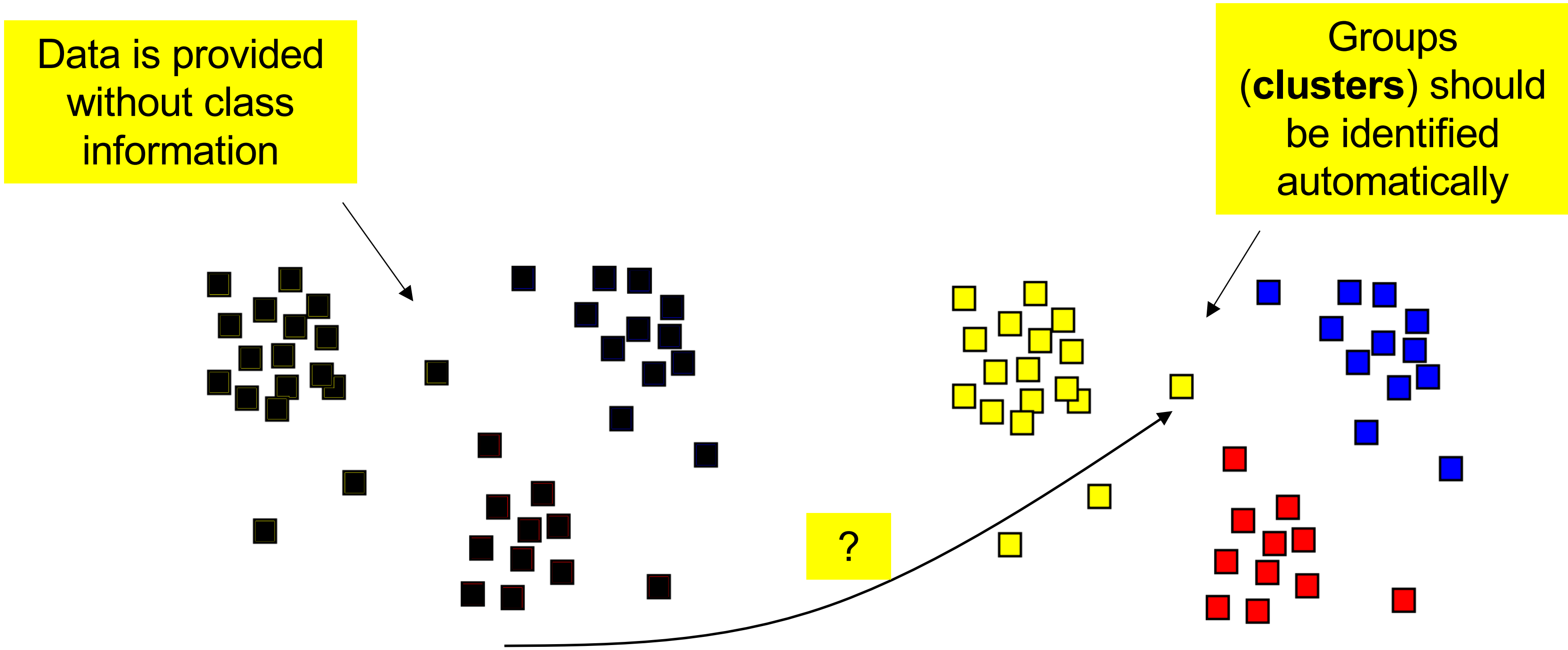
MACHINE LEARNING
MEI/1

University of Beira Interior, Department of Informatics

Hugo Pedro Proença, hugomcp@di.ubi.pt, 2018/2019

Unsupervised Learning

- This concept is associated to learning without a "**supervisor**"
  - It also known as self-organization, or cluster analysis
- The basic idea is that, instead of attempting to mimic the behavior of the supervisor, to identify commonalities in the data



- The notion of "cluster" cannot be objectively defined, which justifies different clustering algorithms.

