MATHEMATICAL TOOLS FOR DATA SCIENCE

Leila GHARSALLI (leila.gharsalli@ipsa.fr)

IPSA, AERO 4

2023-2024

GOALS OF THE COURSE

- *Instructor : Leila GHARSALLI*
- *mail to : leila.gharsalli@ipsa.fr*

- Provide a general introduction to data mining and data science.

- Introduce some important tools for solving problems in data science.

- Grading : Participation in class (practical work) as well as a final exam will be graded.

- Main background needed: basic notions in probabilities and statistics, programming skill.

CONTENT OF THE COURSE

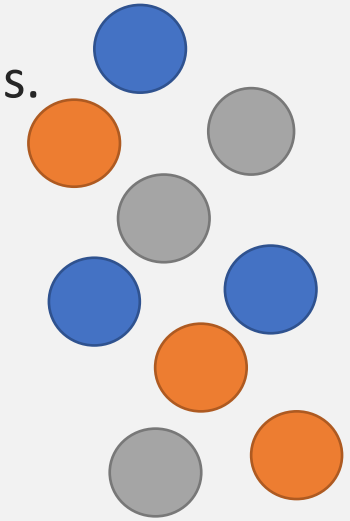
1. Introduction
2. Linear regression
3. Sparse regression
4. Classification
5. Principal Component Analysis
6. Clustering
7. Density estimation

INTRODUCTION

INTRODUCTION

Unsupervised learning: no knowledge of class output or value,

- Data Is unlabeled with a class and value are unknown,
- Goal: determine data patterns – groupings,
- Self-guided learning algorithm (k-means, genetic algorithms, clustering approaches…),
- Example: user behavior analysis.

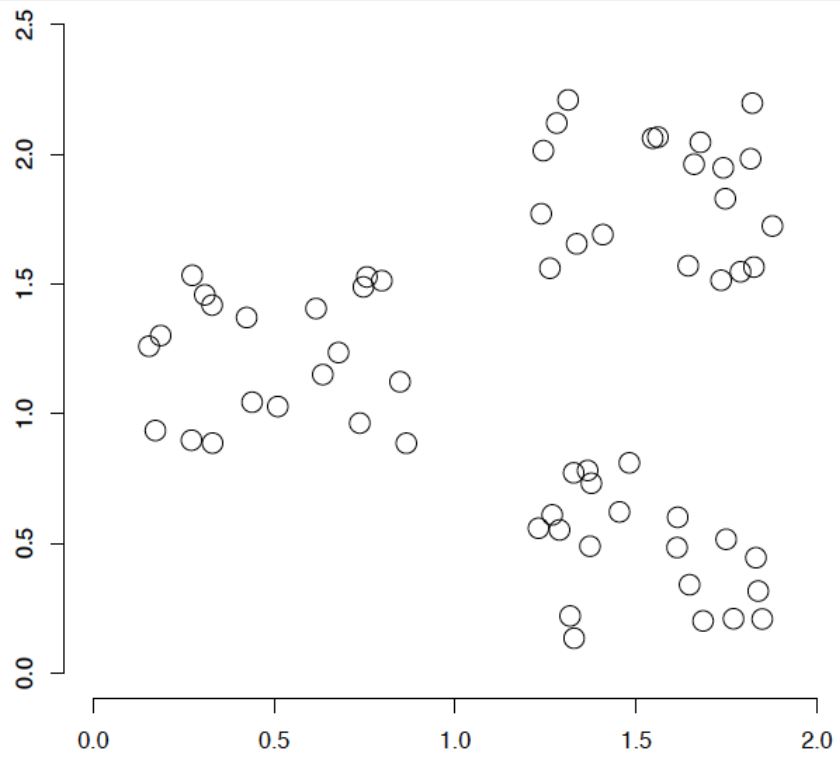


?

INTRODUCTION

- **Unsupervised - Clustering**

INTRODUCTION



How would you design an algorithm for finding the three clusters in this case?

