

## Week 3

# Course. Introduction to Machine Learning

## Theory 3. Introduction to unsupervised learning and Cluster Analysis (Part II)

Dr. Maria Salamó Llorente

[maria.salamo@ub.edu](mailto:maria.salamo@ub.edu)

Dept. Mathematics and Informatics,  
Faculty of Mathematics and Informatics,  
University of Barcelona (UB)

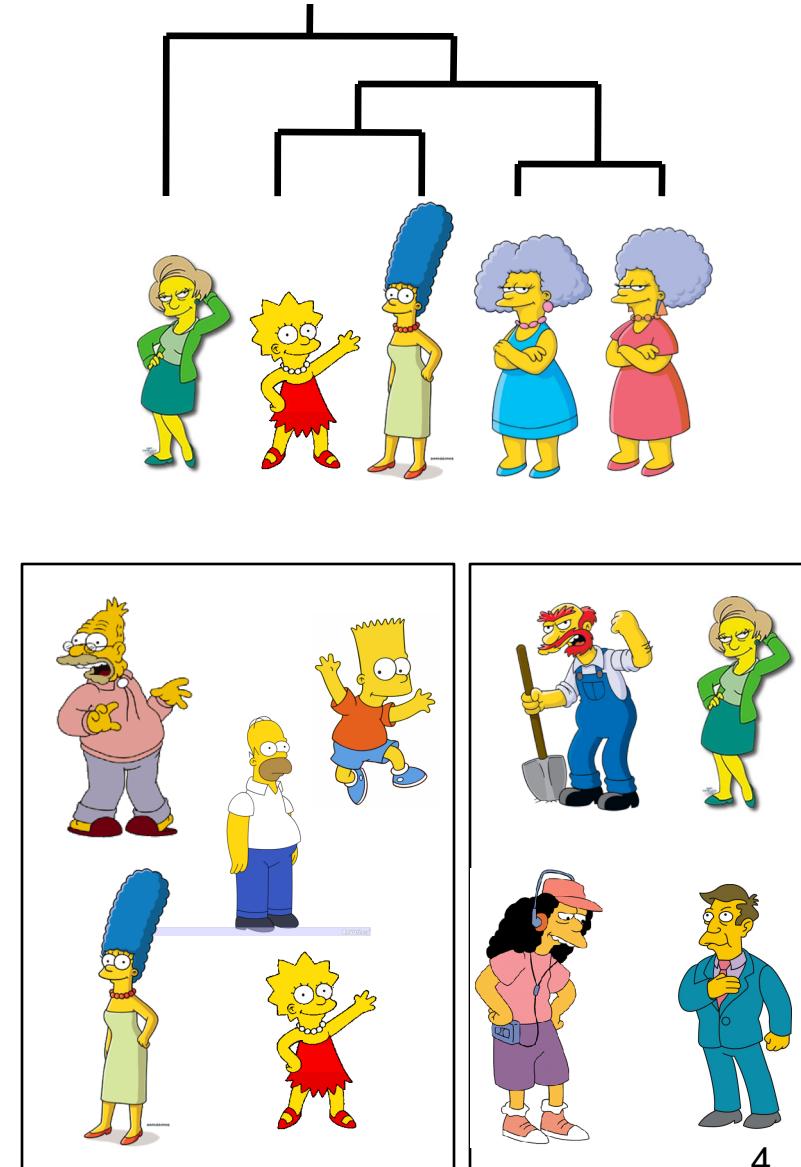
# Contents

- 1. Introduction to unsupervised learning (Theory 2)**
  1. Introduction to unsupervised learning
  2. Examples
  3. Definition of unsupervised learning
  4. Unsupervised learning approaches
- 2. Introduction to Cluster analysis (Theory 2)**
  1. Defining clustering analysis
  2. Areas that apply clustering
  3. Classification of clustering algorithms
- 3. Hierarchical clustering (Theory 2)**
- 4. Partitional clustering (Theory 3)**
  1. K-Means algorithm,
  2. Bisecting K-Means,
  3. Fuzzy C-Means
  4. EM (expectation maximization algorithm)

# Partitional clustering

# Recall the Taxonomy

- **Hierarchical algorithms**
  - Examples are organized as a binary tree
  - No explicit division in groups
    - Bottom-up
    - Top-down
- **Partitional algorithms**
  - Usually start with a random (partial) partitioning
  - Refine it iteratively:
    - K-means clustering
    - Mixture-model based clustering



# Partitional algorithms

- **Method:** construct a partition of  $n$  objects into a set of  $k$  clusters
- **Given:** a set of objects (training set) and typically must provide the number of desired clusters,  $K$ .
- **Basic process:**
  - Randomly choose  $K$  instances as *seeds*, one per cluster
  - Form initial clusters based on these seeds
  - Iterate, repeatedly reallocating instances to different clusters to improve the overall clustering
  - Stop when clustering converges or after a fixed number of iterations



# K-Means



Chapter 8 - Cluster Analysis: Basic concepts and algorithms



Section 8.2 K-means

# K-Means basis

- Assumes instances are **real-valued vectors**
- Clusters based on *centroids*, *center of gravity*, or mean of points in a cluster,  $\mathbf{c}$ :

$$\vec{\mu}(\mathbf{c}) = \frac{1}{|\mathbf{c}|} \sum_{\vec{x} \in \mathbf{c}} \vec{x}$$

- Reassignment of instances to clusters is **based on distance** to the current cluster centroids

- **Euclidean distance ( $L_2$  norm):**

$$L_2(\vec{x}, \vec{y}) = \sum_{i=1}^m (x_i - y_i)^2$$

- **$L_1$  norm:**

$$L_1(\vec{x}, \vec{y}) = \sum_{i=1}^m |x_i - y_i|$$

- **Cosine Similarity** (transform to a distance by subtracting from 1):

$$1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

# K-Means Algorithm

*Let  $d$  be the distance measure between instances*

1. Decide on a value for  $k$
2. Select  $k$  random instances  $\{s_1, s_2, \dots, s_k\}$  as seeds.
3. (Decide the class membership)