Unsupervised Learning

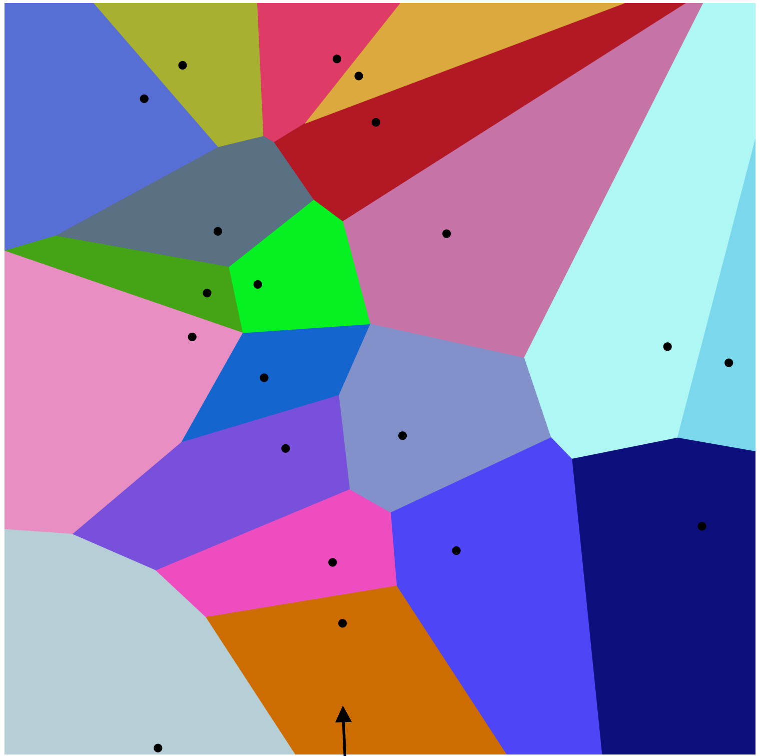
- There are different families of methods to perform clustering:
  - **Connectivity** models, in which models are built based on distance connectivity
    - Hierarchical clustering
  - **Centroid** models, that represent clusters by mean vectors (i.e., centroids)
    - K-means
  - **Distribution** models, where clusters are modelled according to statistical distributions
    - DBSCAN
  - **Neural** models, where networks implement a form of PCA that finds appropriate feature subspaces
    - Self-Organizing Map (SOM)
- **Clusters Evaluation**
  - **Internal** Evaluation, when the model is evaluated based on the data that was clustered itself
    - Davies-Bouldin index: $DB = \frac{1}{N}\sum_{i=1}^{N} \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d(ci, cj)}$

    where "c" represents one centroid, "σ" is the average distance of the elements in one cluster to its centroid and "N" is the number of clusters
  - **External** Evaluation, when the model is evaluated based on new data, typically with class labels
    - Purity: $P = \frac{1}{N}\sum_{i=1}^{M} \max_{d \in D} |m \cap d|$

    where "M" represents the set of clusters, and "D" is the labeled data

K-Means

- It is the most used clustering algorithm, due to its effectiveness and easiness of implementation.
  - Aims to partition **"n" observations** into **"k" clusters**
  - Each observation belongs to the nearest cluster centroid, which is the prototype of the cluster.
  - This results in a partitioning of the data space into Voronoi cells.
    - A Voronoi diagram is a partitioning of a plane into regions based on distance to points in a subset of the plane.
    - These points (a.k.a. prototypes) determine the shape of the corresponding Voronoi cell.
    - For each prototype there is a corresponding region consisting of all points closer to that seed than to any other. These regions are called Voronoi cells.



Positions in each cell (color) are closest to the corresponding centroid than to any other

K-Means

- For K-Means, the value of "K" must be given beforehand
  - There are diferente heuristics to automatically find the optimal value of "K", but depend of the specific problema considered
- Having a data set **X**: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$

1. Initialize (randomly) "K" centroids $\boldsymbol{\mu}$: $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k\}$
2. While (¬ stopping_criterium($\boldsymbol{\mu}$, **X**))
   1. For every $\boldsymbol{x}_i$:

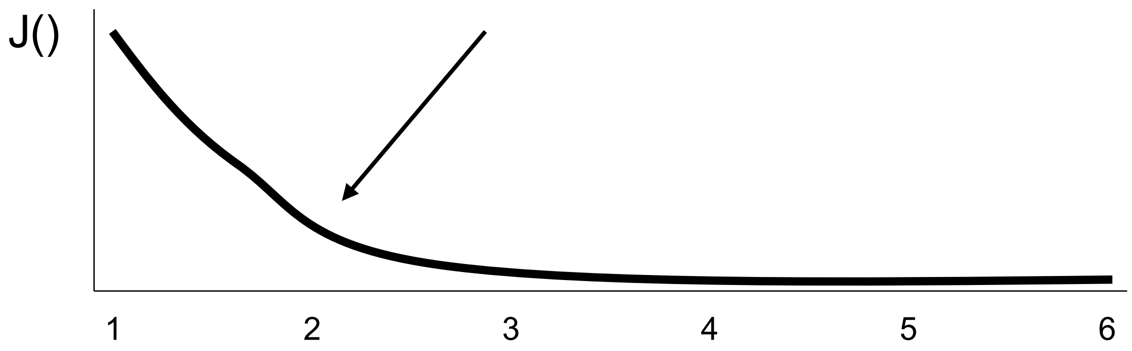
      $\mathbf{c}_i$ = arg min $_i$ d($\boldsymbol{x}_{i,}$ $\boldsymbol{\mu}_i$) //cluster assignment

   2. For every $\boldsymbol{\mu}_i$:

      $\boldsymbol{\mu}_i = \sum_j^n \boldsymbol{x}_j \mid \boldsymbol{x}_j$ assigned to $\mathbf{c}_i$ // centroid update

K-Means

- **Stopping criteria**. There are a number of diferent possibilities
  - Simplistic: Predefine a **number of iterations**
    - Might be "*too many*", or "*too few*", depending of the complexity of the feature space
  - Elaborate 1: Evaluate clusters **stationarity** and stop when the changes in clusters positions between consecutive iterations is less than a small threshold.
  - Elaborate 2: Evaluate samples **assignments** and stop when no samples (or a very small number) of samples changes its centroid between consecutive iterations.
- **Choose the value of "K"**
  - Elbow method.
  - Define a cost function J() and repeat the clustering procedure for a growing number of clusters. Define "K" as the value where the curvature of J() is maximal

