Politecnico di Bari

Politecnico di Bari

Dipartimento di Ingegneria Elettrica e dell'Informazione

Corso di Laurea Triennale in Ingegneria dei Sistemi Medicali

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

Bioinformatics and Big Data Analytics

Cluster Analysis

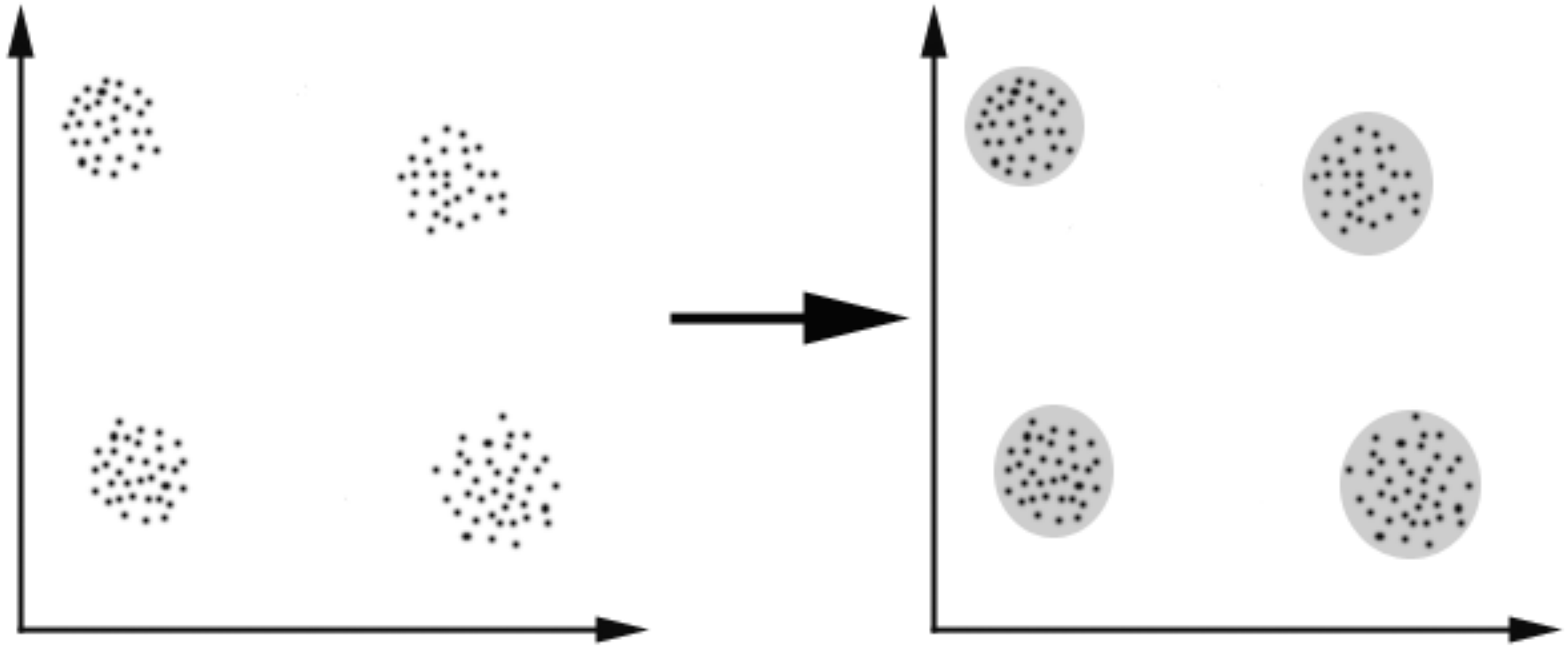
Eng. Nicola **Altini**, *Ph.D. Student*
Eng. Giacomo Donato **Cascarano**, *Ph.D. Student*
Prof. *Eng*. Vitoantonio **Bevilacqua**, *Ph.D.*

INDUSTRIAL INFORMATICS LAB
LABORATORIO DI INFORMATICA INDUSTRIALE

Anno Accademico 2019/2020

apulian bioengineering company

Cluster Analysis



Cluster Analysis

- Cluster analysis methods split the set of patterns into subsets (clusters) based on the mutual similarity of subset elements. Objects with dissimilar characteristics lie in different clusters.
- Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their understanding of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances between cluster members, dense areas of the data space, intervals or particular statistical distributions.

Cluster Analysis

- Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including parameters such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results.
- Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

Taxonomy

- A possible taxonomy of the clustering methods is the following:
  - **Hierarchical methods**. These methods involve the construction of a clustering tree. The set of patterns is separated into the two most unrelated subsets, and subsets are recursively separated into smaller subsets.
  - **Non-hierarchical methods**. These methods sequentially assign each pattern to one cluster. They can be either parametric or non-parametric.
    - **Parametric** approaches are based on known class-conditioned distributions and involve distribution parameter estimation.
    - **Non-parametric** methods are simpler and practically useful.