*For each instance  $x_i$ :*

*Assign  $x_i$  to the cluster  $c_j$  such that  $d(x_i, s_j)$  is minimal.*

4. (Update the seeds to the centroid of each cluster)

*For each cluster  $c_j$*

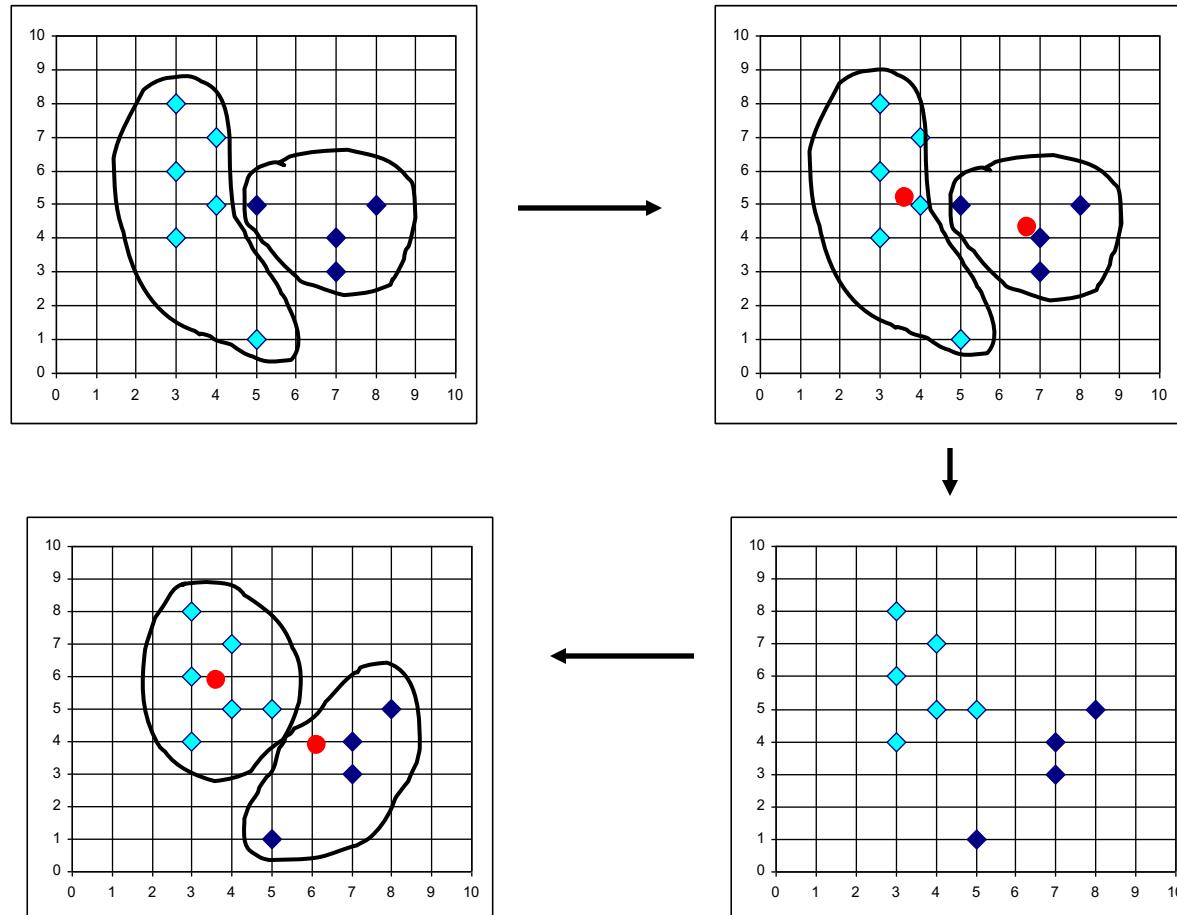
$$s_j = \mu(c_j)$$

$$\vec{\mu}_k = \frac{1}{c_k} \sum_{i \in c_k} \vec{x}_i$$

5. (Until clustering converges or other stopping criterion):  
*If none of the  $N$  instances changed membership, exit  
Otherwise, go to step 3*

# The *K*-Means Clustering Method

# Example



# Time Complexity

- Assume computing distance between two instances is  $O(m)$  where  $m$  is the dimensionality of the vectors
- **Reassigning clusters:**  $O(kn)$  distance computations, or  $O(knm)$ .
- **Computing centroids:** Each instance vector gets added once to some centroid:  $O(nm)$ .
- Assume these two steps are each done once for  $l$  iterations:  $O(lknm)$ .
- Linear in all relevant factors, assuming a fixed number of iterations, more efficient than  $O(n^2)$  or  $O(n^3)$  HAC.

# K-Means Objective

- The **objective** of k-means is to **minimize the total sum of the squared distances of every point to its corresponding cluster centroid**

$$\text{Goodness measure (SD)} = \sum_{l=1}^K \sum_{x_i \in X_l} \| x_i - \mu_l \|^2$$

- *Finding the global optimum is NP-hard*
- *The k-means algorithm is guaranteed to converge a local optimum*

- **Strength**
  - *Relatively efficient*:  $O(tkn)$ , where  $n$  is # instances,  $k$  is # clusters, and  $t$  is # iterations. Normally,  $k, t \ll n$ .
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *simulated annealing* or *genetic algorithms*
- **Weakness**
  - Applicable only when *mean* is defined; what about categorical data?
  - Need to specify **k**, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

- Results can vary based on random seed selection
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic or the results of another method (such as a hierarchical algorithm)
  - Try out multiple starting points (**very important!**)
  - Initialize with the results of another method



# Bisecting K-Means



Chapter 8 - Cluster Analysis: Basic concepts and algorithms



Section 8.2.3 Bisecting K-means

# Variations of the *K-Means* method

- A few variants of the *k-means* which differ in
  - Selection of the initial  $k$  means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

Start with a single cluster of all objects

1. Pick a cluster to split
2. Find two subclusters using the basic k-Means algorithm (bisecting step)
3. Repeat step 2 for  $n$  times and take the split that produces the clustering with the highest overall similarity
4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached

Different ways to choose the cluster which is to be split: (a) largest cluster, (b) the most heterogeneous cluster, (c) the largest cluster which has a predetermined degree of heterogeneity, ....

