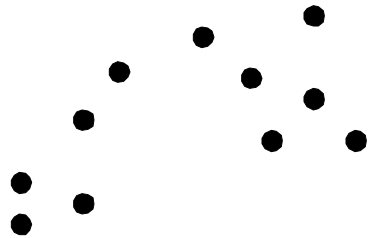
Data Mining
Cluster Analysis

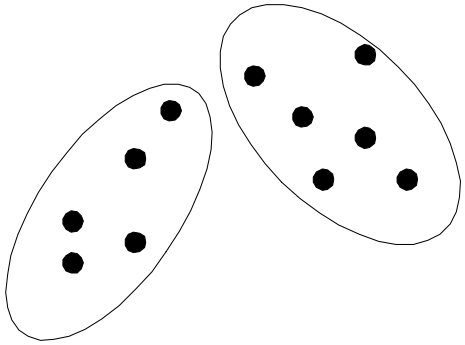
Lecture Notes for Chapter 8

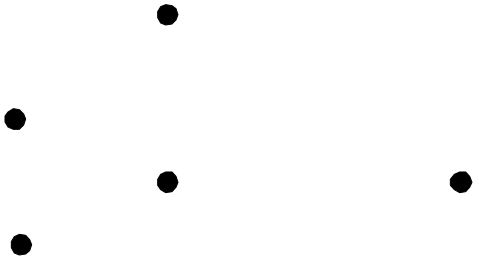
Clustering

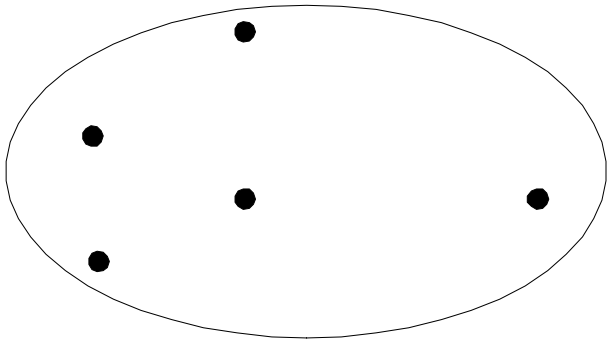
- **Target:** Divide data into a set of groups (**clusters**) based on **similarity**
- **Similar** samples are grouped together, while **dissimilar** samples are placed in different clusters
- Input dataset: **unlabeled data** $X = \{x_1, x_2, \ldots, x_N\}$
  Available only the information of feature values
- **Unsupervised learning**

Examples (I)





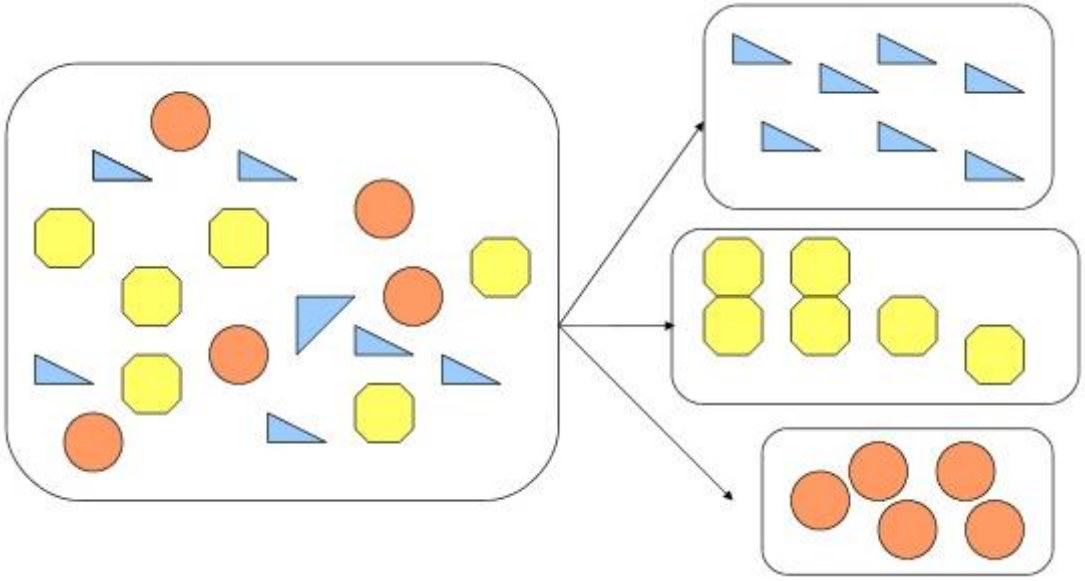


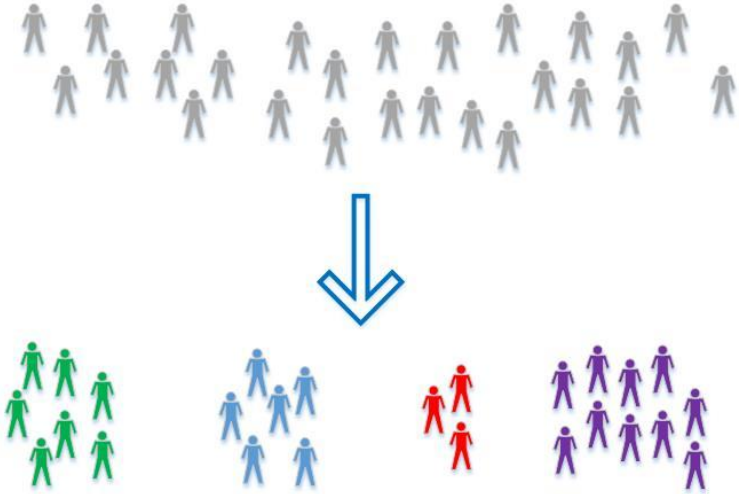


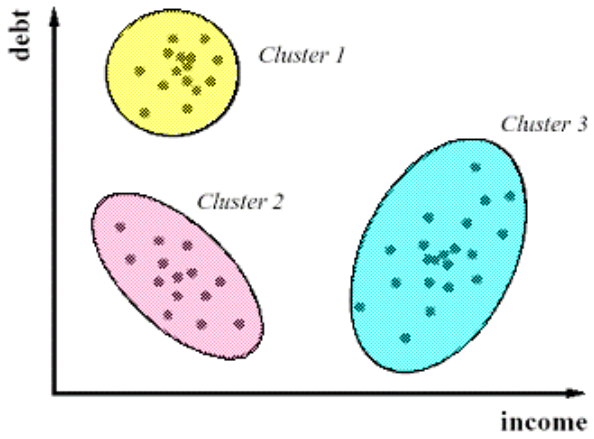
Input dataset

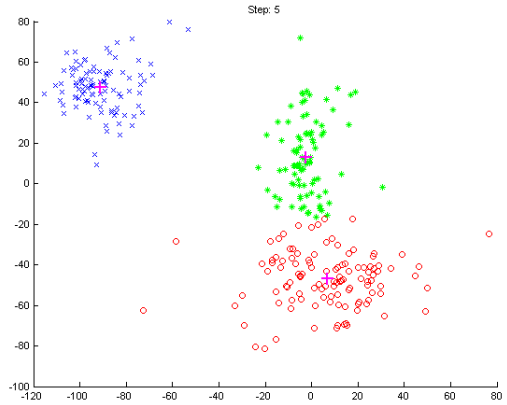
Clustering result

Examples (II)






debt
Cluster 1
Cluster 3
Cluster 2
income


Step: 5
80
60
40
20
0
-20
-40
-60
-80
-100
-120
-100
-80
-60
-40
-20
0
20
40
60
80

Advantages of Clustering

- Clustering for **data understanding**
- Discover **dynamically** categories of data
- Clustering assists **Information retrieval**
  - efficient finding nearest neighbors
- **Summarization** of data
- **Compression** (vector quantization)

Cluster Analysis

Goal:

- Objects that belong to the same cluster are more similar to each other, and **simultaneously** differ from rest objects that belong to other clusters.

- The more similar among members of same cluster (**intercluster**), the more difference among clusters (**intracluster**).

- There is an **objective difficulty**

Cluster Analysis (cont.)

Which is the optimum number of clusters?

Solution with 2 clusters

Solution with 4 clusters

Solution with 6 clusters

Clustering approaches

- **Partitioning** (model-free) methods:
  - Divide the input data set into non-overlapping subsets - groups (clusters). Any point belongs exclusively to a single cluster.
- **Hierarchical** methods:
  - Build a hierarchy of clusters organized in a tree structure
- **Similarity-based** methods:
  - Use a similarity matrix of data and make a spectral analysis of it (graph-based clustering).
- **Model-based** methods:
  - Every cluster is described by a parametric model. During learning model parameters are estimated in order to fit the data. Any point belongs to several clusters with different degrees.

Exclusive vs. Non-exclusive (Overlapping) Clustering

- **Exclusive**
  - Any point belongs exclusively to a single cluster.
- **Overlapping**
  - Any point may belong to several clusters with different degree.
  - e.g. probabilistic clusters (probability of belongingness)
  - Fuzzy clusters (membership value)

Complete vs. Partial Clustering

- **Complete**
  - Clustering is performed to all data
- **Partial**
  - Some examples may not participate to clustering procedure, either
  - because they do not belong to well shaped clusters, or
  - because they are noisy data or outliers and may negatively affect clustering

Types of Clusters

- **Well-separated clusters**
  - Any point is more similar to all points of the same cluster in comparison with points from other clusters

- **Prototype-based or Center-based clusters**
  - The distance of any point with the cluster center it belongs is less than distances with other clusters' centers
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster

Types of Clusters (cont.)

- **Geometric clusters**
  - Clusters have geometric properties, i.e. have geometric rules to identify which data point belong to them. For example, hyperplanes or hyperspheres that surround a cluster region.

- **Graph-based clusters**
  - Graph representation of data where data points are vertices which communicate only with other points (vertices) of the same cluster. *Clusters as cliques*.