K-means

- Among **non-hierarchical non-parametric** methods there is the K-means algorithm, which is an elementary but very popular method.
- The input of the algorithm is an N n-dimensional data points. We assume that the number of clusters K is known.
- We initialize the location of cluster exemplars using random values or exploiting known data structure if available. We iteratively assign data points to their closest exemplar, and then we recalculate the exemplars as the centroid of their associated data points. When the exemplars' position stabilize the algorithm terminates.

K-means

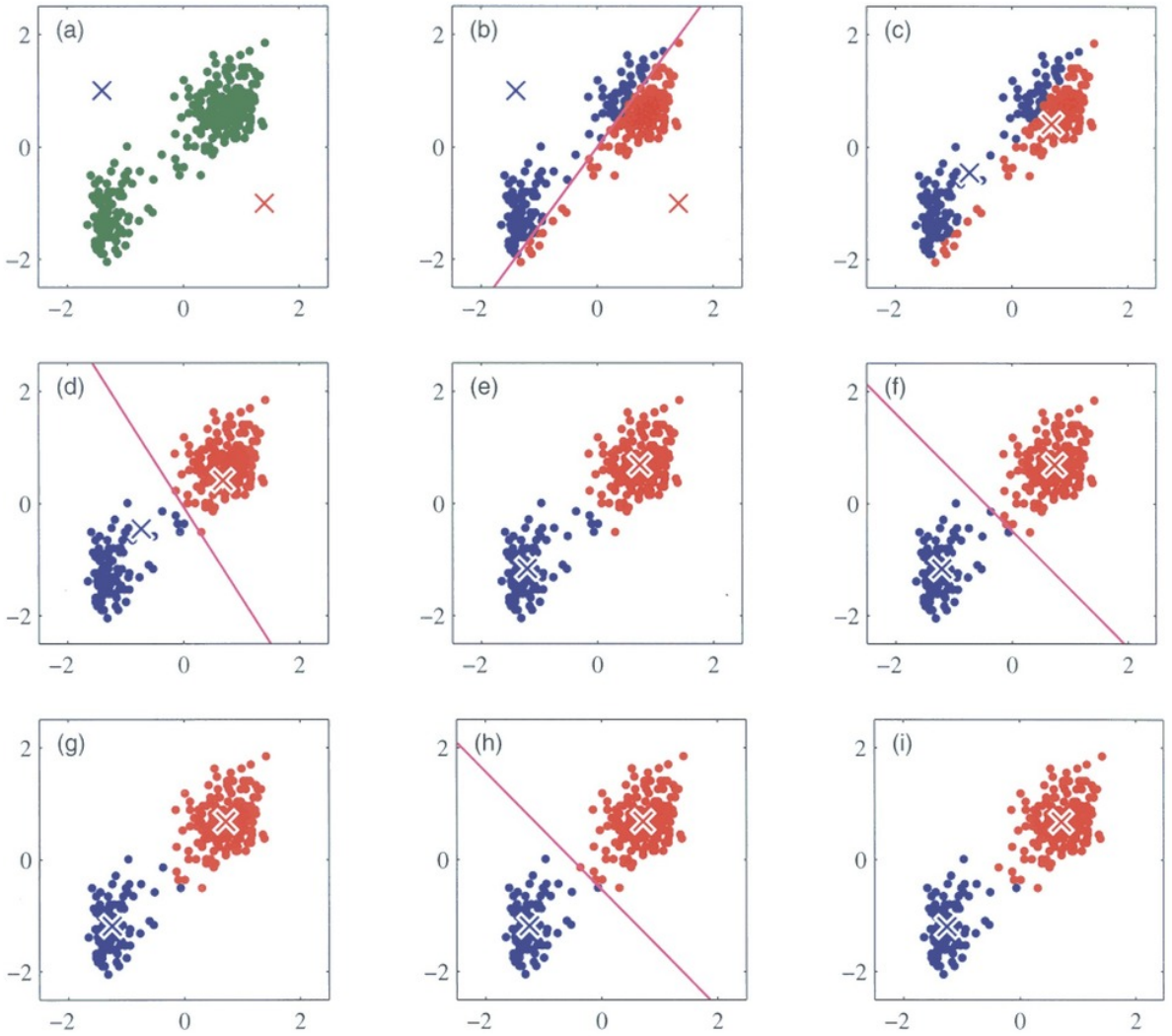
K-means cluster analysis.