# Fuzzy C-Means

# Crisp Clustering

- Attempts to cluster data by grouping related attributes in uniquely defined clusters
  - Each data point in the data set is assigned to only one cluster
- Clustering is an iterative process of finding better and better clusters centers
- When clusters are well separated, a crisp classification of data points into clusters make sense
- But in many cases, clusters are not well separated
  - In a crisp classification, a borderline data point ends up being assigned to a cluster in an arbitrary manner

# Fuzzy Clustering

- Fuzzy set theory was introduced by Lofti Zadeh in 1969 to overcome the idea that all things can be absolutely *True* or *False*.
  - Zadeh was the first to claim that something can be 0.70 true
- According to fuzzy Algebra every element of the universe can belong to any **fuzzy set (FS)** with a **degree of membership** that varies from 0 to 1 taking real values
  - Each instance belongs to every cluster with some weight
- If an element of the universe belongs to a FS with a degree of  $\mu_1$  then it belongs to its complement with a degree  $(1 - \mu_1)$



# Example of Fuzzy Set

$$X = \{x_1, x_2, x_3, x_4, x_5\}$$

$$A: \mu_A = [\mu_A(x_1) \ \mu_A(x_2) \dots \ \mu_A(x_n)]$$

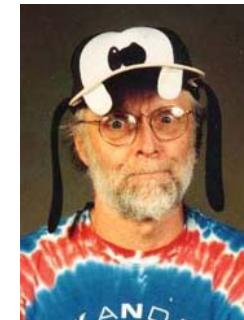
$$\begin{aligned}\mu_A(x_1) &= .5 & \mu_A(x_2) &= .8 & \mu_A(x_3) &= .2 \\ \mu_A(x_4) &= .3 & \mu_A(x_5) &= .2\end{aligned}$$

*Or equivalently*

$$\begin{array}{ll} \{x_1, x_2, x_3, x_4, x_5\} & \{x_1, x_2, x_3, x_4, x_5\} \\ \mu_A = [0.5, 0.8, 0.2, 0.3, 0.2] & \mu_B = [0.5, 0.2, 0.8, 0.7, 0.8] \end{array} \quad 21$$

- “**linguistic**” terms like “*much*”, “*much more*”, “*less*”, “*more or less*”, “*more than*” and others can be used in fuzzy clustering
- Data points are given partial **degree of membership** in multiple nearby clusters
- Central point in the fuzzy clustering is always **no unique partitioning** of the data in a collection of clusters. In this **membership value** is assigned to each cluster.
  - Sometimes this membership has been used to decide whether the data points belong to the cluster or not

- Useful in Fuzzy Modeling
  - Identification of the fuzzy rules needed to describe a “black box” system, on the basis of observed vectors of inputs and outputs
- Several approximations
  - **FCM**: Fuzzy C-Means Clustering (Bezdek, 1981)    **Prof. Bezdek**
  - **PCM**: Possibilistic C-Means Clustering (Krishnapuram - Keller, 1993)
  - **FPCM**: Fuzzy Possibilistic C-Means (N. Pal - K. Pal - Bezdek, 1997)



- **Input:** Unlabeled data set
  - N is the number of data points in  $X$ ,  $X = \{x_1, x_2, \dots, x_n\}$
  - $x_i \in \mathcal{R}^p$  where  $p$  is the number of features in each vector
- **Main output:**
  - A  $c$ -partition of  $X$ , which is  $c \times n$  matrix  $U$ 
    - $c$  is the number of clusters
    - $U$  is called the universe
- **Additional output**
  - Set of vectors  $V = \{v_1, v_2, \dots, v_c\} \subset \mathcal{R}^p$
  - $v_i$  is called “**cluster center**”

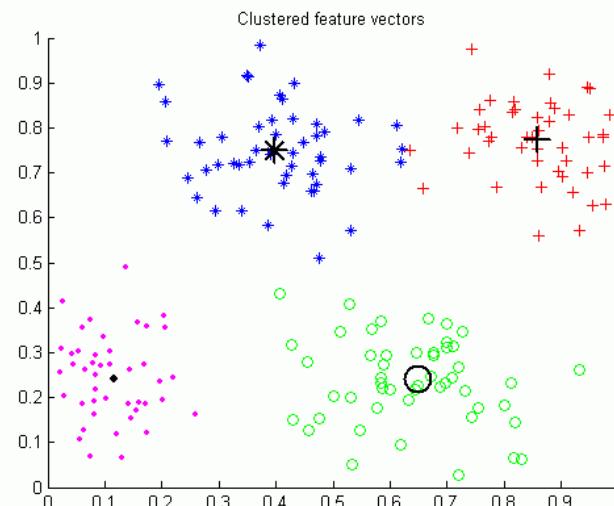
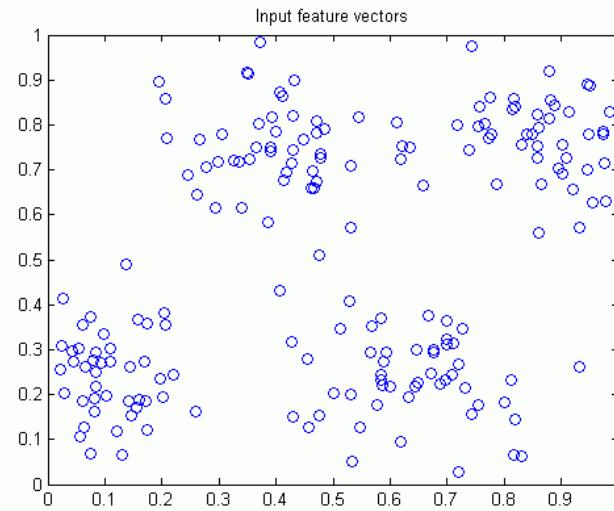
# Fuzzy C-Means Clustering