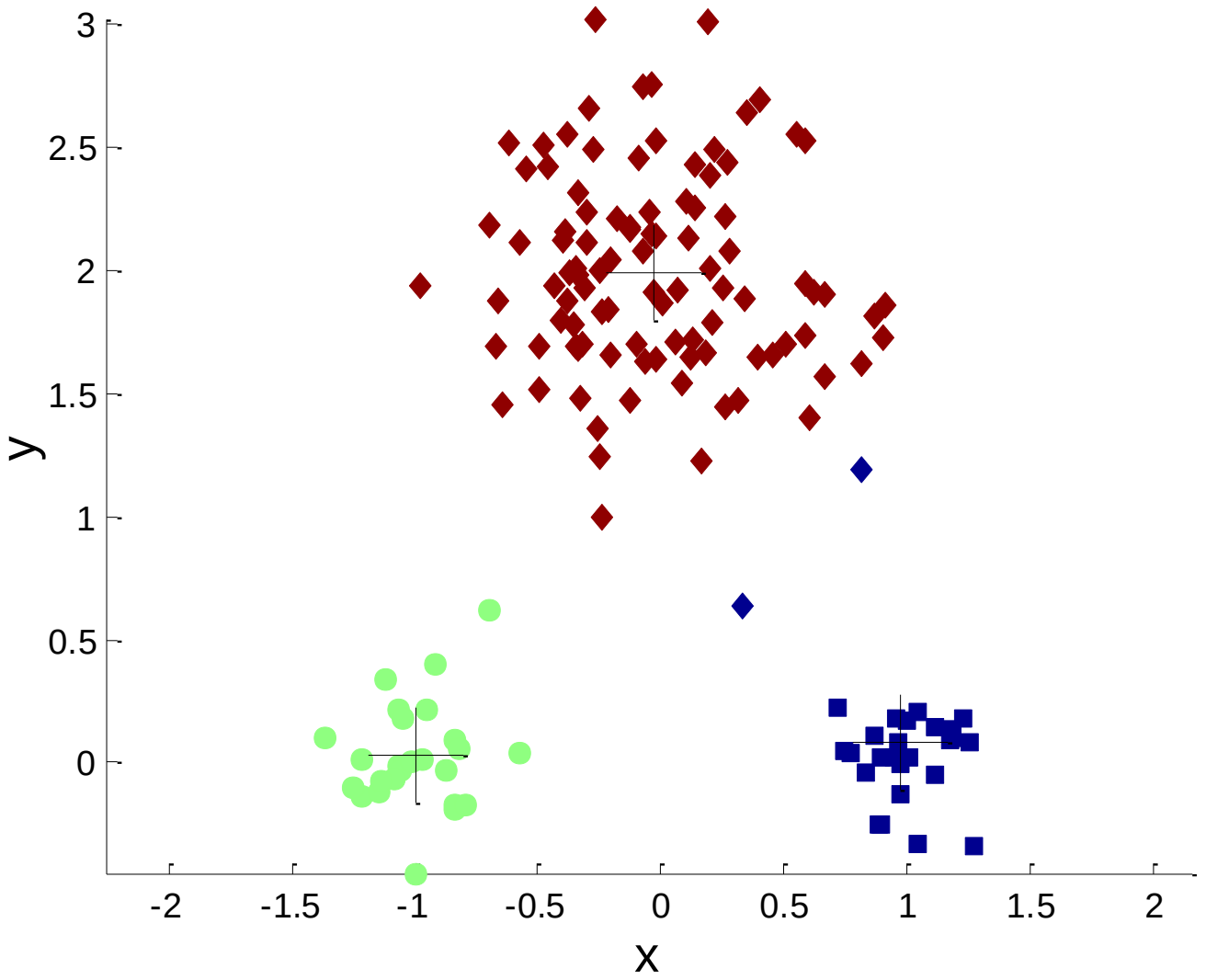
Types of Clusters (cont.)

- **Density-based clusters**
  - Cluster is a region of high density that surrounds similar points and is separated with other clusters with regions of minimum variance
- **Property or Conceptual clusters**
  - Cluster is a set of data sharing a common property (e.g. distance, geometry)

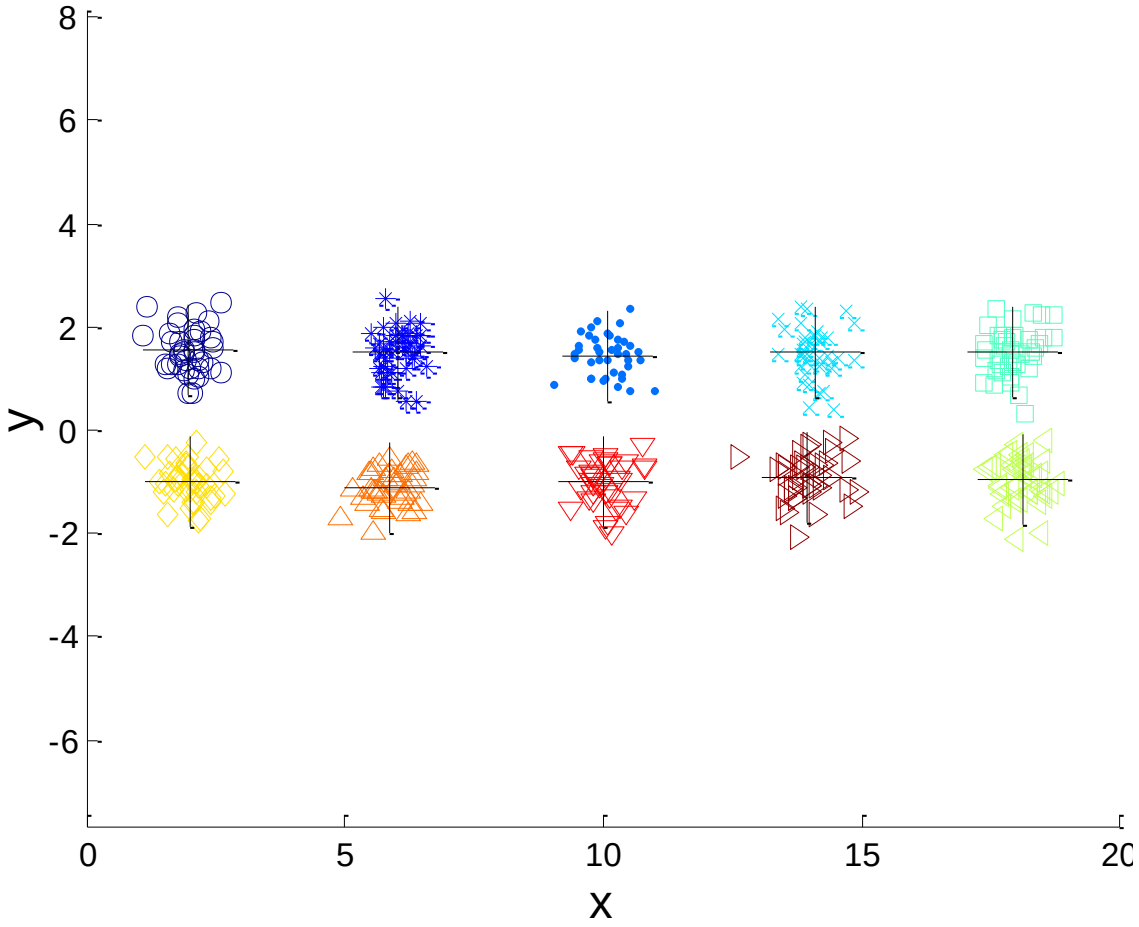
[1]. Partitioning Clustering: finding cluster representatives

- Every cluster Ω_j is described with a representative ($\boldsymbol{\mu_j}$) that describes it uniquely.
- **Summarization of data**
  - Representative summarizes all cluster members
  - Reducing dataset to a set of representatives of clusters
- **Compression** of information
  - Vector quantization
  - Useful in text, images, sounds and video (text, video or sound summarization by keeping most characteristic topics, paragraphs, or scenes)

3 cluster representatives



10 cluster representatives



Partitioning method

- **Decision mechanism**:

  A pattern x belongs **<u>exclusively</u>** to the **closest** cluster j^* that has the minimum distance (or the highest similarity) with its representative

$$x \in j^* = \arg\min_{j=1,\ldots,K} \left\{ d\left(x, \mu_j\right) \right\}$$

- **Learning goal**:

  Estimate the proper values of the representatives $\{ \mu_j \}$ given an input set of examples.

Ο αλγόριθμος K-means (MacQueen, 1967)

- Input dataset $X = \{x_1, x_2, \ldots, x_N\}$
- **Goal**: division of set *X* into ***K* clusters** and discovery ***K* representatives** { μ_j }. (*K: known*)
- **Cluster representatives**: Means or center of data belong to the same cluster.
- **Rule**: Every point belongs to the cluster with the minimum distance of its center.
- **Objective function**: $E = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \{d(x_i, \mu_j)\}$

Ο αλγόριθμος K-means (MacQueen, 1967)

- Αρχικοποίηση (***t=0***) των K μέσων: $\left\{\mu_j^{(0)}\right\} \quad j=1,\ldots,K$

$$E^{(0)} = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j^{(0)}\right) \right\}$$

- Επαναληπτικά

1. **Τοποθέτηση** των N προτύπων σε K ομάδες ανάλογα με την απόστασή τους από τα τρέχοντα μέσα των ομάδων

$$\Omega_j = \{\ \} \quad \forall\ j=1,\ldots,K$$

$$\forall\ i=1,\ldots,N \quad q = \arg\min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j\right) \right\} \quad \Omega_q = \Omega_q \cup \{x_i\}$$

2. **Ενημέρωση** των K μέσων $\forall\ j=1,\ldots,K \quad \mu_{\mathrm{j}}^{(\mathrm{t}+1)} = \dfrac{1}{\left|\Omega_j\right|} \sum_{x_i \in \Omega_j} x_i$

$$E^{(t+1)} = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j^{(t+1)}\right) \right\}$$

Ο αλγόριθμος K-means (MacQueen, 1967)

Termination criterion:

- Cluster centers stop modified between successive steps

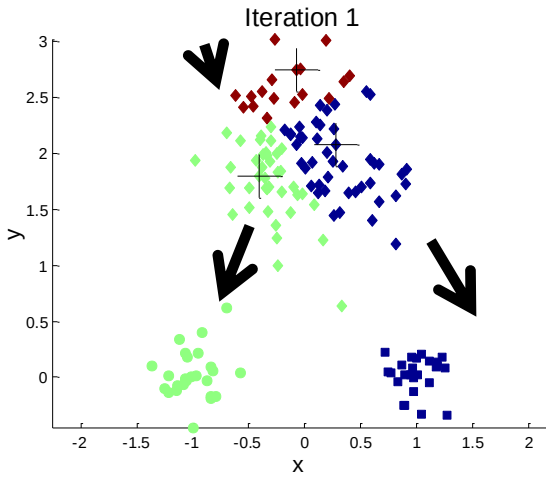
$$\text{STOP} \quad \text{if} \quad \sum_{j=1}^{K} \left\| \mu_j^{(t+1)} - \mu_j^{(t)} \right\|^2 \to 0$$