INTRODUCTION

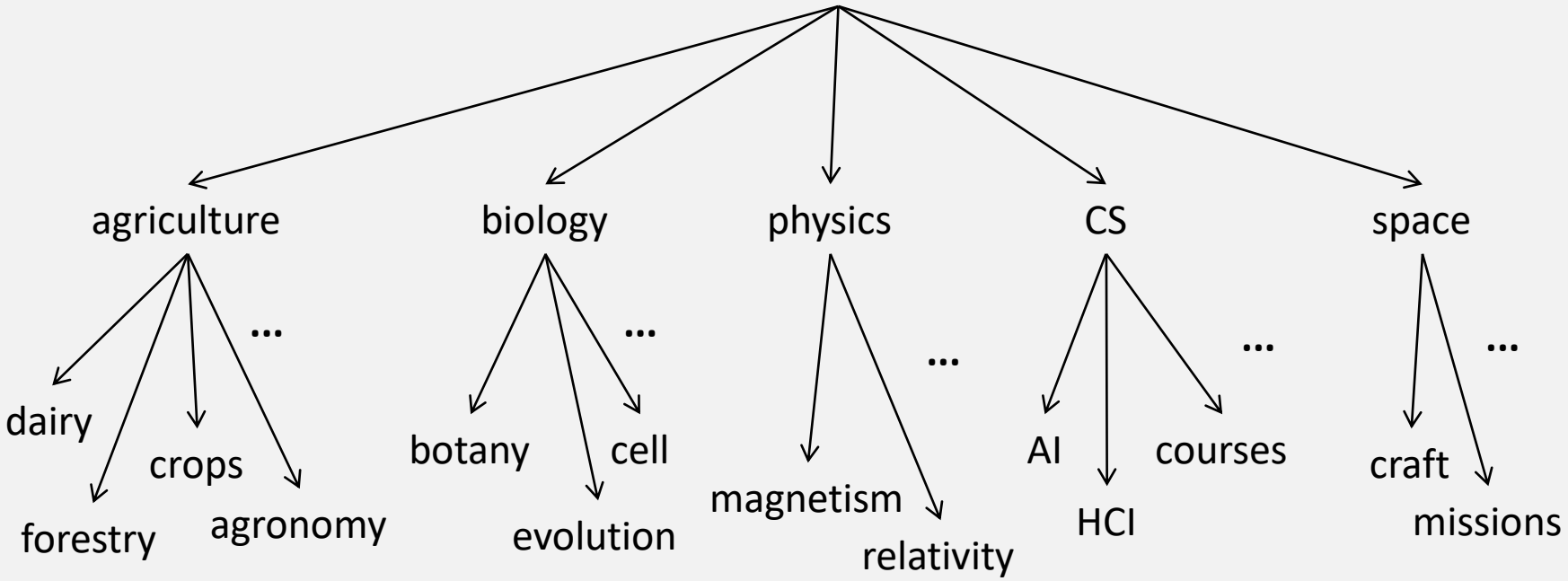
CLUSTERING

CLUSTERING

- **The commonest form of unsupervised learning**
  - A common and important task that finds many applications in data science,
  - Pre-processing before other analyzes dimension reduction, regression in high-dimensional data work on the characteristics of the clusters rather than on all the data, compression, efficiently find the nearest neighbors.

- **Clustering: the process of grouping a set of objects into classes of similar objects**
  - Documents within a cluster should be similar.
  - Documents from different clusters should be dissimilar.

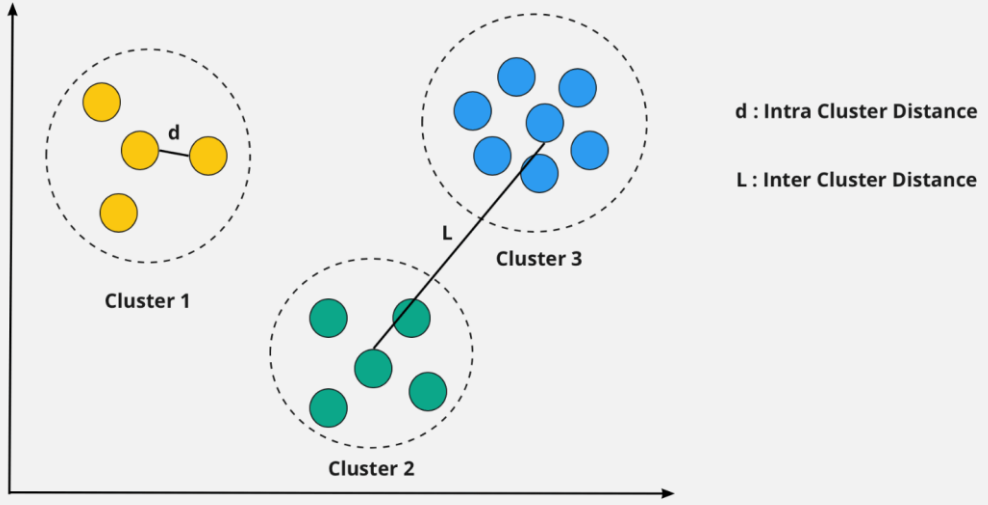
CLUSTERING



www.yahoo.com/Science

CLUSTERING

- A good method of clustering makes it possible to guarantee
  - Great intra-group similarity,
  - Low intergroup similarity (dissimilar when they belong to different groups).
- The quality of a clustering therefore depends on the measurement of **similarity** used by the method and of its implementation.