features  
 $p = 2$

instances  
 $X$   
 $n = 188$

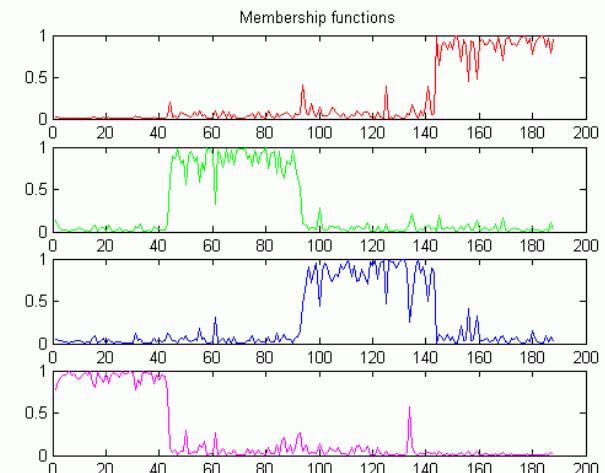
$U$

$c = 4$   
number of  
clusters



## Sample Illustration

*Rows of  $U$   
(Membership Functions)*



- Goal: Optimization of an “**objective function**” or “**performance index**”

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c (U_{ik})^m \|x_k - v_i\|^2,$$

$$1 \leq m \leq \infty$$

- $J_m$  minimizes the total sum of all distances
- Constraint  $\sum_{i=1}^c u_{ik} = 1, \forall k$
- Degree of fuzzification  $m \geq 1$

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c (U_{ik})^m \|x_k - v_i\|^2,$$

$$1 \leq m \leq \infty$$

- $m$  is the fuzzy exponent
- $U$  is the membership matrix
- $V$  are the cluster centers
- $\|x_k - v_i\|$  is the distance between the data  $x_k$  and the cluster center  $v_i$
- $m \geq 1$  governs the influence of membership grades
  - FCM converges from any  $m \in (1, \infty)$

## ***GOAL: Minimizing Objective Function***

- Zeroing the gradient of  $\mathcal{J}_m(U, V)$  with respect to  $V$

$$U_{ik} = \left( \sum_{j=1}^c \left( \frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{2/(m-1)} \right)^{-1} \quad \forall i, \forall k \quad (\text{Eq. 1})$$

- Zeroing the gradient of  $\mathcal{J}_m(U, V)$  with respect to  $U$

$$v_i = \frac{\sum_{k=1}^n (U_{ik})^m x_k}{\sum_{k=1}^n (U_{ik})^m} \quad (\text{Eq. 2})$$

*Note: It is the Center of Gravity*

## Pick

- Initial Choices
  - Number of clusters ,  $1 < c < n$
  - Maximum number of iterations (Typ.: 100),  $T$
  - Weighting exponent (Fuzziness degree),  $m$ 
    - $m=1$ : crisp
    - $m=2$ : Typical
  - Termination measure  $E_t = \| V_t - V_{t-1} \|$ ,  $\leftarrow$  1-norm
  - Termination threshold (Typ. 0.01) ( $0 < \varepsilon$ )

## ***Iterative FCM algorithm***

- Guess Initial Cluster Centers  $V_0 = (V_{1,0}, \dots, V_{c,0}) \in \mathcal{R}^{cp}$
- Alternating Optimization (AO)

$t \leftarrow 0$

**REPEAT**

$t \leftarrow t + 1$

Compute matrix  $U_t$  (Eq. 1)

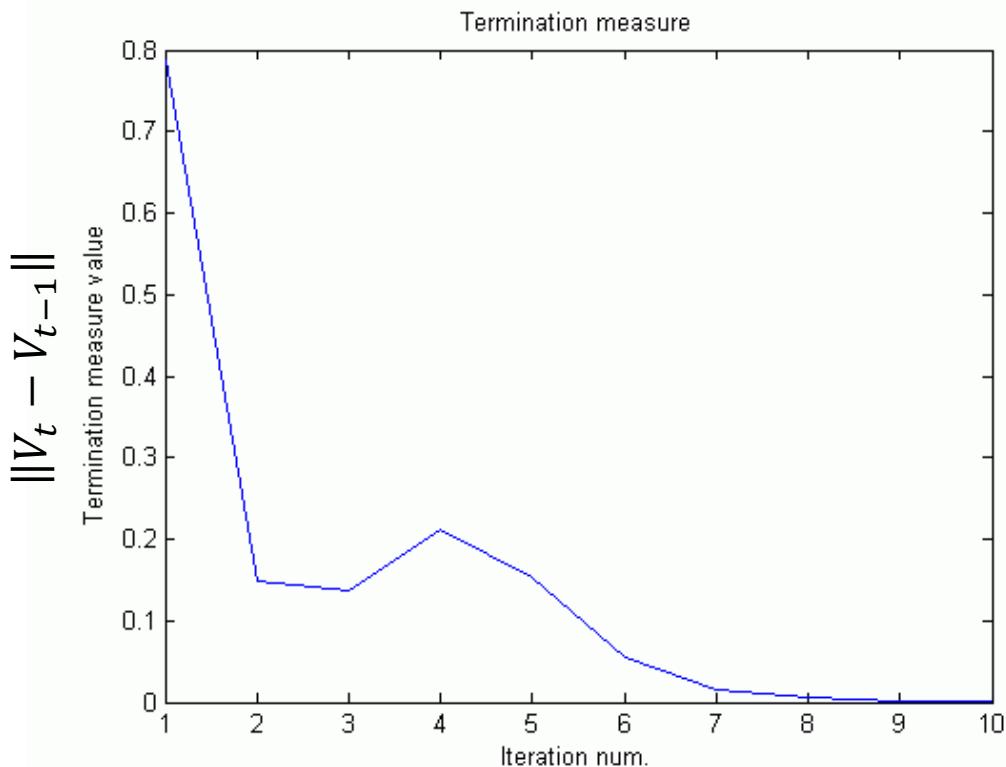
Compute associated clusters centers  $V_t$  (Eq. 2)