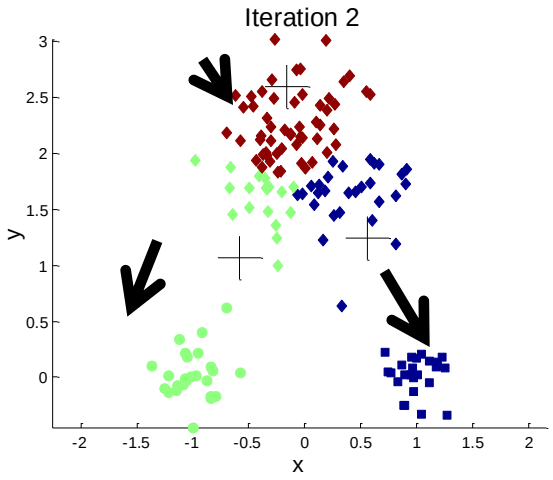
or

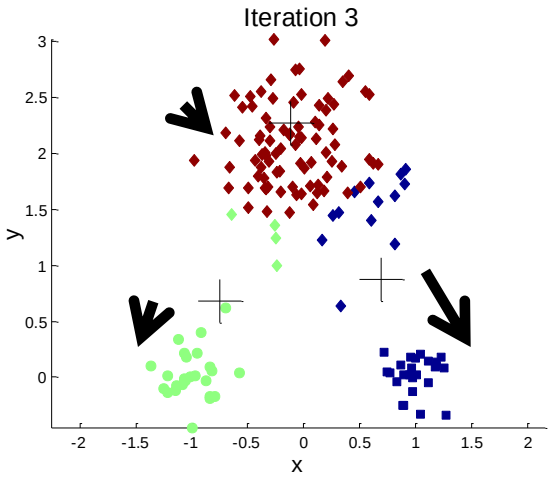
- Objective function stops modified between successive steps

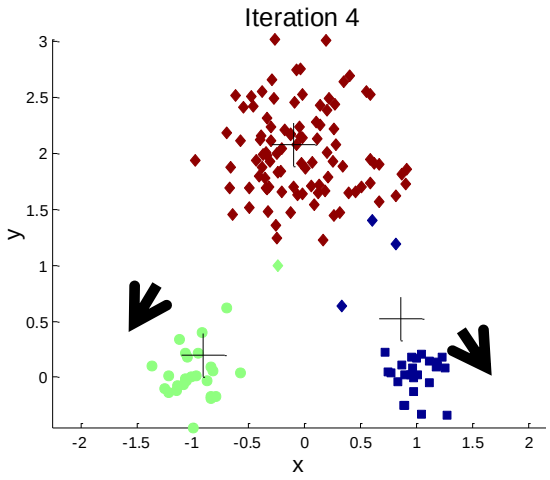
$$\text{STOP} \quad \text{if} \quad \Delta E = E^{(t)} - E^{(t+1)} < \varepsilon$$

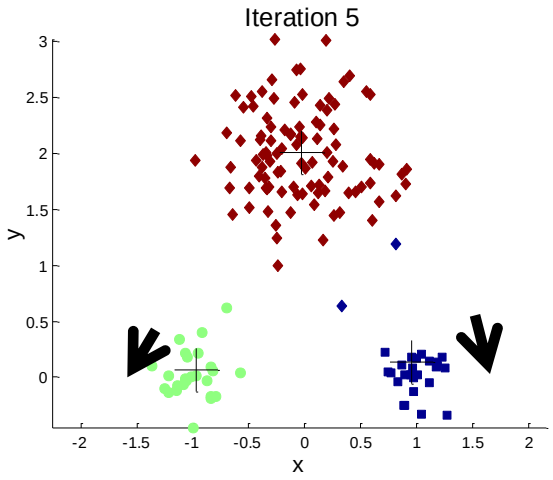
An example of execution of k-means algorithm

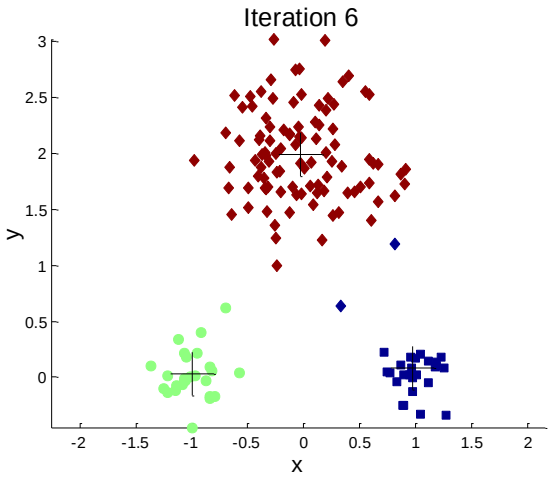












An alternative interpretation (I)

- Assuming Euclidean spaces:

$$E = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j\right) \right\} = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ \left\| x_i - \mu_j \right\|^2 \right\} =$$

$$= \sum_{j=1}^{K} \left\{ \sum_{x_i \in \Omega_j} \left\| x_i - \mu_j \right\|^2 \right\}_{\left\| x_i - \mu_j \right\|^2}$$

- **Target** of K-means is to **minimize the sample variance** of data belong to every cluster.
- Minimum variance clusters construction, or, **maximum coherence** clusters construction.

An alternative interpretation (II)

- Objective function:

$$E = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j\right) \right\} = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ \left\| x_i - \mu_j \right\|^2 \right\}$$

- Objective function as an **error function** of data to their closest center.

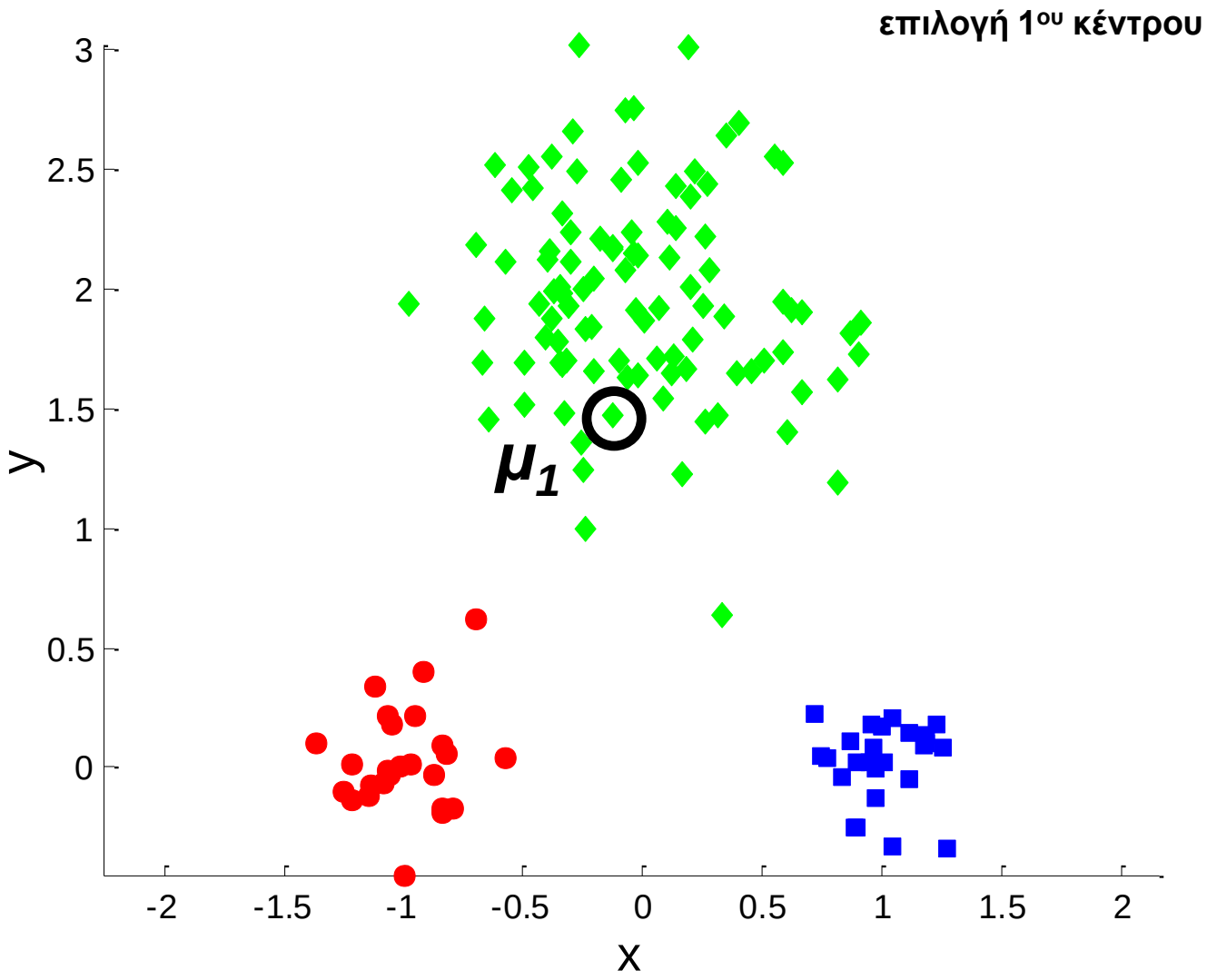
- During learning try to minimize the sum of squared error