1. Define the number of clusters K to be sought in a set $\mathrm{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$, with $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N \in \mathbb{R}^n$.
2. Initialize K cluster exemplars $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K \in \mathbb{R}^n$. This may be random, or from K data points, or from other prior knowledge.
3. Allocate data points to the closest $\boldsymbol{\mu}_i$ using some distance metric d (Euclidean distance is a common choice).
4. Recompute the $\boldsymbol{\mu}_i$ as centroids of their associated data, M_i

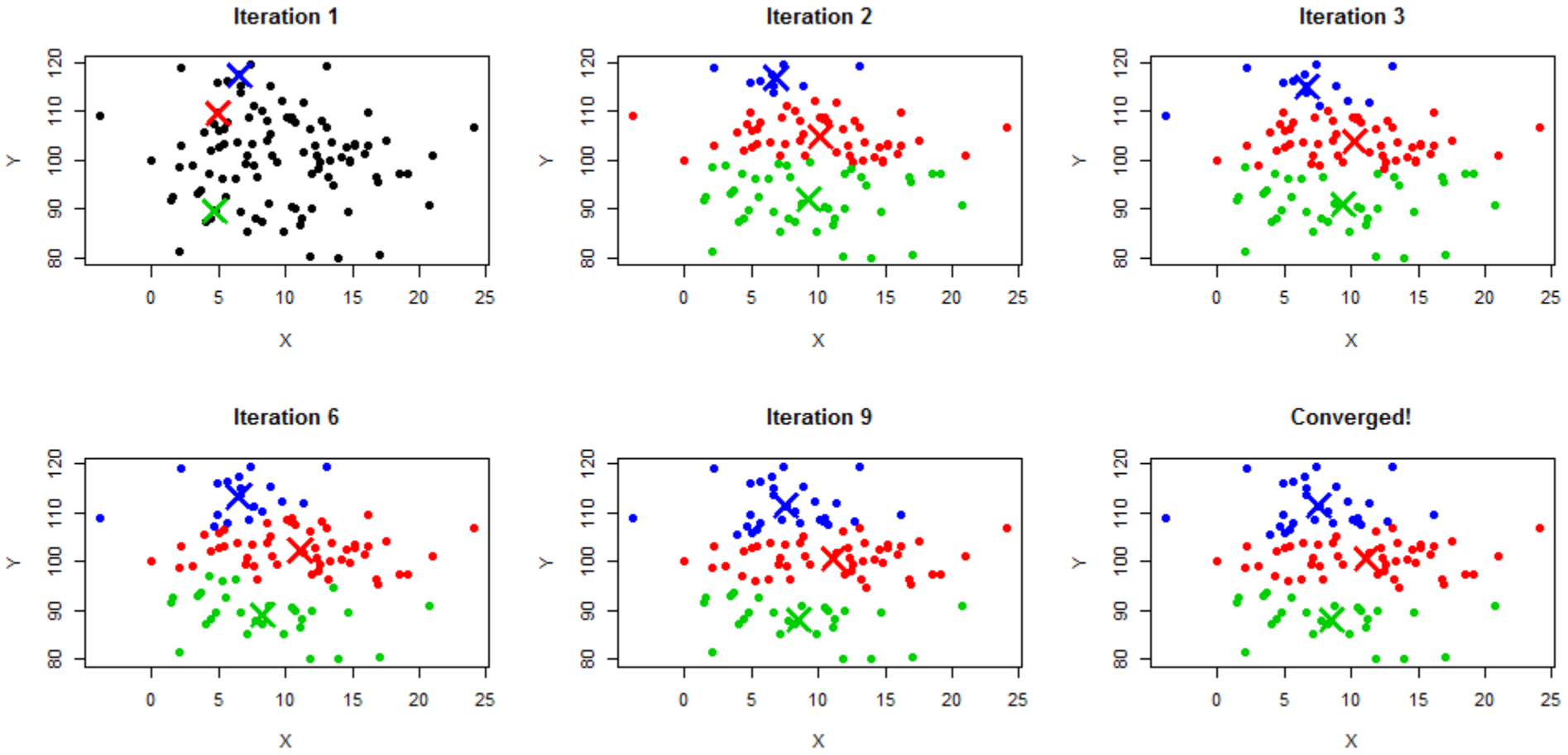
$$M_i = \left\{\boldsymbol{x}_j : d\left(\boldsymbol{x}_j, \boldsymbol{\mu}_i\right) = \min_k \left(d\left(\boldsymbol{x}_j, \boldsymbol{\mu}_k\right)\right)\right\}$$