CLUSTERING

- To define the homogeneity of a group of observations, it is necessary to measure the resemblance between two observations.
- **Dissimilarity function**: it is a function d which for any pair (x_1, x_2)associates a value in $\mathbb{R}$+ such that:

$$d(x_1, x_2) = d(x_2, x_1) \geq 0$$
$$d(x_1, x_2) = 0 \rightarrow x_1 = x_2$$

The lower the measurement is, the more similar are the points.

- **Similarity function**: it is a function s which for any pair (x_1, x_2) associates a value in $\mathbb{R}$+ such that:

$$s(x_1, x_2) = s(x_2, x_1) \geq 0$$
$$s(x_1, x_1) \geq s(x_1, x_2)$$

The larger the measure is, the more similar are the points.

CLUSTERING

- By finding the **eigenvalues** and **eigenvectors** of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the **strongest correlation in the dataset**.
- **The covariance** measures the linear connection that may exist between a pair of statistical variables or a pair of quantitative random variables.
- This is **the principal component**.
- PCA is a useful statistical technique that has found application in:

  - fields such as face recognition and image compression.

  - finding patterns in data of high dimension.

CLUSTERING

Examples of distances:

- Euclidean distance (numerical data): $\boldsymbol{d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} \left(x_i^{(k)} - x_j^{(k)} \right)^2}}$

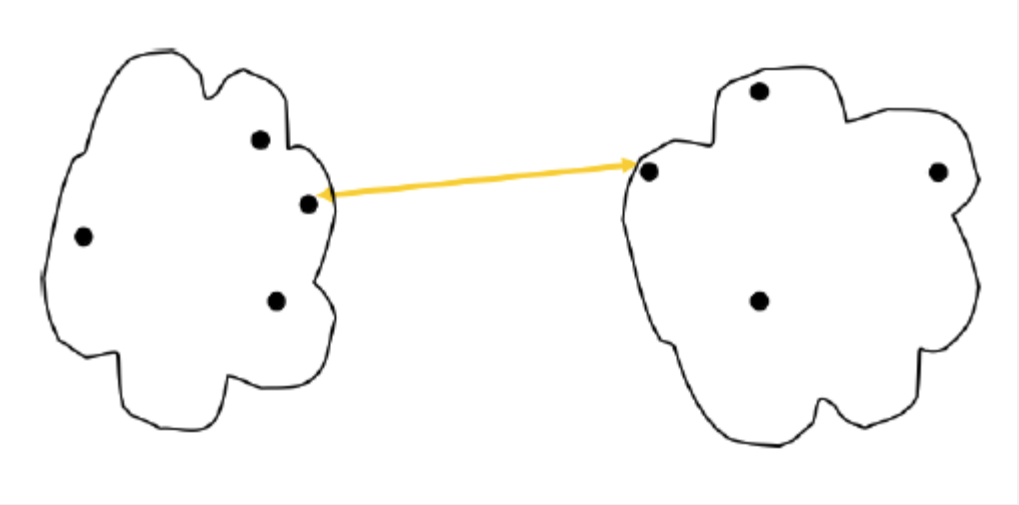
- Manhattan distance: $\boldsymbol{d(x_i, x_j) = \sum_{k=1}^{d} \left| x_i^{(k)} - x_j^{(k)} \right|}$

- Minkowski distance: $\boldsymbol{d_p(x_i, x_j) = \left(\sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|^p \right)^{\frac{1}{p}}}$

CLUSTERING

Measure of distance between two clusters or classes: **Nearest neighbor:**

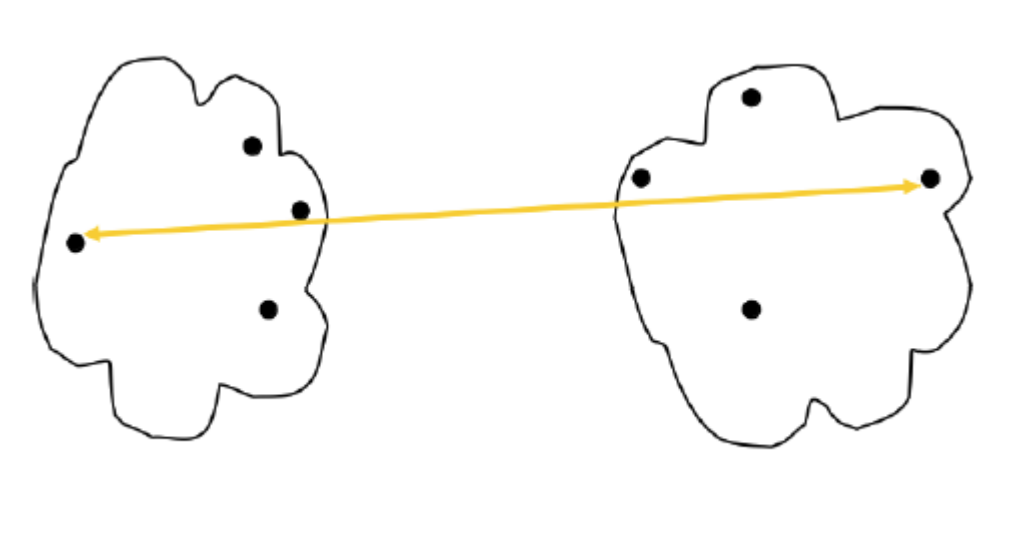
$$\boldsymbol{D_{min}(C_1, C_2) = min\{d(x_i, x_j), x_i \in C_1, x_j \in C_2\}}$$