**UNTIL** ( $t = T$  or  $\|V_t - V_{t-1}\| \leq \varepsilon$ )

$(U, V) \leftarrow (U_t, V_t)$

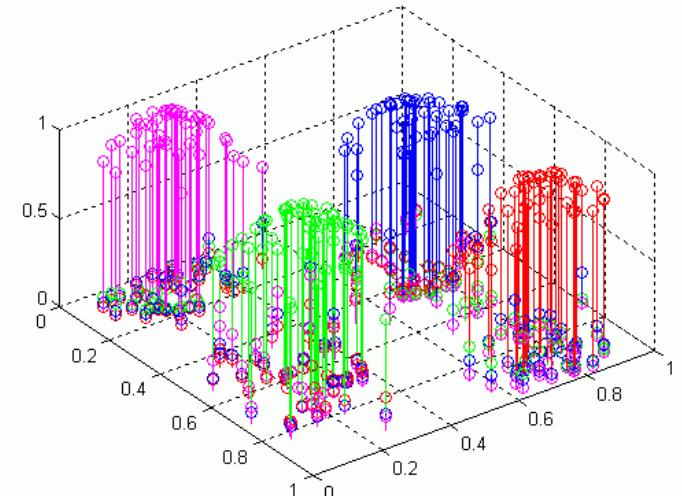
## Sample Termination Measure Plot

**Termination Measure Values**



$m = 2.0$

**Final Membership Degrees**



*Note that when advancing iterations,  
the error is decreasing*

$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$  and  $c = 3$

## Fuzzy Clusters (*this is U matrix*)

$$Cl_1 : [ 0.6 \quad 0.7 \quad 0.3 \quad 0.1 \quad 0.4 \quad 0.2 \quad 0.1 ]$$

$$Cl_2 : [ 0.1 \quad 0.1 \quad 0.3 \quad 0.5 \quad 0.1 \quad 0.7 \quad 0.1 ]$$

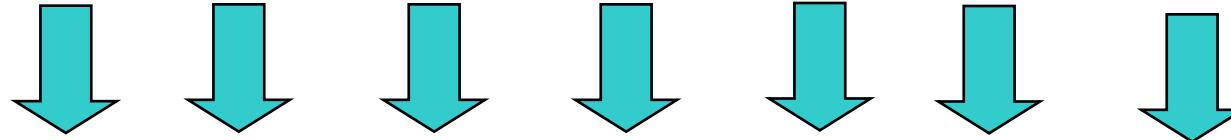
$$Cl_3 : [ 0.3 \quad 0.2 \quad 0.4 \quad 0.4 \quad 0.5 \quad 0.1 \quad 0.8 ]$$

## Fuzzy Clusters

$$Cl_1 : [ 0.6 \quad 0.7 \quad 0.3 \quad 0.1 \quad 0.4 \quad 0.2 \quad 0.1 ]$$

$$Cl_2 : [ 0.1 \quad 0.1 \quad 0.3 \quad 0.5 \quad 0.1 \quad 0.7 \quad 0.1 ]$$

$$Cl_3 : [ 0.3 \quad 0.2 \quad 0.4 \quad 0.4 \quad 0.5 \quad 0.1 \quad 0.8 ]$$



## Crisp Clusters from Fuzzy Clusters

$$Cl_1 : [ 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 ]$$

$$Cl_2 : [ 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 ]$$

$$Cl_3 : [ 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 ]$$

## Example Fuzzy clusters to Crisp clusters

$$Cl_1 : [ \begin{array}{ccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} ]$$

$$Cl_2 : [ \begin{array}{ccccccc} 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array} ]$$

$$Cl_3 : [ \begin{array}{ccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{array} ]$$

$$Cl_1 : \{ x_1 \quad x_2 \quad | \quad | \quad | \quad \} \\$$

$$Cl_2 : \{ \quad \downarrow \quad x_4 \quad \downarrow \quad x_6 \quad \downarrow \quad \}$$

$$Cl_3 : \{ x_3, x_5, x_7 \}$$

# *Crisp Clusters*

$$Cl_1 : \{ x_1, x_2 \}$$

$$Cl_2 : \{ x_4, x_6 \}$$

$$Cl_3 : \{ x_3, x_5, x_7 \}$$

- **Advantages**
  - Unsupervised
  - Always converges
- **Disadvantages**
  - Long computational time
  - Sensitivity to the initial guess (speed, local minima)
  - Sensitivity to noise
    - One expects low (or even no) membership degree for outliers (noisy points)

## **Optimal Number of Clusters**

- Performance Index

$$\min_{(c)} \left\{ P(c) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m (\|x_k - v_i\|^2 - \|v_i - \bar{x}\|^2) \right.$$

*Average of all feature vectors*       $\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k$

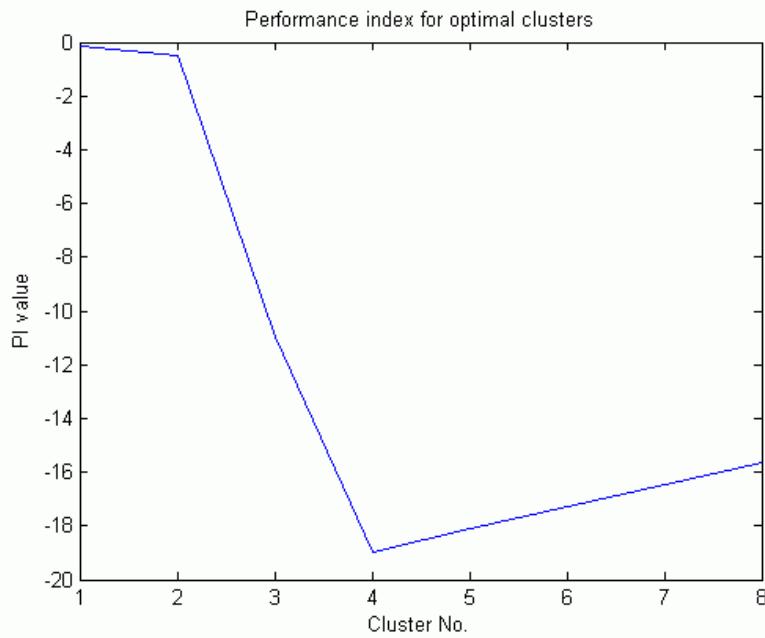
$$\sum_{i=1}^c \sum_{k=1}^n u_{ik}^m (\|x_k - v_i\|^2) - \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m (\|v_i - \bar{x}\|^2)$$