5. If the $\boldsymbol{\mu}_i$ have not stabilized, go to 3.

K-means



K-means

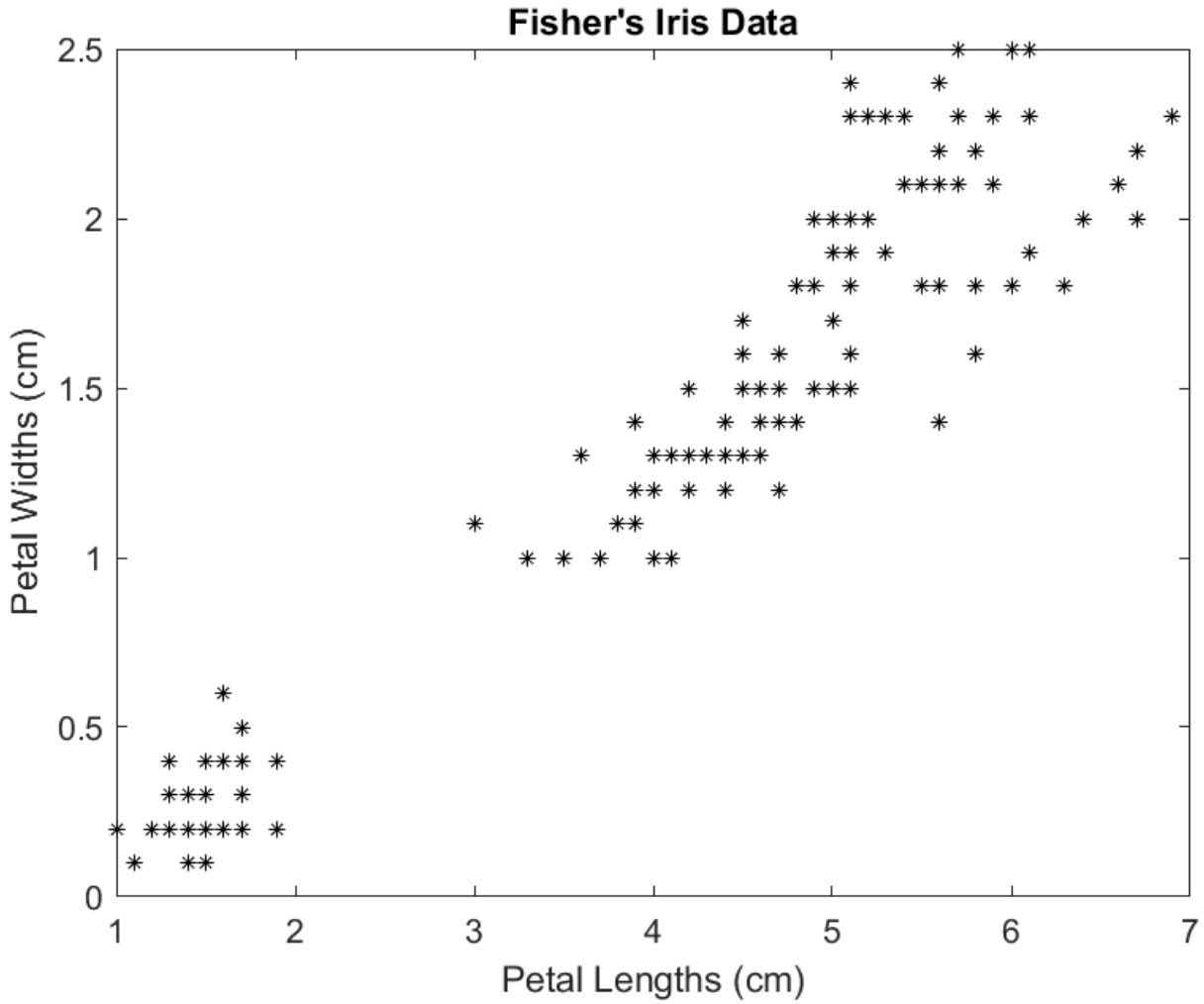

Iteration 1
Iteration 2
Iteration 3
Iteration 6
Iteration 9
Converged!
X
Y
0
5
10
15
20
25
80
90
100
110
120