CLUSTERING

Max diameter:

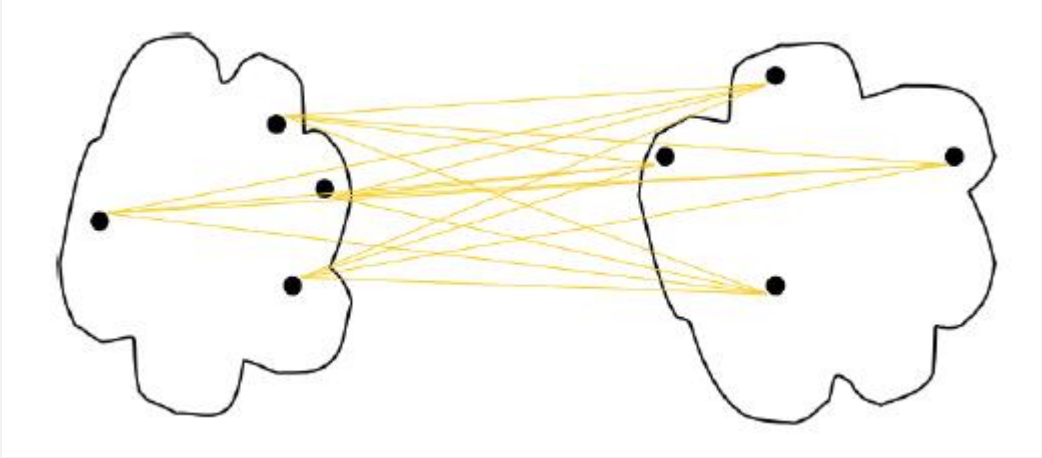
$$D_{max}(C_1, C_2) = max\{d(x_i, x_j), x_i \in C_1, x_j \in C_2\}$$



CLUSTERING

Average distance:

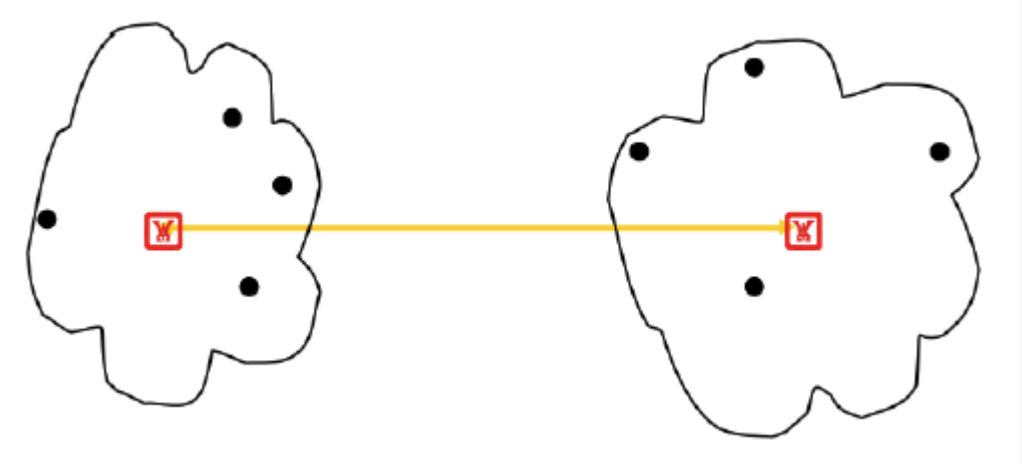
$$D_{moy}(C_1, C_2) = \frac{\sum_{x_i \in C_1} \sum_{x_j \in C_2} d(x_i, x_j)}{n_1 n_2}$$



CLUSTERING

Distance between centers of gravity:

$$D_{cg}(C_1, C_2) = d(\mu_1, \mu_2)$$



CLUSTERING

Evaluation of the quality of the cluster:

- **Intra-cluster cohesion (compactness):**

- Cohesion measures how close points in a cluster are to the cluster's center of gravity.