$$\mu_j = \frac{1}{\left|\Omega_j\right|} \sum_{x_i \in \Omega_j} x_i$$

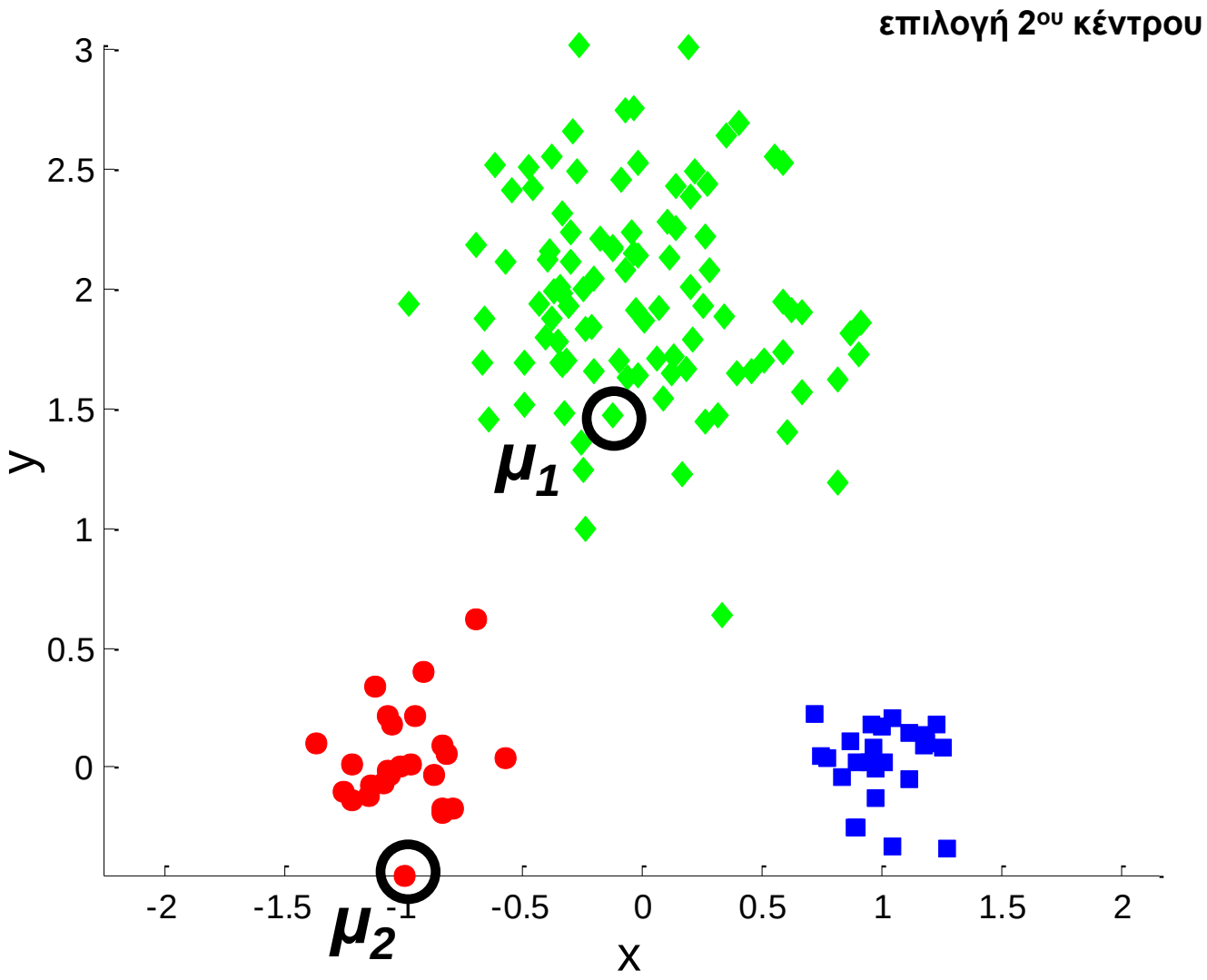
[1]. **Initialization strategies of cluster centers**

1. Συνήθως τυχαία επιλογή από τα δείγματα.
2. Ομοιόμορφα από το πεδίο τιμών των χαρακτηριστικών
3. Πολλές επαναλήψεις του K-means. Επιλογή της λύσης με την μικρότερη τιμή συνάρτησης (min{E}).
4. Με διαδοχική επιλογή κέντρων:
   - Επιλογή αρχικά ενός κέντρου τυχαία (j=1) ή συνολικό κέντρο
   - Επαναληπτικά επιλογή ως μέσο της j+1 ομάδας το πιο «απομακρυσμένο» σημείο του συνόλου δεδομένων από όλα τα μέσα { μ_j } που έχουν επιλεχθεί μέχρι το τρέχον βήμα.
   - Έτσι περισσότερο ευδιάκριτες ομάδες στο αρχικό βήμα
   - **Κίνδυνος** να επιλεγούν ως μέσα ακραίες τιμές (**outliers**)