K-means in MATLAB

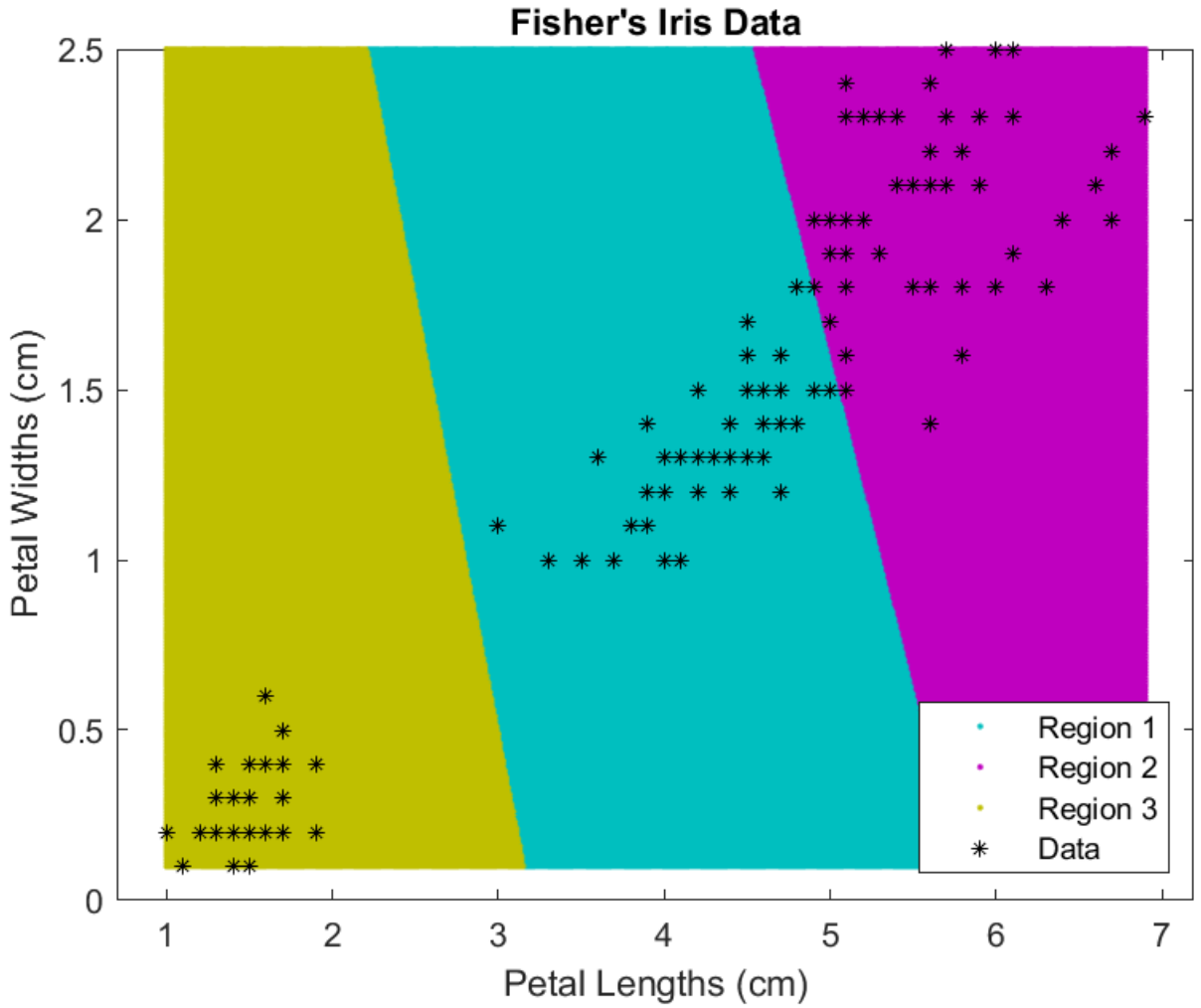
- `idx = kmeans(X,k)` performs k-means clustering to partition the observations of the n-by-p data matrix `X` into `k` clusters, and returns an n-by-1 vector (`idx`) containing cluster indices of each observation. Rows of `X` correspond to points and columns correspond to variables.

- By default, `kmeans` uses the squared Euclidean distance metric and the k-means++ algorithm for cluster center initialization.

K-means in MATLAB



K-means in MATLAB



Self-Organizing Maps

- A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction.
- Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space.

Self-Organizing Maps

- This makes SOMs useful for visualization by creating low-dimensional views of high-dimensional data, akin to multidimensional scaling. The artificial neural network introduced by the Finnish professor Teuvo Kohonen in the 1980s is sometimes called a Kohonen map or network.
- The Kohonen net is a computationally convenient abstraction building on biological models of neural systems from the 1970s and morphogenesis models dating back to Alan Turing in the 1950s.

Self-Organizing Maps

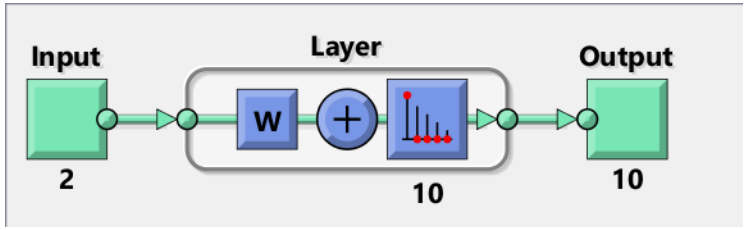
- Like most artificial neural networks, SOMs operate in two modes: training and mapping. "Training" builds the map using input examples (a competitive process, also called vector quantization), while "mapping" automatically classifies a new input vector.
- The visible part of a self-organizing map is the map space, which consists of components called nodes or neurons. The map space is defined beforehand, usually as a finite two-dimensional region where nodes are arranged in a regular hexagonal or rectangular grid.