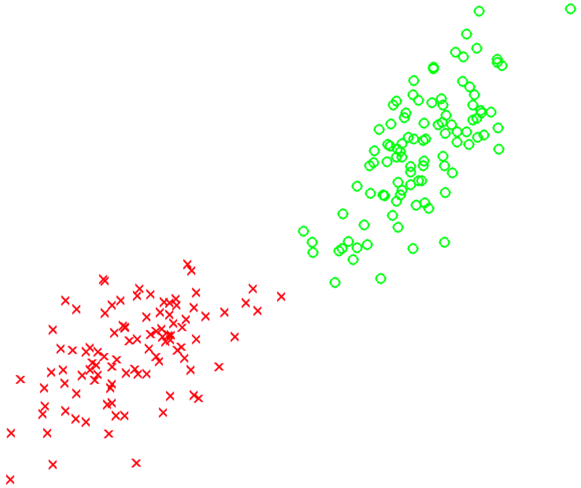


K-Means

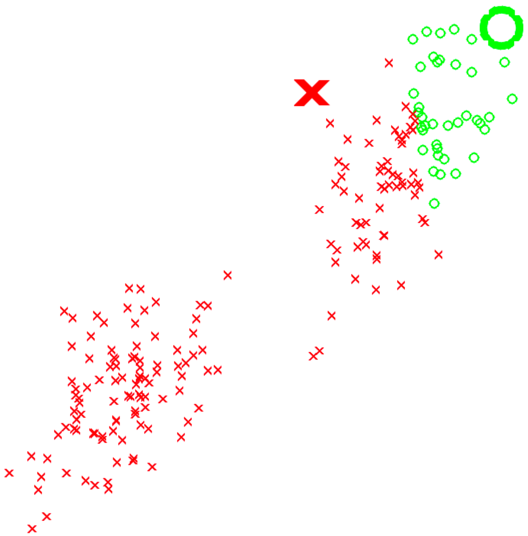
- **Distance Functions**
  - Different functions can be used, as long as they met the properties of being a "**metric**"
  - A metric on a set X is a function d : X × X → [0 , ∞), where for all x , y , z ∈ X, the following conditions are satisfied:
    - d(**x**, **y**) ≥ 0 // non-negativity or separation axiom
    - d(**x**, **y**) = 0 ⇔ **x** = **y** //identity of indiscernibles
    - d(**x**, **y**) = d(**y**, **x**) //symmetry
    - d(**x**, **z**) ≤ d(**x**, **y**) + d(**y**, **z**) //triangle inequality
  - Examples:
    - Euclidean distance: d(**x**,**y**) = $\sqrt{\sum(x_i - yi)^2}$
    - Manhatan distance: d(**x**,**y**) = $\sum(|xi - yi|)$
    - Chebyshev distance: d(**x**,**y**) = $\max|xi - yi|$

K-Means: Example

- Consider the following synthetic dataset:

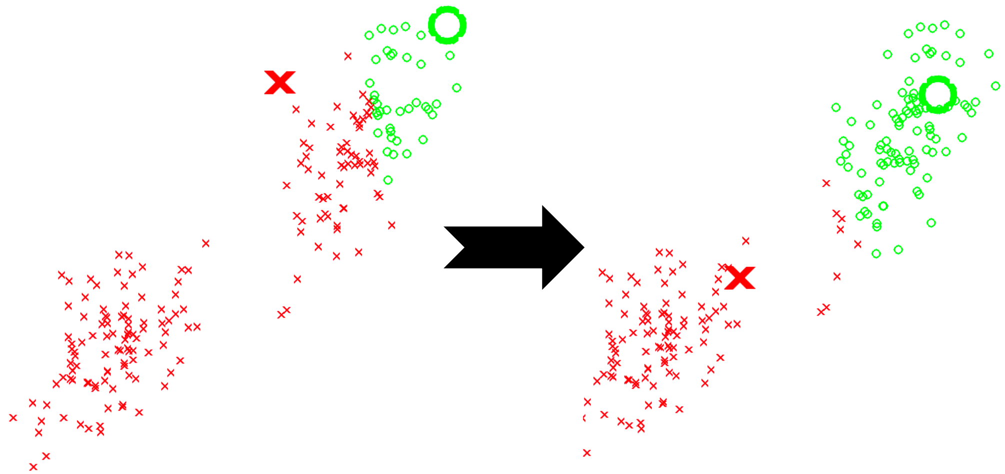


- Random initialization of 2 clusters:

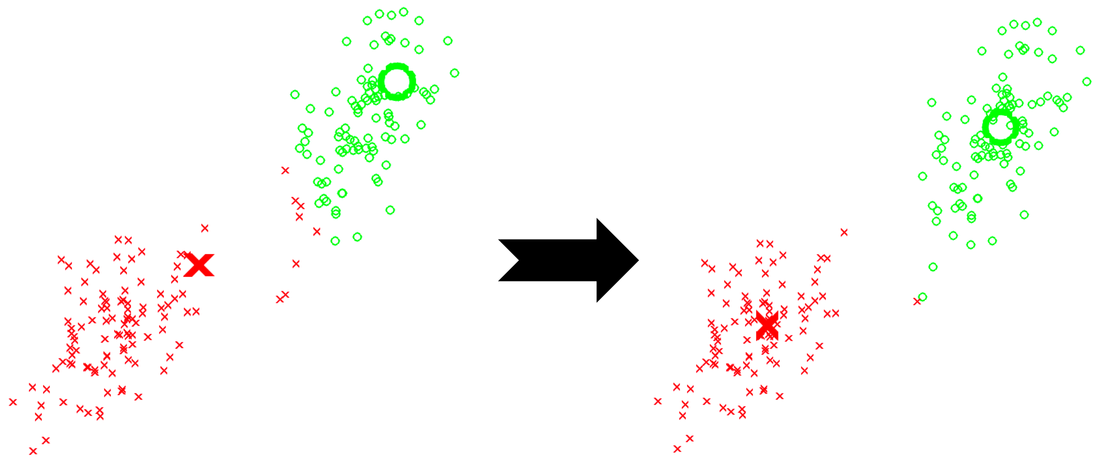


K-Means: Example

- K-Means: Iteration 1

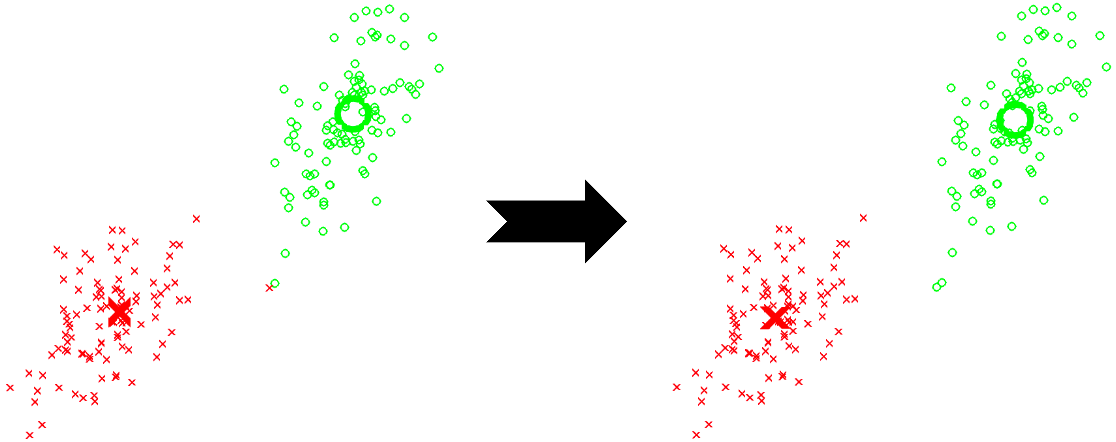


- K-Means: Iteration 2

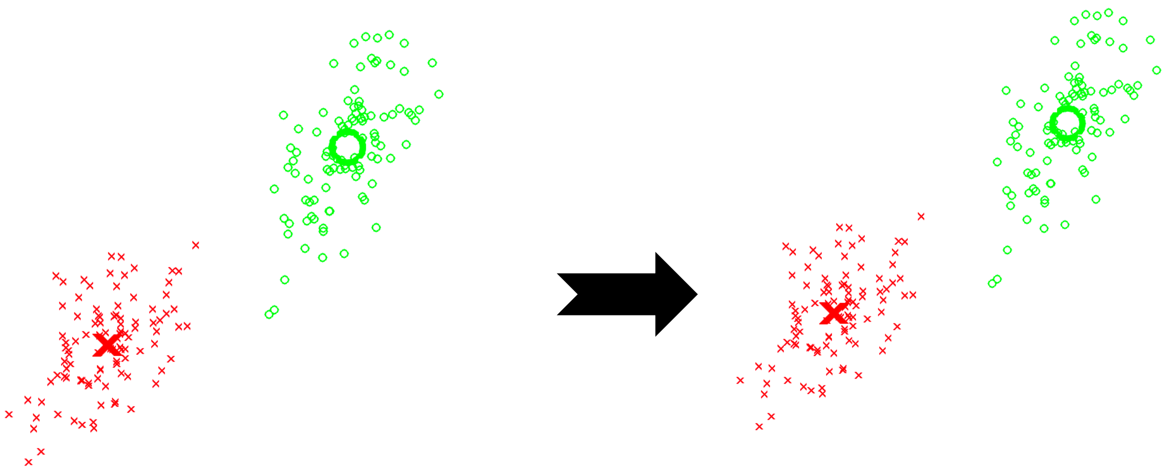


K-Means: Example

- K-Means: Iteration 3



- K-Means: Iteration 4



K-Means: Example

- K-Means: Iteration 5