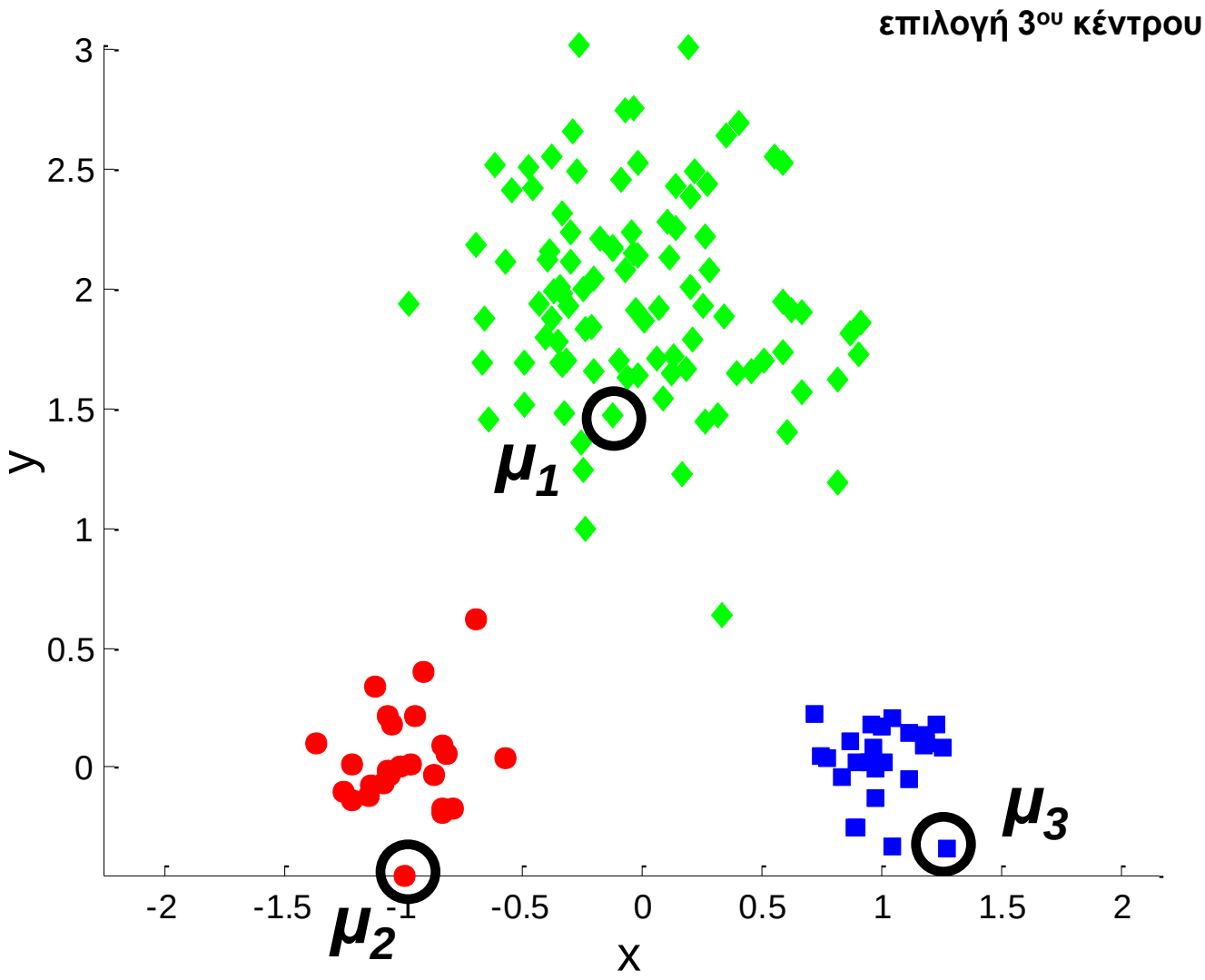
Παράδειγμα αρχικοποίησης



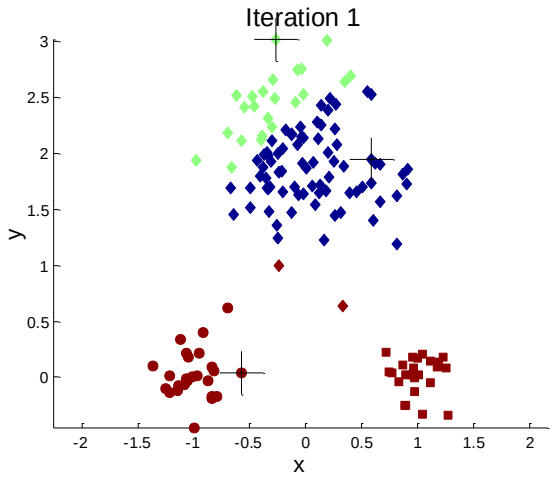
Παράδειγμα αρχικοποίησης

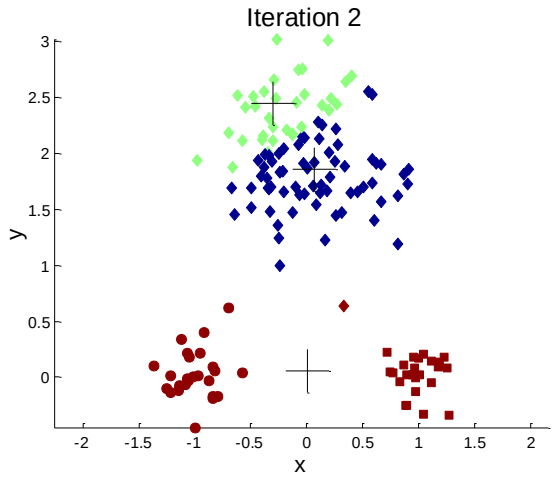


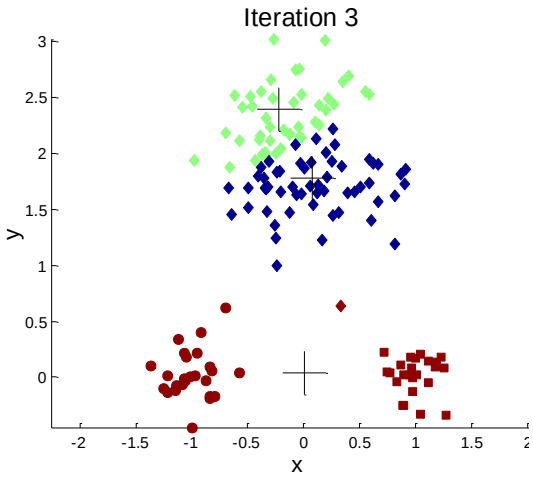
Παράδειγμα αρχικοποίησης

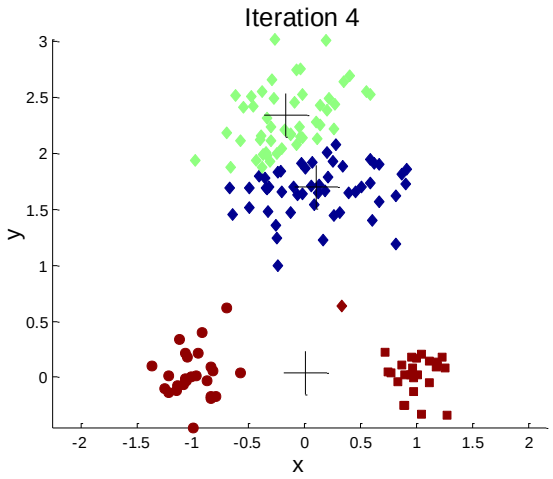


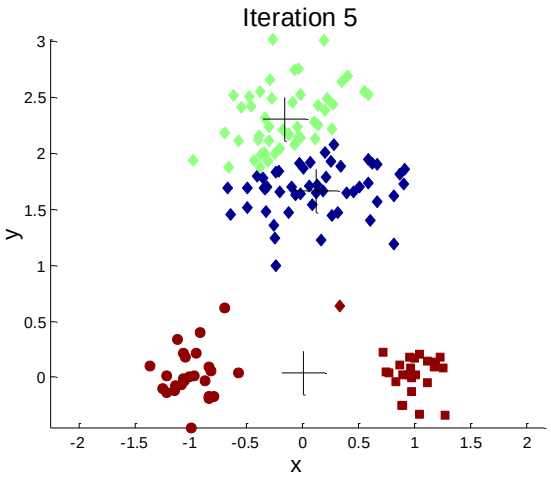
Παράδειγμα «κακής» αρχικοποίησης











[2]. **Πρόβλημα με empty clusters**

- Υπάρχει περίπτωση σε κάποιο επαναληπτικό βήμα του αλγορίθμου να υπάρχει μία κενή ομάδα, δηλ. να μην έχει κανένα σημείο.

- ✓ **Λύση**: Αντικαθιστούμε το κέντρο της κενής ομάδας με το πιο απομακρυσμένο σημείο από τα κέντρα των άλλων ομάδων.

[3]. **Πρόβλημα με ακραία σημεία (outliers)**

- Τα ακραία σημεία μπορούν να επηρεάσουν σημαντικά την διαδικασία ομαδοποίησης, καθώς μπορεί να μεταβάλλουν σημαντικά τα μέσα τους.

- ✓ **Λύση**: Μηχανισμός εντοπισμού των ακραίων σημείων, είτε πριν την ομαδοποίηση (προεπεξεργασία) είτε μετά (μετα-επεξεργασία), και αφαίρεσής τους.

[4]. **Complexity**

- Η πολυπλοκότητα σε **μνήμη** είναι μικρή καθώς επιπλέον μόνο τα K κέντρα απαιτούνται. Έτσι πολυπλοκότητα σε χώρο (***space***) :

$$O((N+K)*d) \text{ ,}$$

N: size of dataset – *d:* dimension of data

- Η πολυπλοκότητα σε **χρόνο** (***time***) είναι γραμμική ως προς τον αριθμό των δεδομένων, δηλ. *O(N)*, καθώς σε κάθε επανάληψη απαιτούνται *NxKxd* πράξεις.

[5]. **Επέκταση σε μη-Ευκλείδιους χώρους (K-medoids)**

Τροποποιήσεις του βασικού σχήματος

I. Συνάρτηση **ομοιότητας** (αντί για απόστασης)

$$sim\left(x_i, \mu_j\right)$$

II. Αντικειμενική συνάρτηση (**μεγιστοποίηση**)

$$E = \sum_{i=1}^{N} \max_{j=1,\ldots,K} \left\{ sim\left(x_i, \mu_j\right) \right\}$$

III. **Διάμεσος** (medoid) ως κέντρο της ομάδας Ω_j

$$\mu_j = x_k \in \Omega_j : \max_{x_k \in \Omega_j} \left\{ \sum_{x_i \in \Omega_j} sim\left(x_i, x_k\right) \right\}$$

[6]. Bisecting k-means (incremental learning)

Επαναληπτικά, επιλέγουμε μία ομάδα και κάνουμε **split**

- Αρχικά ***m=1*** ομάδα με ένα κέντρο για όλα τα σημεία.

- Repeated until *m=K*
  1. Select cluster $j \in [1, m]$ having center $\boldsymbol{\mu_j}$
  2. Split of j-th cluster by executing k-means locally for *K=2* to the subset of selected cluster's data (local k-means)
  3. Two new clusters are produced with centers:

$$\mu_j = \mu_j^{(new)} \quad , \quad \mu_{m+1}$$

  4. *m=m+1*

- Finally, execute (global) k-means with K centers to all data.

[7]. **Limitations** of K-means

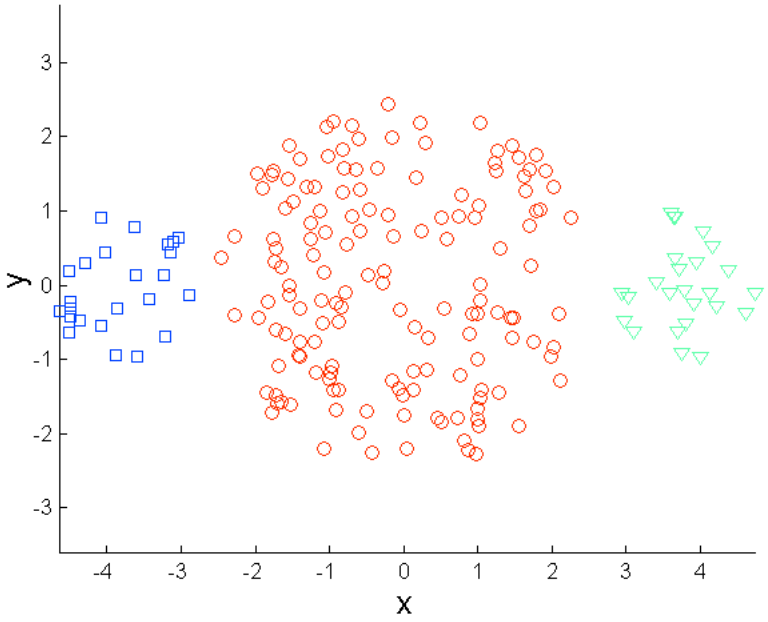
- Ο αλγόριθμος παρουσιάζει προβλήματα όταν οι ομάδες των δεδομένων είναι μη-σφαιρικές ή όταν είναι διαφορετικού μεγέθους ή διασποράς.

- Το μειονέκτημα του kmeans είναι ότι οι ομάδες που ψάχνει να βρει είναι του ιδίου μεγέθους, της ίδιας πυκνότητας και ότι το σχήμα τους είναι σφαιρικό.

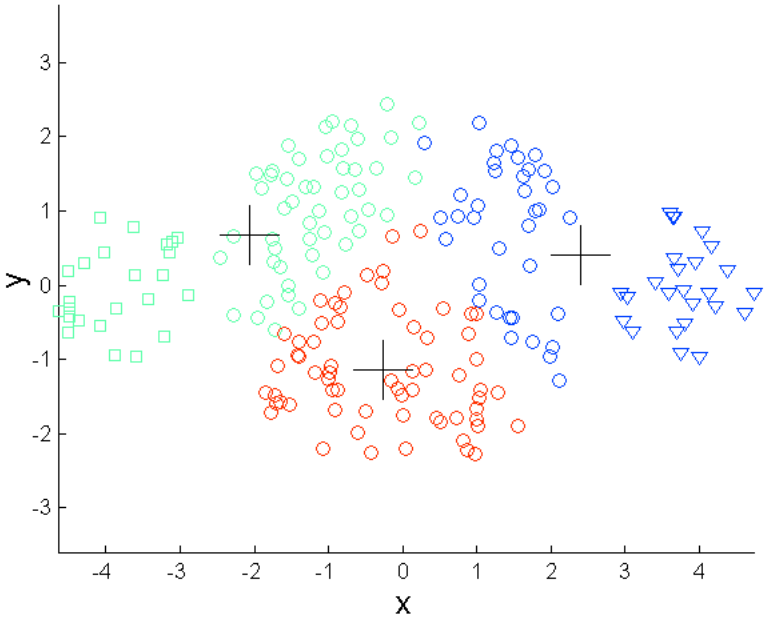
- Αντιμετώπιση:
  - Κάνουμε split στις ομάδες στο τέλος του αλγορίθμου
  - Εκτελώντας τον αλγόριθμο k-means για μεγαλύτερο αριθμό K από clusters.