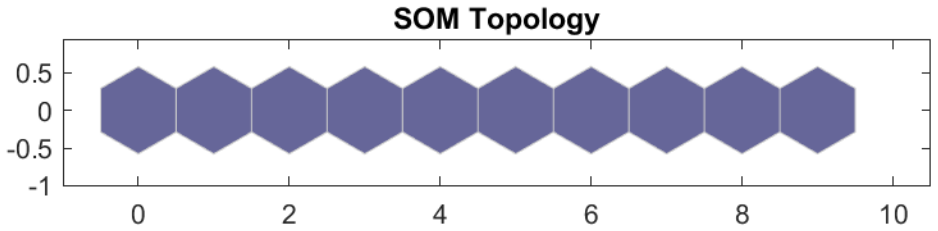
Self-Organizing Maps

- Each node is associated with a "weight" vector, which is a position in the input space; that is, it has the same dimension as each input vector. While nodes in the map space stay fixed, training consists in moving weight vectors toward the input data (reducing a distance metric) without spoiling the topology induced from the map space.
- Thus, the self-organizing map describes a mapping from a higher-dimensional input space to a lower-dimensional map space. Once trained, the map can classify a vector from the input space by finding the node with the closest (smallest distance metric) weight vector to the input space vector.

Self-Organizing Maps in MATLAB

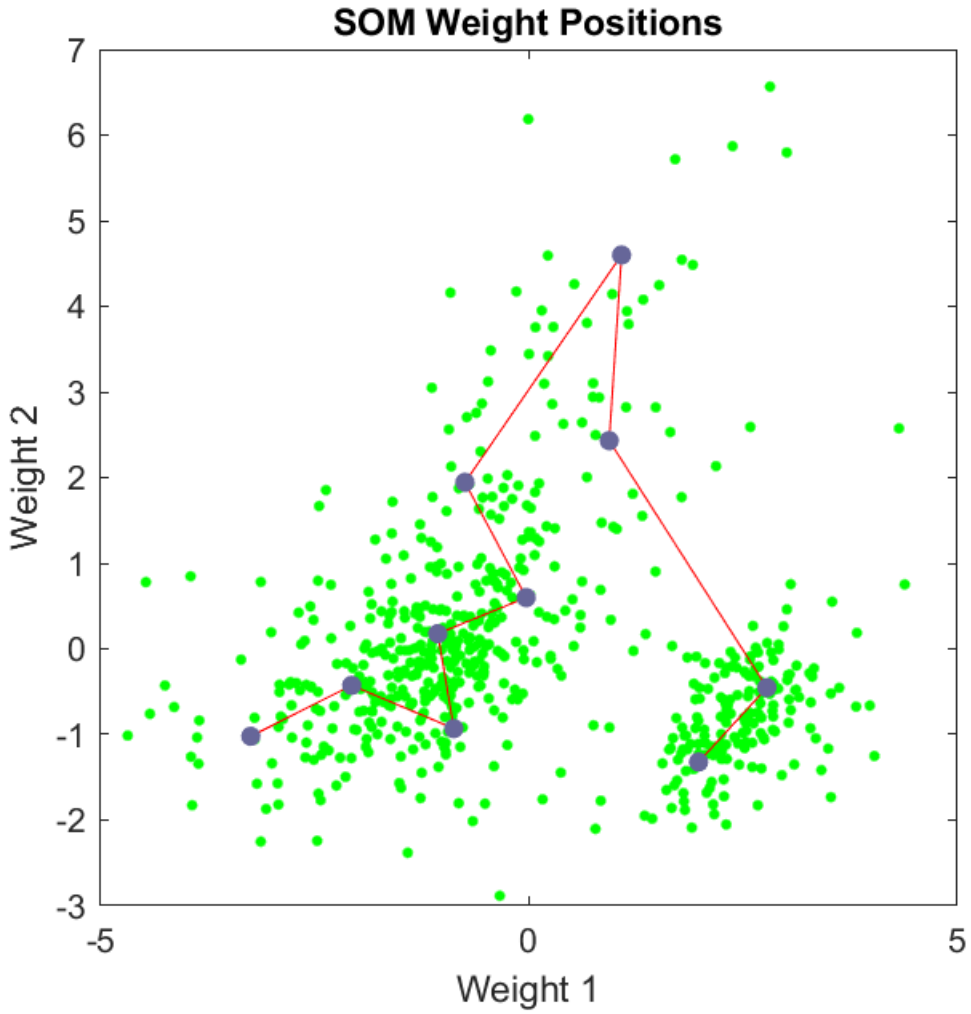
```
dimensions = [10];
% Creazione rete
net = selforgmap(dimensions);
```