*Sum of the*  
**within fuzzy cluster fluctuations**  
*(small value for optimal c)*

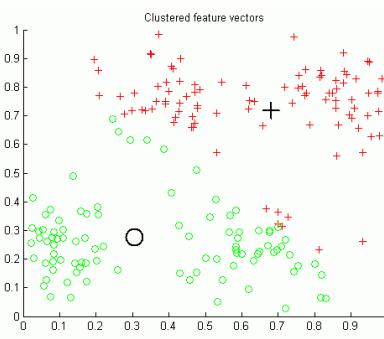
*Sum of the*  
**between fuzzy cluster fluctuations**  
*(big value for optimal c)*

## *Optimal Cluster No. (Example)*

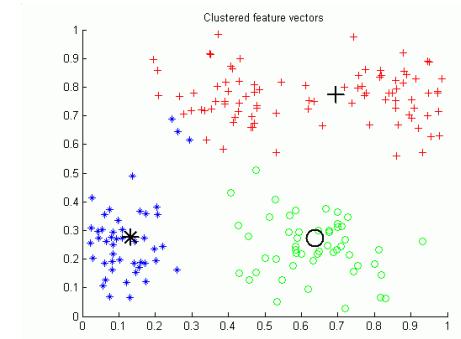
**Performance index for optimal clusters  
(is minimum for  $c = 4$ )**



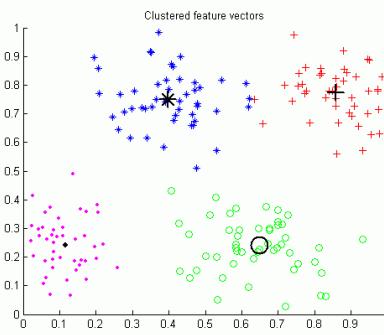
$c = 2$



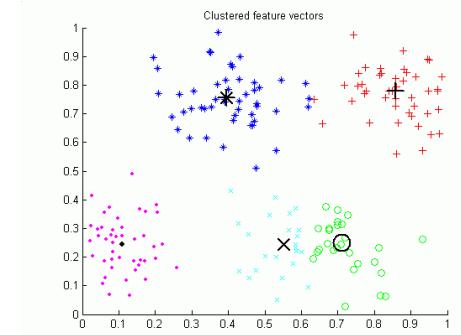
$c = 3$



$c = 4$



$c = 5$



# Fuzzy c-Means

Arriving at this point if you need to clarify concepts, you can read:



- Efficient Implementation of the Fuzzy c-Means Clustering Algorithms



Robert L. Cannon, Jitendra V. Dave, James C. Bezdek

IEEE Transactions on pattern analysis and machine intelligence,  
vol PAMI-8, nº 2, March 1986

- Pages 248 and 249, which correspond to Sections I and II
- The remaining of the article is interesting but it is an optional reading

# More references of FCM

- J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3: 32-57.
- **C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York**
- **J. C. Bezdek, R. Ehrlich, W. Full (1984). FCM: The fuzzy c-Means Algorithm.**
- James C. Bezdek, James Keller, Raghu Krishnapuram and Nikhil R. Pal (1999), *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic Publishers, TA 1650.F89.
- R. Krishnapuram and J. M. Keller (1993) A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 2, pp. 98-110.
- N. R. Pal, K. Pal and J. C. Bezdek (1997), "A mixed c-means clustering model," *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Vol. 1, pp. 11-21.
- Jun Yan, Michael Ryan and James Power, *Using fuzzy logic Towards intelligent systems*, Prentice Hall, 1994.



# Expectation-Maximization (EM)



***Look at the video***

[https://youtu.be/REypj2sy\\_5U](https://youtu.be/REypj2sy_5U)

# Soft Clustering

- ***Hard clustering*** Clustering typically assumes that each instance is given a “hard” assignment to exactly one cluster
  - Does not allow uncertainty in class membership or for an instance to belong to more than one cluster
- ***Soft clustering* gives probabilities** that an instance belongs to each of a set of clusters
- Each instance is assigned a probability distribution across a set of discovered categories
  - probabilities of all categories must sum to 1

## Mixture Models

- Probabilistically-grounded way of doing soft clustering
- Each cluster: a generative model (Gaussian or multinomial)
- Parameters (e.g. mean/covariance are unknown)
- Expectation Maximization (EM) algorithm
  - Automatically discover all parameters for the K “sources”

# Expectation Maximization (EM)

- The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin
  - They pointed out that the method had been “proposed in many times in special circumstances” by earlier authors.
- EM is typically used to compute **maximum likelihood estimates** (MLE) of parameters of an underlying distribution from given data set when the data is incomplete or has missing values

- Filling in missing data in samples
- Discovering the value of latent variables
- Estimating the parameters of GMMs
- Estimating parameters of finite mixtures
- **Unsupervised learning of clusters**
- **Semi-supervised classification and clustering**

- **Main idea:** Use probabilities instead of distances
- **Goal:**
  - Find the most likely clusters given the data
  - Determine the probability with which an object belongs to a certain cluster

$$\Pr(C | x) = \frac{\Pr(x | C)\Pr(C)}{\Pr(x)} = \frac{\Pr(x | C)\Pr(C)}{\sum_C \Pr(C)\Pr(x | C)}$$

where  $\Pr(C)$  is the probability that a randomly selected object belongs to cluster C, and  $\Pr(x|C)$  is the probability of observing the object x given the cluster C

- Let us assume that we know that there are  $k$  clusters
- To learn the clusters, we need to determine their parameters (means and standard deviations)

- Probabilistic method for soft clustering
  - It uses probabilities instead of distances!
- Direct method that assumes  $k$  clusters: $\{c_1, c_2, \dots, c_k\}$
- Soft version of  $k$ -means
- Assumes a probabilistic model of categories that allows computing  $P(C | x)$  for each category or cluster,  $C$ , for a given example,  $x$
- For text, typically assume a naïve-Bayes category model

# EM Algorithm

- **Iterative method** for learning probabilistic categorization model from unsupervised data.
- Initially assume random assignment of examples to categories.
- Learn an initial probabilistic model by estimating model parameters  $\theta$  from this randomly labeled data.
- **Algorithm:** iterate following two steps until convergence:
  - ***Expectation (E-step):*** Compute  $P(C | x)$  for each example given the current model, and probabilistically re-label the examples based on these posterior probability estimates.
  - ***Maximization (M-step):*** Re-estimate the model parameters,  $\theta$ , from the probabilistically re-labeled data.