[7]. **Limitations of K-means (cont.)**

- Clusters of different shape



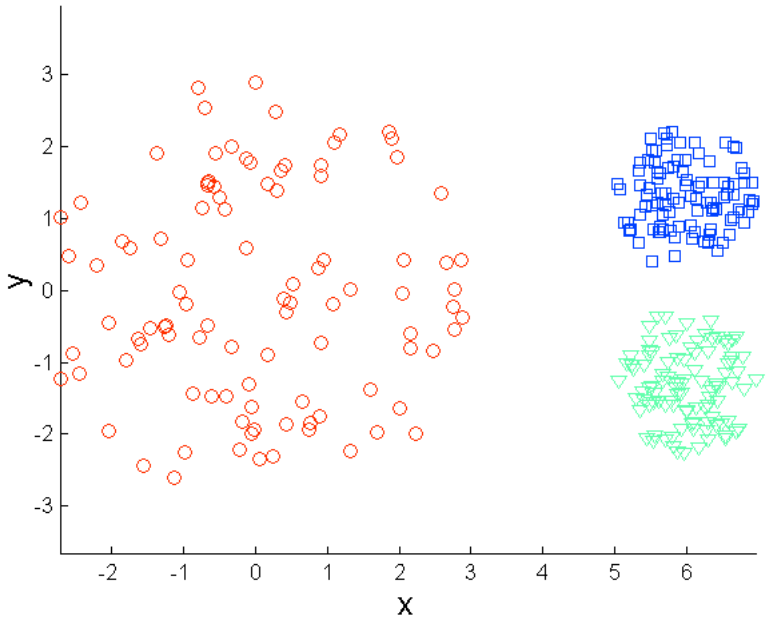
Initial dataset



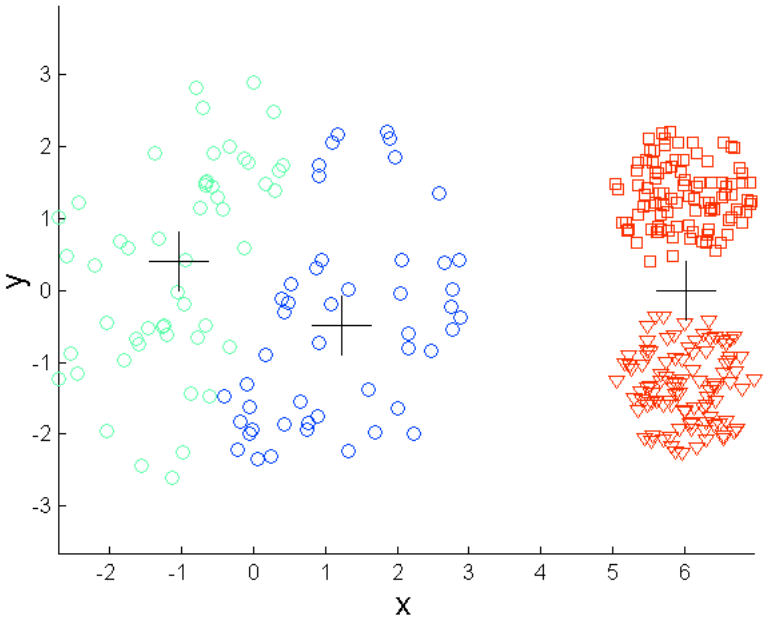
K-means solution (3 Clusters)

[7]. **Limitations of K-means (cont.)**

- Clusters of different variance



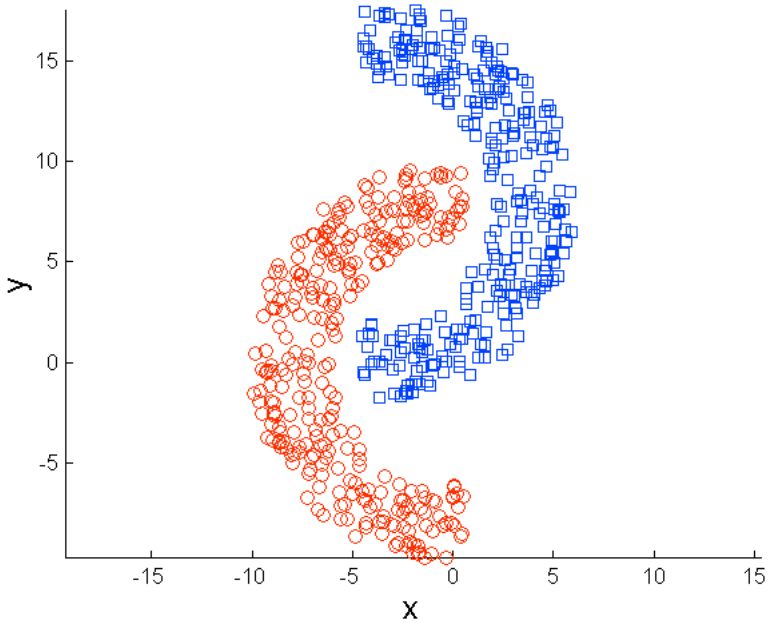
Initial dataset



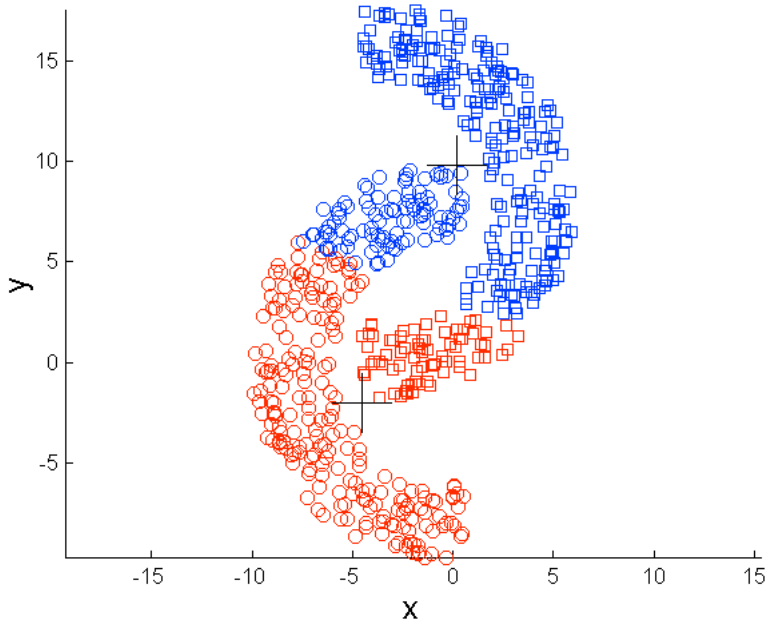
K-means solution (3 Clusters)

[7]. **Limitations of K-means (cont.)**

- Non-spherical shaped clusters



Initial dataset



K-means solution (2 Clusters)

[8]. Geometry of k-means

- Assume clustering into K=2 clusters.
- Decision mechanism of k-means:

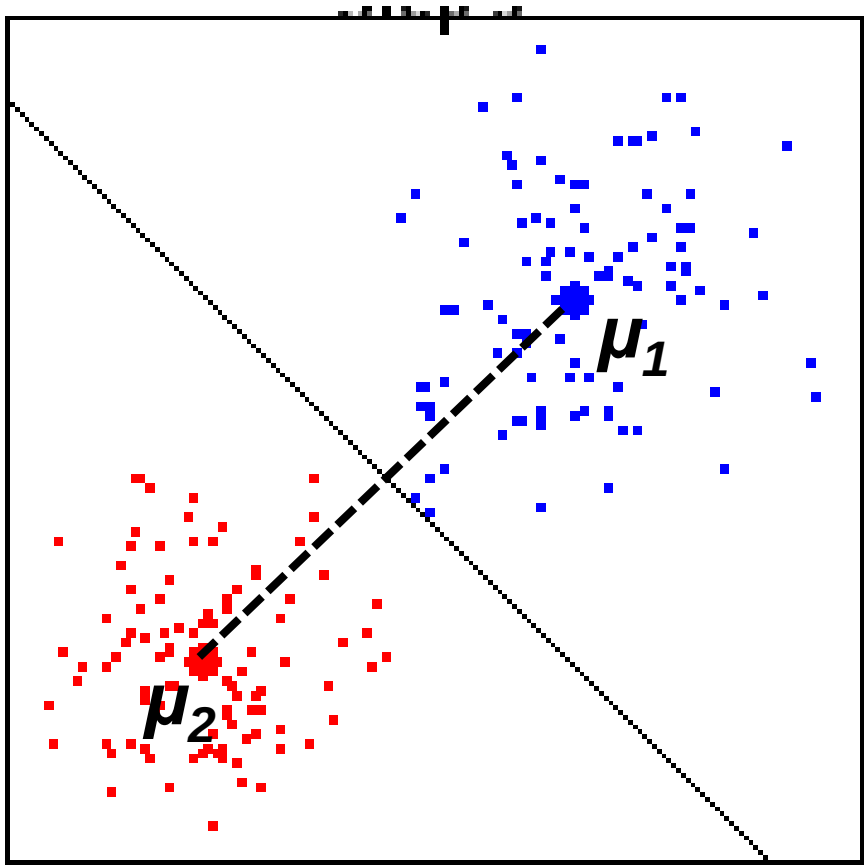
$$\|x-\mu_1\|^2 \underset{\Omega_2}{\overset{\Omega_1}{<>}} \|x-\mu_2\|^2$$

- Then we have: $(\mu_2-\mu_1)^T x + \frac{1}{2}\left(\|\mu_1\|^2 - \|\mu_2\|^2\right) \underset{\Omega_2}{\overset{\Omega_1}{<>}} 0$

of the form: $w^T x + b \underset{\Omega_2}{\overset{\Omega_1}{<>}} 0$

[8]. **Geometry of k-means (cont.)**

- **Thus**: k-means constructs specific **linear discriminant hyperplanes** among clusters, Bisector of **cluster centers' line (** μ_j , μ_k **)**.



[9]. Alternative Objective function (III)

$$E = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ dist\left(x_i, \mu_j\right) \right\} \Rightarrow \; E = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij} dist\left(x_i, \mu_j\right)$$

- όπου τα δυαδικά βάρη w_{ij} εκφράζουν την πληροφορία του σε ποια ομάδα ανήκουν τα σημεία

$$w_{ij} = \begin{cases} 1 & x_i \in \Omega_j \\ 0 & x_i \notin \Omega_j \end{cases}$$

- Το $n_j = \sum_{i=1}^{N} w_{ij}$ εκφράζει το πλήθος των δεδομένων που ανήκουν στην j-οστή ομάδα.
- Cluster centers: $\mu_j = \frac{1}{n_j} \sum_{i=1}^{N} w_{ij} x_i$

[10]. K-means as an optimization problem

- Objective function in Euclidean spaces:

$$E = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j\right) \right\} = \sum_{j=1}^{K} \sum_{x_i \in \Omega_j} \left\| x_i - \mu_j \right\|^2$$

- *Minimization problem* of centers $\{ \mu_j \}$

$$E = \sum_{j=1}^{K} \sum_{x_i \in \Omega_j} \left(x_i^T x_i - 2\mu_j^T x_i + \mu_j^T \mu_j \right)$$

- Setting the derivative of μ_j equal to zero:

$$\frac{\partial E}{\partial \mu_j} = 0 \Rightarrow \sum_{x_i \in \Omega_j} \left(-2x_i + 2\hat{\mu}_j \right) = 0 \Rightarrow \hat{\mu}_j = \frac{1}{\left| \Omega_j \right|} \sum_{x_i \in \Omega_j} x_i$$

- Sample mean is the optimum center parameter that minimizes the objective function.