Self-Organizing Maps in MATLAB

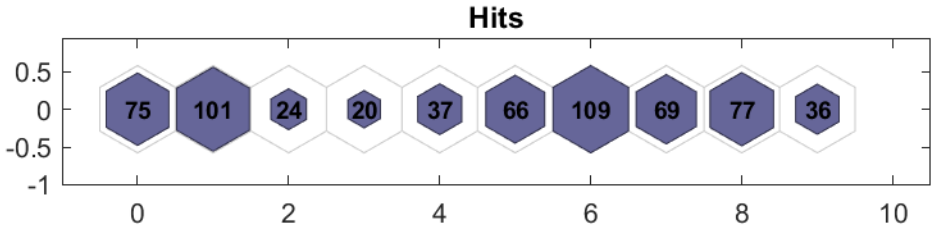
```
figure,plotsompos(net,x);
```



Self-Organizing Maps in MATLAB

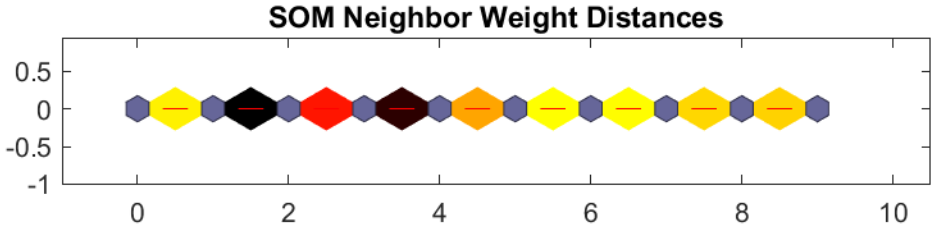
- **Hits**

```
figure,plotsomhits(net,x);
```



- **SOM Neighbor Weight Distances**

```
figure,plotsomnd(net,x);
```



Hierarchical Clustering

- **Hierarchical clustering** (also called *hierarchical cluster analysis* or *HCA*) is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:
  - **Agglomerative**: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
  - **Divisive**: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.
- In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a dendrogram.

Cluster dissimilarity

- In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required.
- In most methods of hierarchical clustering, this is achieved by use of an appropriate **metric** (a measure of distance between pairs of observations), and a **linkage criterion** which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.
- The choice of an appropriate **metric** will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another.
- The **linkage criterion** determines the distance between sets of observations as a function of the pairwise distances between observations.

Metrics

Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Maximum distance	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the Covariance matrix

Linkage criteria

Names	Formula
Maximum or complete-linkage clustering	$\max \{ d(a,b) : a \in A,\, b \in B \}.$
Minimum or single-linkage clustering	$\min \{ d(a,b) : a \in A,\, b \in B \}.$
Unweighted average linkage clustering (or UPGMA)	$\frac{1}{\lvert A \rvert \cdot \lvert B \rvert} \sum_{a \in A} \sum_{b \in B} d(a,b).$
Weighted average linkage clustering (or WPGMA)	$d(i \cup j, k) = \frac{d(i,k) + d(j,k)}{2}.$
Centroid linkage clustering, or UPGMC	$\lVert c_s - c_t \rVert$ where c_s and c_t are the centroids of clusters s and t, respectively.
Minimum energy clustering	$\frac{2}{nm} \sum_{i,j=1}^{n,m} \lVert a_i - b_j \rVert_2 - \frac{1}{n^2} \sum_{i,j=1}^{n} \lVert a_i - a_j \rVert_2 - \frac{1}{m^2} \sum_{i,j=1}^{m} \lVert b_i - b_j \rVert_2$

Hierarchical Clustering in MATLAB

- `Z = linkage(X)` returns a matrix `Z` that encodes a tree containing hierarchical clusters of the rows of the input data matrix `X`.

- `Z = linkage(X,method)` creates the tree using the specified method, which describes how to measure the distance between clusters.