- Sum of Squared Errors (SSE) is a possible measure.

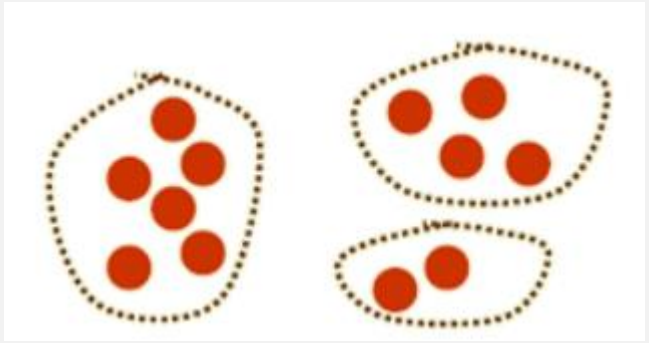
- **Inter-cluster separation (isolation):**

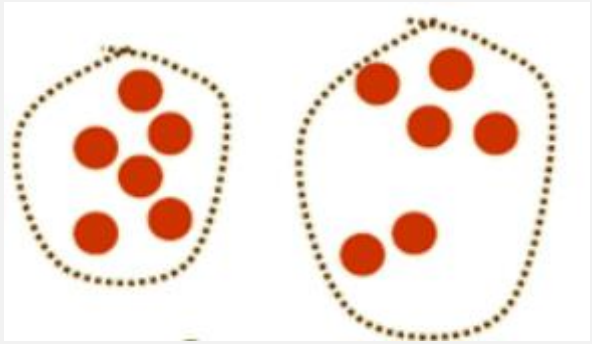
- Separation means that different cluster centroids should be far from each other.

- In most applications, expert judgment is still the key.

CLUSTERING

How many clusters?



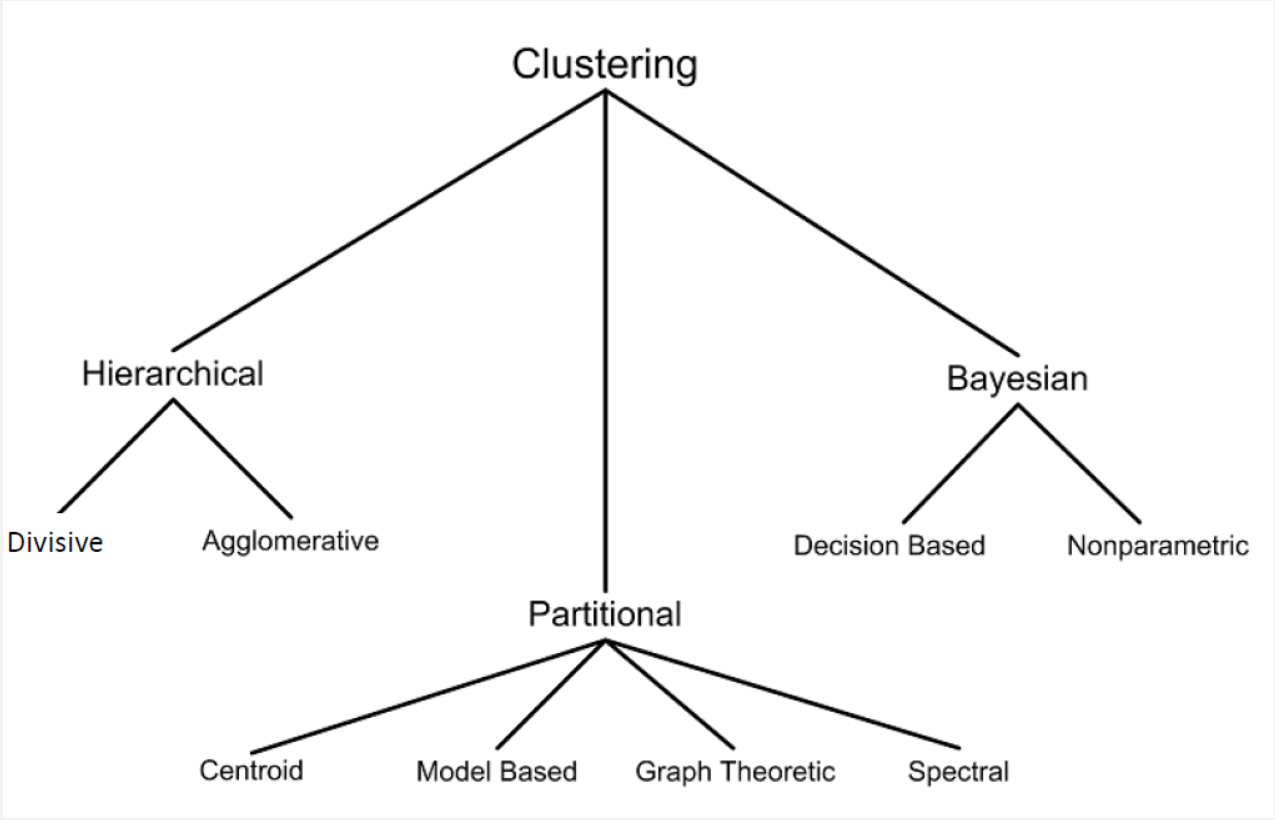


Possible approaches:

- Set the number of classes *a priori.*
- Find the best clustering relative to a criterion function (the number of classes may vary).

CLUSTERING

- **Clustering techniques:**



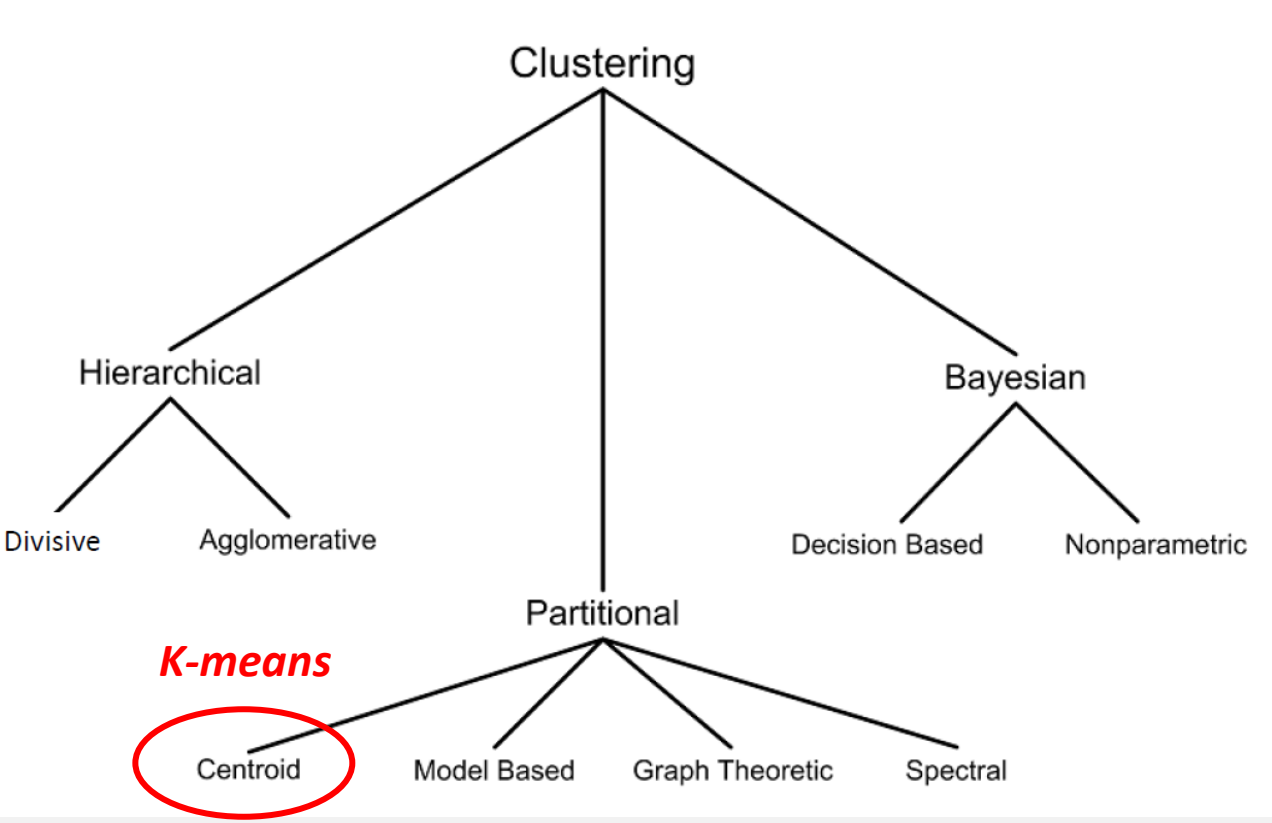
CLUSTERING

- **Clustering techniques:**

1. Hierchical clustering: set of nodes organized as a tree. Each node (of the tree structure (except leaf nodes) is the union of its children (subclusters). The root of the tree is the cluster containing all the objects.
2. Clustering by partitioning: division of data into subsets not superimposed (then evaluate them according to certain criteria).
3. Bayesian clustering: generation of posterior distributions on the collection of all data partitions.