[11]. Computer Vision application

Image segmentation and image compression

- **Image segmentation**: Division of image into regions (*segments*) of the same intensity.

- Let gray-scale image of size 512 x 512 pixels. Using 8 bits /pixel (256 intensity levels) a memory space of ***256 kB*** is required.

- **Apply the K-means** clustering approach to the 2^{18} pixel intensities for finding K clusters.

[11]. Computer Vision application (cont.)

- After finishing clustering, we **assign all pixels** of the same cluster with the intensity of their cluster center μ_j they belong.

- Then, new space required for image is ≈ ***(32 log$_2$K) kB***

e.g.	*K=2*	*32KB*
	K=4	*64KB*
	K=8	*96KB*

- **Image compression** with an error equal to the k-means objective function (after convergence):

$$E = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j\right) \right\} = \sum_{j=1}^{K} \sum_{x_i \in \Omega_j} \left\| x_i - \mu_j \right\|^2$$

[12]. Fuzzy c-means

- Extension of k-means using **Fuzzy sets theory**
- Objective function is written as

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{K} u_{ij}^{m}\left\|x_i - \mu_j\right\|^2 \qquad 1 < m \le \infty$$

- u_{ij} is the degree of **membership** of input x_i to cluster j , calculated (iteratively) as:

$$u_{ij} = \frac{1}{\sum_{k=1}^{K}\left(\frac{\left\|x_i - \mu_j\right\|}{\left\|x_i - \mu_k\right\|}\right)^{\frac{2}{m-1}}}$$

- Cluster centers calculation

$$\mu_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

[13]. Kernel k-means

- Extension of k-means using **kernel function $k(x_i, x_j)$**
- $x - \varphi(x)$
- Objective function is written as

$$E = \sum_{i=1}^{N} \min_{j=1,\ldots,K} \left\{ d\left(x_i, \mu_j\right) \right\} = \sum_{j=1}^{K} \sum_{x_i \in \Omega_j} \left\| x_i - \mu_j \right\|^2$$

or

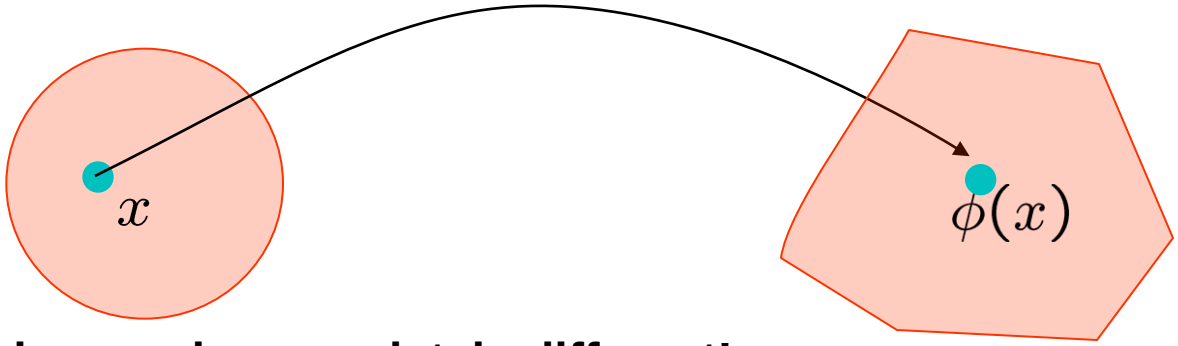
$$E = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij} \left\| x_i - \mu_j \right\|^2 \qquad w_{ij} = \begin{cases} 1 & x_i \in \Omega_j \\ 0 & x_i \notin \Omega_j \end{cases}$$

- Cluster centers

$$\mu_j = \frac{1}{n_j} \sum_{i=1}^{N} w_{ij} x_i$$

Idea on Kernel methods

1. **represent a point by its image in a feature space:**



2. **Domains can be completely different!**

3. **Kernel Trick: In many applications we do not need to know $\phi(x)$ explicitly, we only need to operate $\phi(x_i)^T \phi(x_j) = k(x_i, x_j)$ if the kernel can be computed efficiently (e.g. $\phi(x)$ can be infinite dimensional)**

[13]. Kernel k-means (cont.)

- Extended to kernel k-means

$$E = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij} \left\| \varphi(x_i) - \varphi(\mu_j) \right\|^2$$

$$n_j = \sum_{i=1}^{N} w_{ij}$$

$$\varphi(\mu_j) = \frac{1}{n_j}\sum_{i=1}^{N} w_{ij}\varphi(x_i)$$

- Each term is written (**kernel distance**)

$$\left\| \varphi(x_i) - \varphi(\mu_j) \right\|^2 = \left(\varphi(x_i) - \varphi(\mu_j)\right)^T \left(\varphi(x_i) - \varphi(\mu_j)\right) =$$

$$= \left(\varphi(x_i) - \frac{1}{n_j}\sum_{n=1}^{N} w_{nj}\varphi(x_n) \right)^T \left(\varphi(x_i) - \frac{1}{n_j}\sum_{n=1}^{N} w_{nj}\varphi(x_n) \right) =$$

$$= \varphi(x_i)^T\varphi(x_i) - \frac{2}{n_j}\sum_{n=1}^{N} w_{nj}\varphi(x_i)^T\varphi(x_n) + \frac{1}{n_j^2}\sum_{n=1}^{N}\sum_{m=1}^{N} w_{nj}w_{mj}\varphi(x_n)^T\varphi(x_m) =$$

$$= k(x_i, x_i) - \frac{2}{n_j}\sum_{n=1}^{N} w_{nj}k(x_i, x_n) + \frac{1}{n_j^2}\sum_{n=1}^{N}\sum_{m=1}^{N} w_{nj}w_{mj}k(x_n, x_m)$$

[13]. Kernel k-means (cont.)

- kernel k-means **objective function**

$$E = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\left[k(x_i, x_i) - \frac{2}{n_j}\sum_{n=1}^{N} w_{nj} k(x_i, x_n) + \frac{1}{n_j^2}\sum_{n=1}^{N}\sum_{m=1}^{N} w_{nj} w_{mj} k(x_n, x_m) \right]$$

- Need of **kernel (gram) matrix** calculation

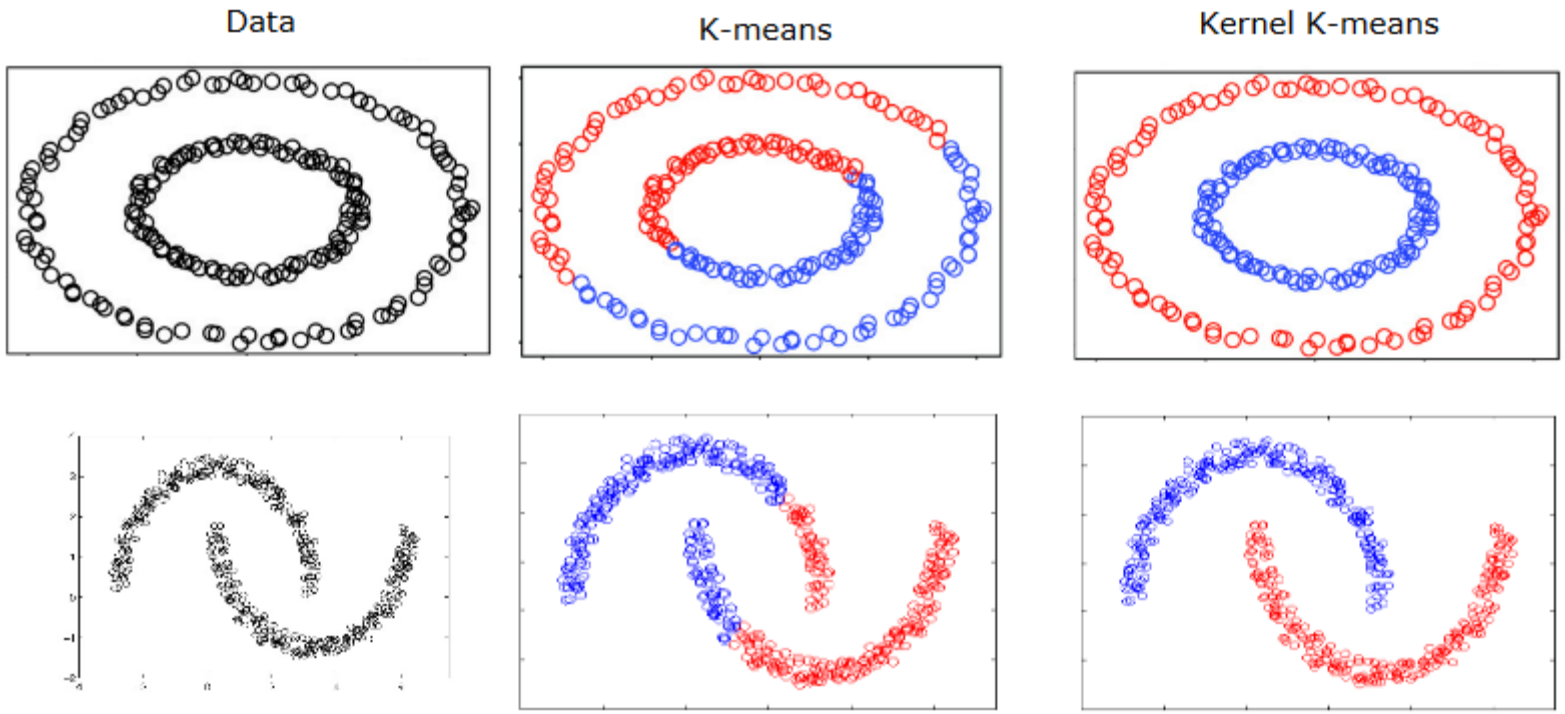
$$K = \left[K_{ij} = k(x_i, x_j) \right]_{N \times N}$$

- Binary (or weighted) w_{ij} by calculating the kernel distance

$$\left\| \varphi(x_i) - \varphi(\mu_j) \right\|^2$$

[13]. Kernel k-means (cont.)