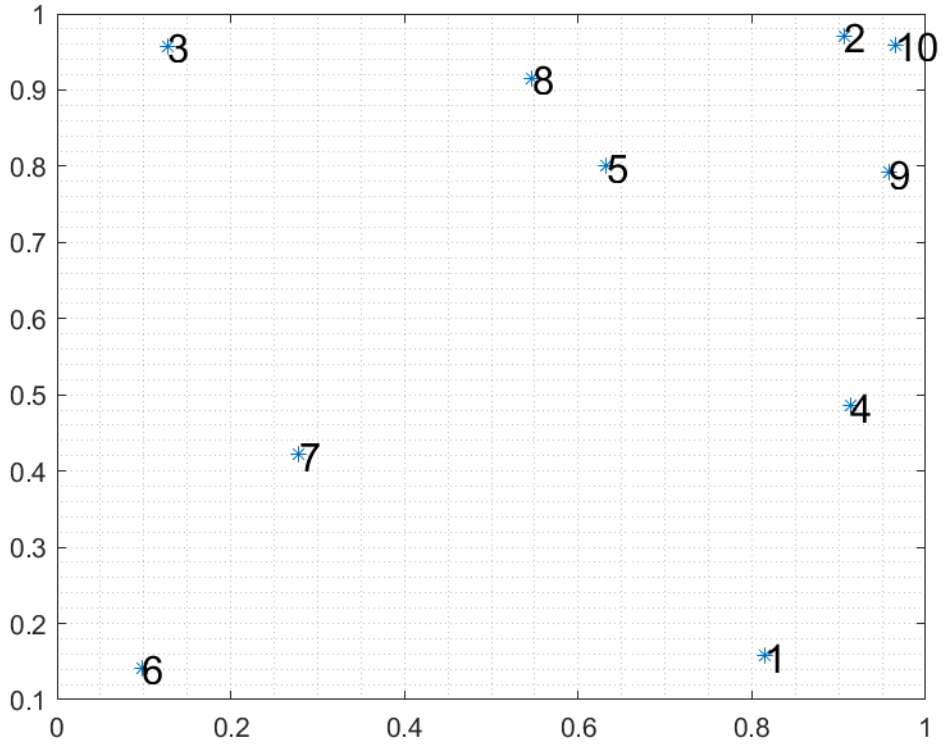
- `Z = linkage(X,method,metric)` performs clustering by passing metric to the pdist function, which computes the distance between the rows of `X`.

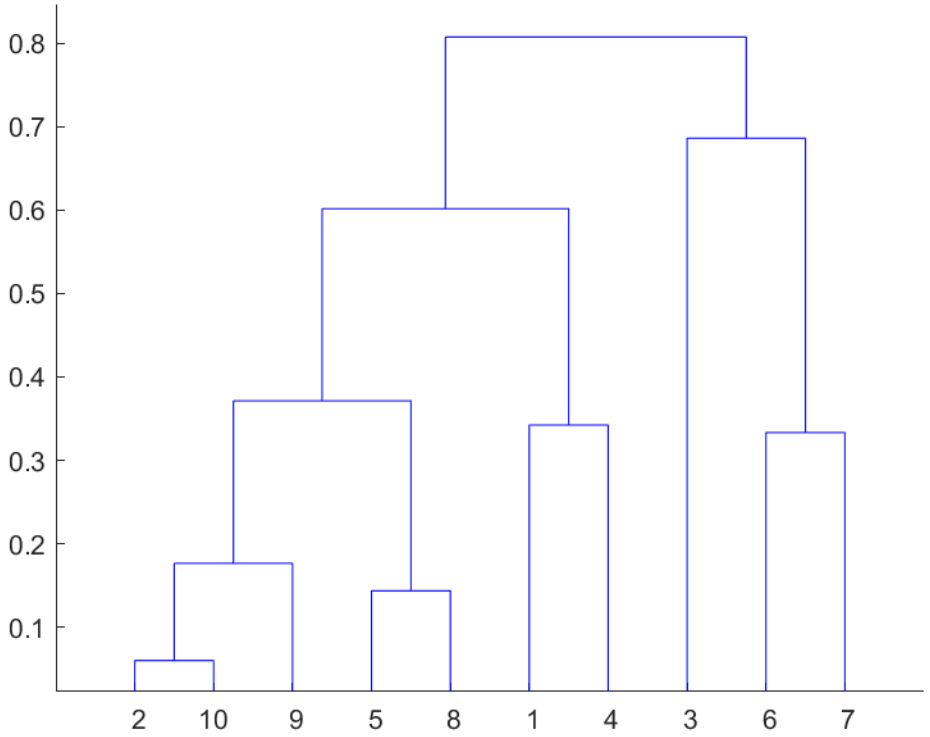
Dendrogram

- A dendrogram is a diagram representing a tree. This diagrammatic representation is frequently used in different contexts:
  - in **hierarchical clustering**, it illustrates the arrangement of the clusters produced by the corresponding analyses.
  - in **computational biology**, it shows the clustering of genes or samples, sometimes in the margins of heatmaps.
  - in **phylogenetics**, it displays the evolutionary relationships among various biological taxa. In this case, the dendrogram is also called a phylogenetic tree.

Dendrogram in MATLAB

```
tree = linkage(X,'average','euclidean'); % Hierarchical Clustering
figure,dendrogram(tree) % Dendrogram plot
```





References

- MATLAB Documentation
  - K-means clustering.
    URL: https://www.mathworks.com/help/stats/kmeans.html
  - Self-Organizing Maps.
    URL: https://www.mathworks.com/help/deeplearning/gs/cluster-data-with-a-self-organizing-map.html
  - Agglomerative hierarchical cluster tree
    URL: https://www.mathworks.com/help/stats/linkage.html
  - Dendrogram
    URL: https://www.mathworks.com/help/stats/dendrogram.html
- Wikipedia
  - Self-Organizing Maps.
    URL: https://en.wikipedia.org/wiki/Self-organizing_map