# EM algorithm (cont.)

1. Calculate cluster probability (represented as object weights) for each object (**Estimation step**)
2. Estimate distribution parameters based on the cluster probabilities (**Maximization step**)
3. Procedure stops when log-likelihood saturates

Log-likelihood:  $\sum_i \log(p_A \Pr(x_i|A) + p_B \Pr(x_i|B))$

## Estimating parameters from weighted objects:

Mean of cluster  $A$ :  $\mu_A = \frac{w_1x_1 + w_2x_2 + \dots + w_nx_n}{w_1 + w_2 + \dots + w_n}$

Standard deviation:  $\sigma_A^2 = \frac{w_1(x_1 - \mu)^2 + w_2(x_2 - \mu)^2 + \dots + w_n(x_n - \mu)^2}{w_1 + w_2 + \dots + w_n}$

# EM example

- Suppose we are given two sets of samples, red and blue, drawn from two different normal distributions.
- Our goal will be to find the mean and standard deviation for each group of points.
- Just so it's clear what we are working with, let's plot these red and blue groups

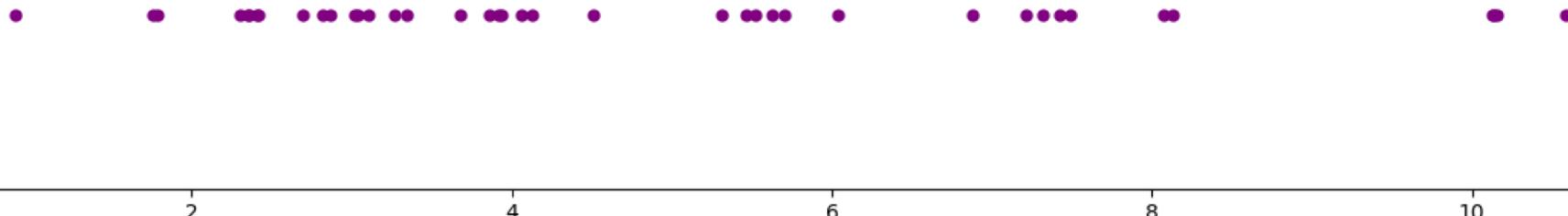
Distribution of red and blue points (known colours)



# EM example

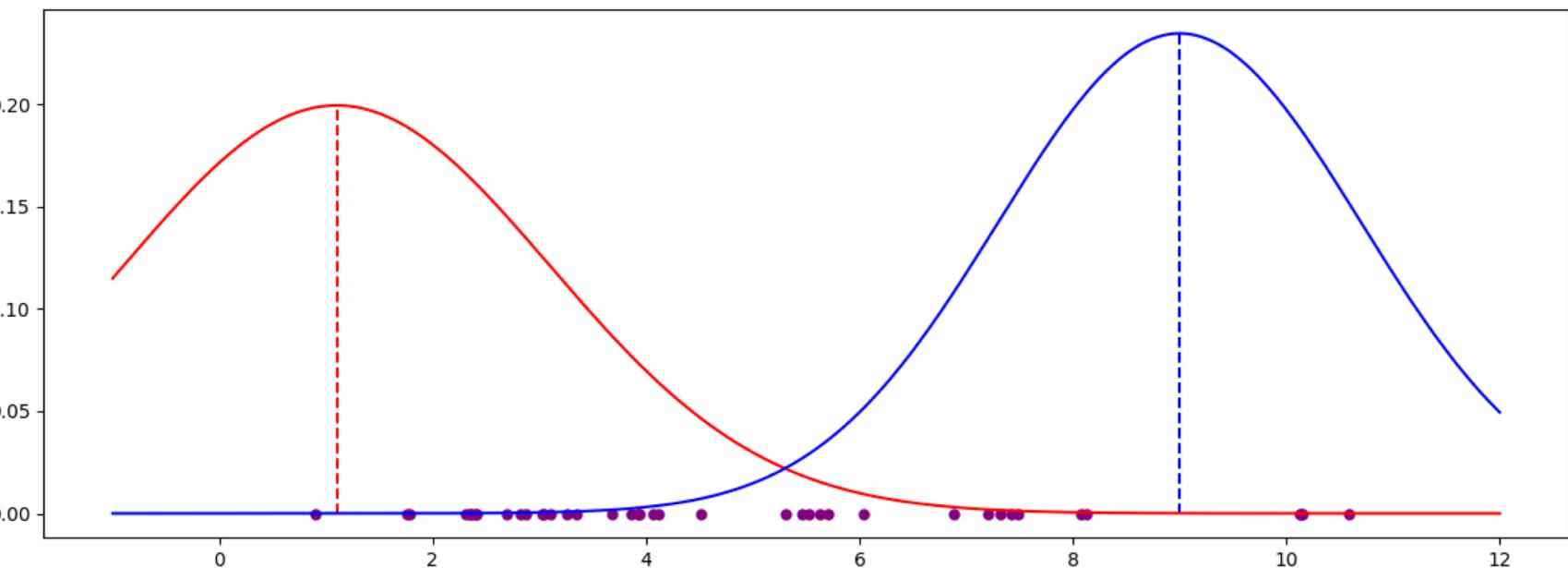
- Now suppose we paint every point purple. Now we have *hidden variables*.
- We know each point is really *either* red or blue, but the actual colour is not known to us. As such, we don't know which values to put into the formulae for the mean and standard deviation.
- How can we estimate the most likely values for the mean and standard deviation of each group now?
- We will use **Expectation Maximization** to find the best estimates for these values.