- K-means vs. Kernel K-means



Kernel K-means is able to find "complex" clusters.

Learning Vector Quantization - LVQ

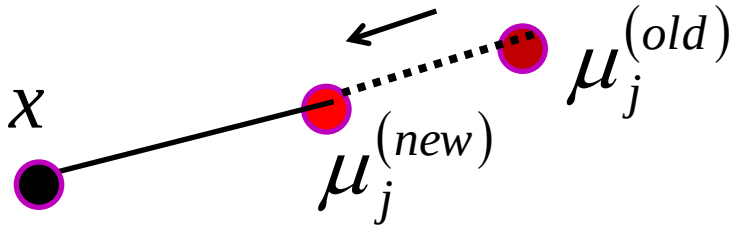
- Στόχος είναι η εύρεση K αντιπροσώπων { μ_j } ενός συνόλου δεδομένων
- Οι αντιπρόσωποι δρουν ως **κβαντιστές πληροφορίας** και προκαλούν **συμπίεση** των δεδομένων
- **Ανταγωνιστική μάθηση** (*competitive learning*):
  - Οι *K* κβαντιστές συναγωνίζονται μεταξύ τους για το ποιος θα «αποκτήσει» ένα νεοεισερχόμενο πρότυπο.
  - Η διανυσματική έκφραση του νικητή **προσαρμόζεται**
- 2 εκδόσεις: Με ή χωρίς επίβλεψη

LVQ for clustering

- Initialization of K centers $\left\{\mu_j^{(0)}\right\} \quad j=1,\ldots,K$
- Repeat
  1. Random selection of an input point $x \in X$
  2. Find winner cluster: $q = \arg\min_{j=1,\ldots,K}\left\{d\left(x,\mu_j\right)\right\}$
  3. Update center of winner cluster:

$$\mu_j^{(new)} = \mu_j^{(old)} + \eta\left(x - \mu_j^{(old)}\right) = \left(1-\eta\right)\mu_j^{(old)} + \eta x$$

$\eta < 1$ *learning rate*



[2]. Hierarchical Clustering

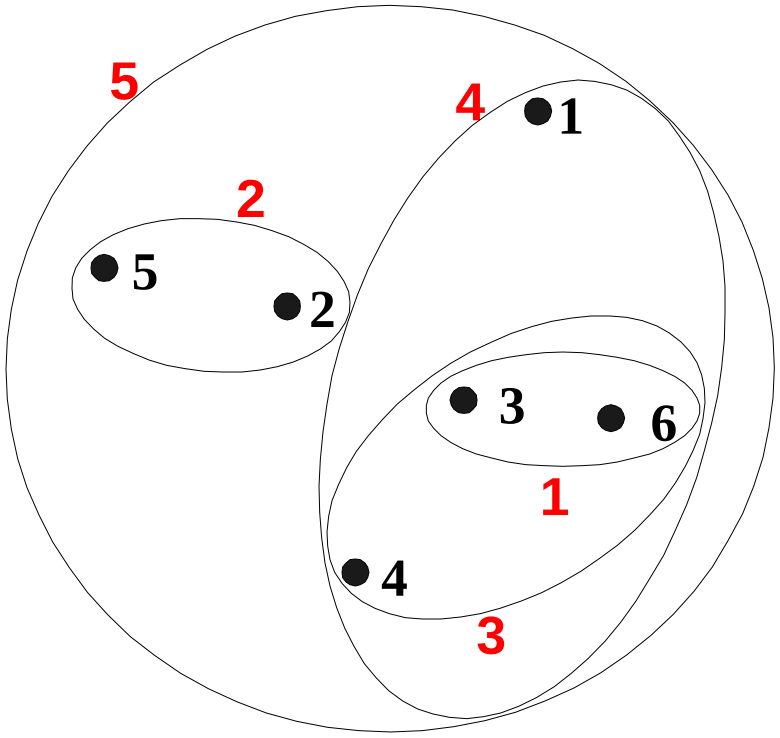
Δενδρική αναπαράσταση των δεδομένων

Πλεονεκτήματα :

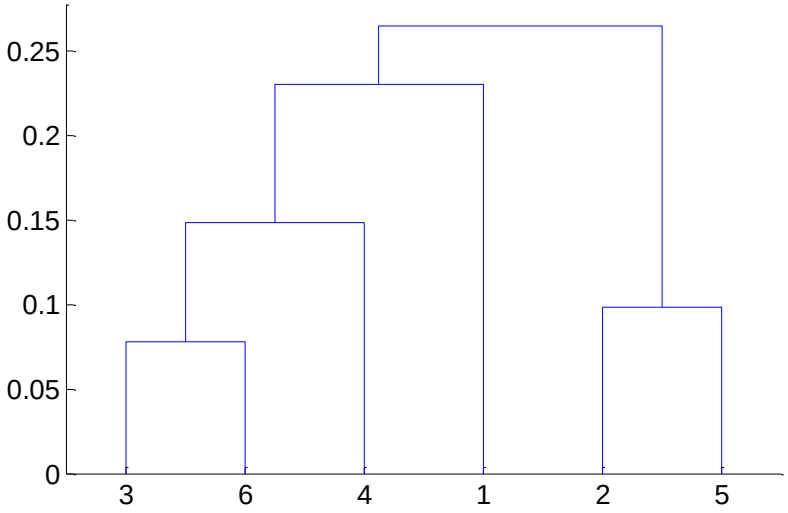
- Όχι εξάρτηση από αρχικοποίηση
- Ταυτόχρονα **πολλαπλές λύσεις** για διαφορετικό αριθμό ομάδων (ύψος δέντρου)
- Η μέθοδος είναι γενική καθώς μπορεί εύκολα να επεκταθεί σε μη-Ευκλείδιους χώρους.

Υπάρχουν **2 τρόποι κατασκευής** του δέντρου

Hierarchical Clustering



Nested clusters



Dendrogram

Divisive method

- **Top-down** tree construction

Initially one cluster – root of tree (***k=1***)

Repeat

1. **Select a cluster j** (leaf node of current tree) according to an appropriate criterion.
2. **Split this cluster** (parent) into two non-overlapping children using a proper mechanism.
3. *k=k+1*

Until *k=M*

Divisive method (cont.)

- **Selection method:** Usually based on max variance criterion or cluster's sparseness.

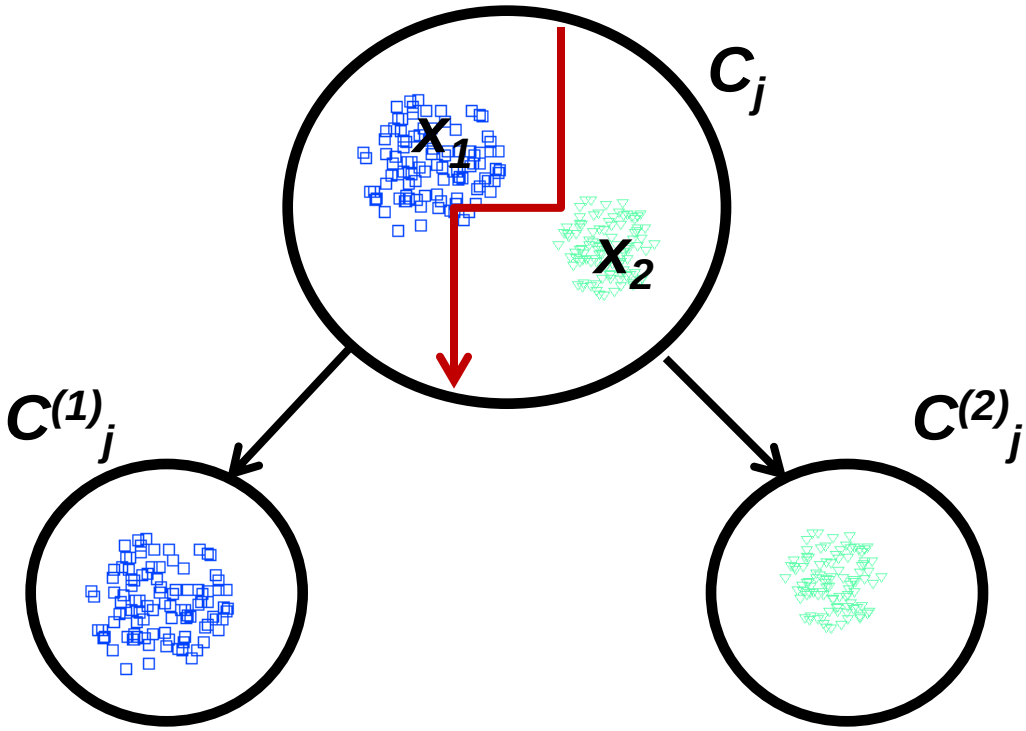
- **Cluster Splitting**: find two cluster members $\{x_1, x_2\}$ such that:

$$\min_{\{x_1, x_2\} \in C_j} \sum_{x_n \in C_j} \min\{dist(x_n, x_1), dist(x_n, x_2)\}$$

Then locate members of j-cluster to two children according to distances with two sub-cluster representatives $\{x_1, x_2\}$.

Divisive method (cont.)

- Splitting procedure



Agglomerative method

- **Bottom-up** tree construction.

1. Initially a tree with **N leaf-nodes** (every point forms a separated cluster). (***k=N***)
- **Repeat**
  - **Find two most common clusters (parents)** C_1, C_2 from the current tree.
  - And ***merge*** them to a larger cluster $C=C_1 \cup C_2$
  - *k=k-1*
- **Until *k=M***

Agglomerative method (cont.)

- **Disadvantage**: High complexity $O(N^2)$

- **Criteria for merging**
  - Minimum distance of cluster centers.
  - Total mean distance among the members of both clusters (***Group average***)
  - Maximum distance among both clusters' members
  - Increment of cluster's variance after merging two clusters (***Ward's method***)

Agglomerative method (cont.)

Agglomerative clustering example