sum of the clusters

$$e = \sum_{k=1}^{K} \frac{N_k}{N} e_k$$

External Indexes (knowing classes) (3)

Jaccard Similarity

The *cluster similarity matrix* ideal has entries with value 1 if two objects belong to the same cluster and 0 otherwise

the *class similarity matrix* ideal has entries with value 1 if two objects belong to the same class and 0 otherwise

We can use similarity between binary vectors

- f_{00}: number of peers with different class and different cluster
- f_{01}: number of pairs with different class and same cluster
- f_{10}: number of pairs with the same class and different cluster
- f_{11}: number of pairs with equal class and equal cluster

$$Jaccard = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

UPT

UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.