K-MEANS CLUSTERING

K-MEANS CLUSTERING



K-MEANS CLUSTERING

- K-means algorithm is a famous clustering algorithm that is ubiquitously used.

- *Kmeans* algorithm is an iterative algorithm that tries to partition the dataset into Kpre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.

- It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.

K-MEANS CLUSTERING

1. Choose the number of clusters (K) and obtain the data points

2. Place the centroids $c_1, c_2, \ldots, c_k$ randomly

3. Repeat steps 4 and 5 until convergence or until the end of a fixed number of iterations:

4. for each data point x_i :

    - find the nearest centroid $(c_1, c_2, \ldots, c_k)$

    - assign the point to that cluster

5. for each cluster $j = 1..k$

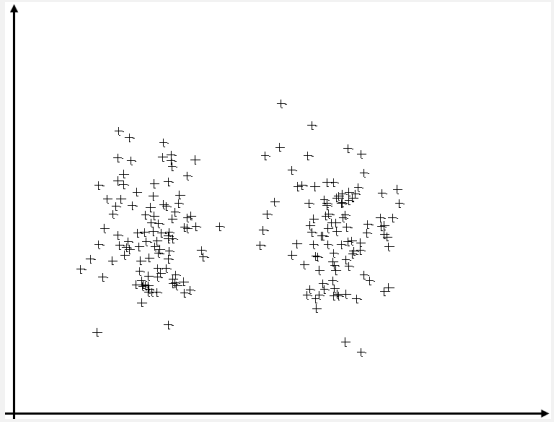
    - new centroid = mean of all points assigned to that cluster

6. End

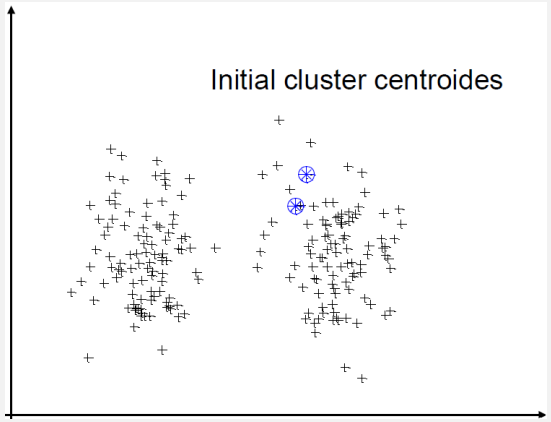
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

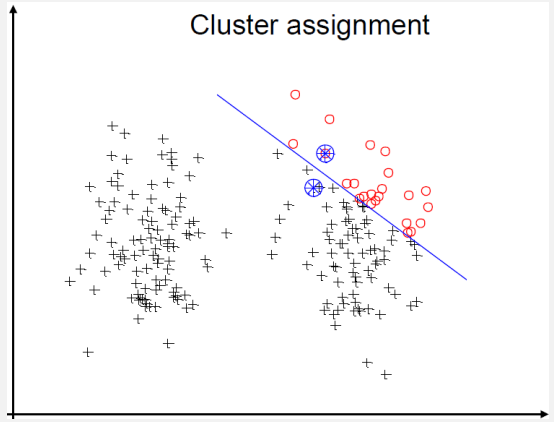
K-MEANS CLUSTERING

Step 1 consists of rallying each example to the nearest center. After this step, we have K Clusters (here 2 clusters).



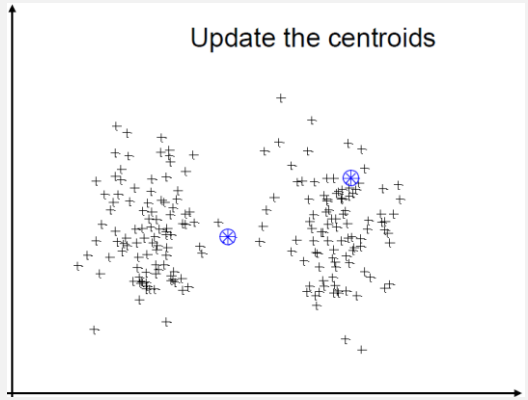
Step 2 consists in moving the centers in the middle of their Cluster.

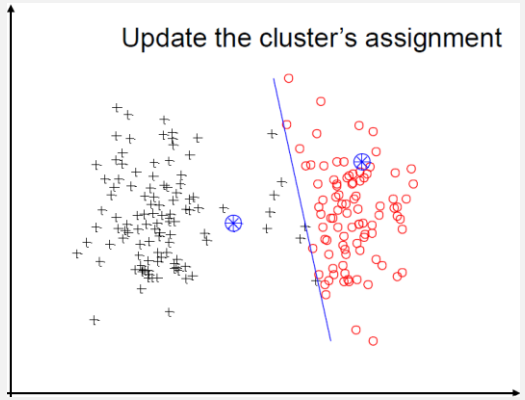


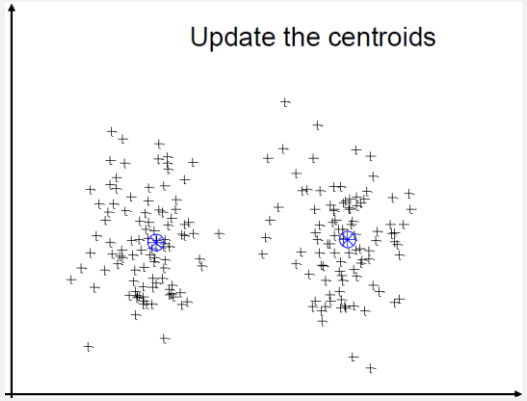


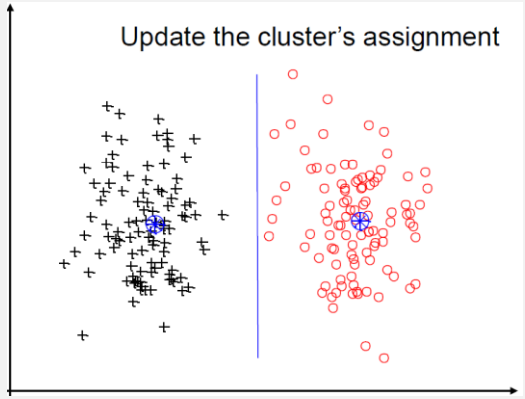
K-MEANS CLUSTERING

We repeat steps 1 and 2 in a loop until the centers no longer move.









K-MEANS CLUSTERING

How to choose K?

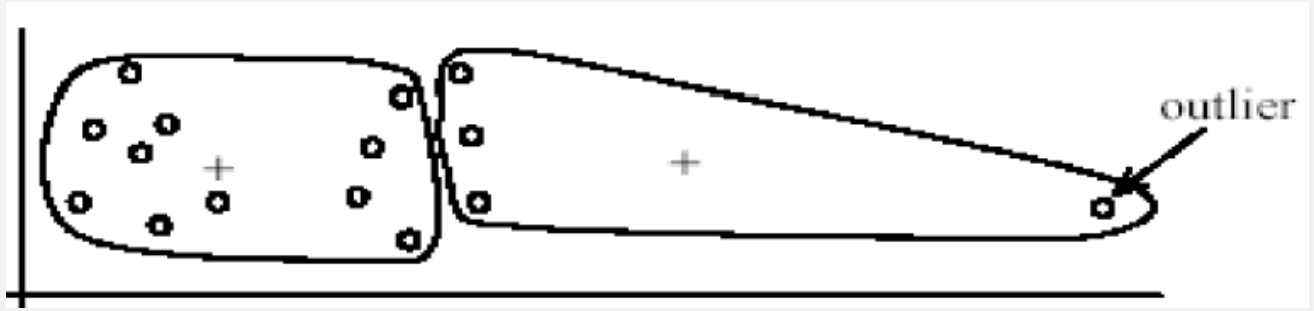
- A large number K can lead to an overly fragmented partitioning of the data which will prevent the discovery of interesting patterns in the data.

- A too small number of clusters will potentially lead to having too general clusters containing a lot of data in this case , there will be no patterns to discover.

- The most common way to choose the number of clusters is to run KMeans with different values of K and calculate the variance of the different clusters:

$$V = \sum_{j} \sum_{x_i \to c_j} D(c_j, x_i)^2$$

K-MEANS CLUSTERING



K-MEANS CLUSTERING

How to handle the outliers?

- Remove some data points that are much farther from the centroids than other data points.
- To be sure, one can watch for possible outliers over a few iterations and then decide to remove them.
- Perform random sampling by choosing a small subset of data points, the chance of selecting an outlier is much smaller.
- Assign the rest of the data points to the clusters by distance or similarity comparison, or classification.

K-MEANS CLUSTERING

In this exercise, we will use the non-hierarchical clustering algorithm *K-means* to classify a randomly created data set.

1. Import the python libraries necessary for reading, writing and viewing data.
2. Generate random data in a two-dimensional space of 100 points and divide them into two groups of 50 points each.
3. Visualize the data displayed on a two-dimensional space.
4. Use K-means algorithm from Scikit-learn library to process randomly generated data (k= number of clusters =2).
5. Find the centers of the clusters and visualize them graphically.
6. Use the *Kmean.labels_* command to display data classified into two clusters. What do you notice?
7. Use the model to predict the cluster of the new entry [−3,−3]. Comment on the result.