Distribution of red and blue points (hidden colours)



# EM example

- We know that we have two groups of points, each drawn from a normal distribution
- We also have a likelihood function and we would like to find values for the mean and standard deviation that maximise this function (maximum likelihood estimation)



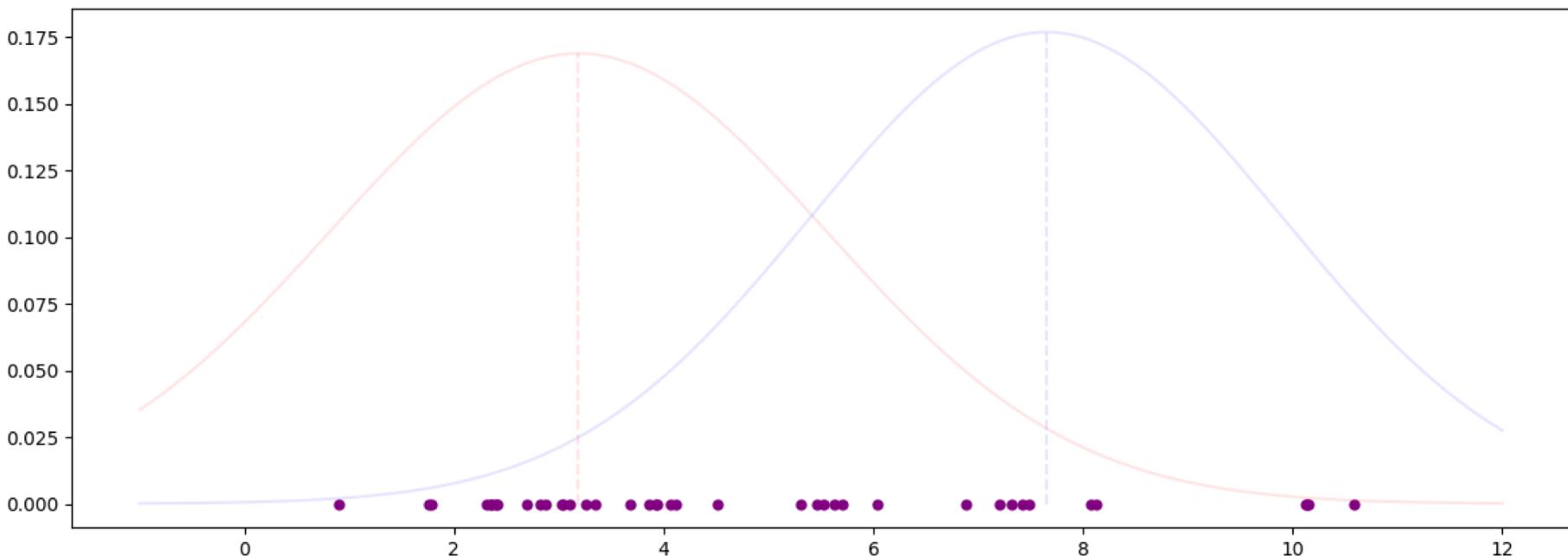
# EM example

1. Start with initial estimates of the mean and standard deviation for the red and blue groups
2. Check how likely each (mean, standard deviation) estimate is to produce each sequence each of the data points (using the likelihood function)
3. Produce a weighting for each (mean, standard deviation) pair for each data point (the **Expectation step**)
  1. These weights will allow us to "rescale" the data points along the axis
4. Use formulae to compute new maximum likelihood estimates of each parameter based on the rescaled data points (the **Maximization step**)
5. Repeat steps 2-4 until each parameter estimate has converged, or a set number of iterations has been reached

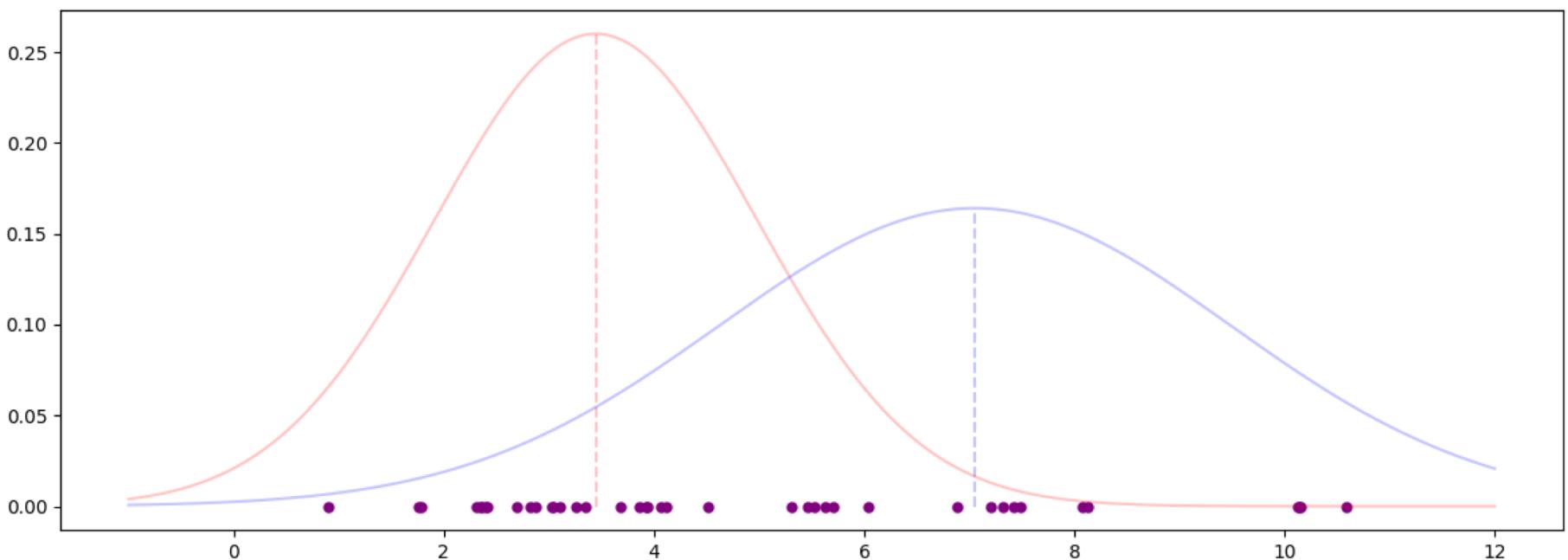
# EM example

This is the first distribution

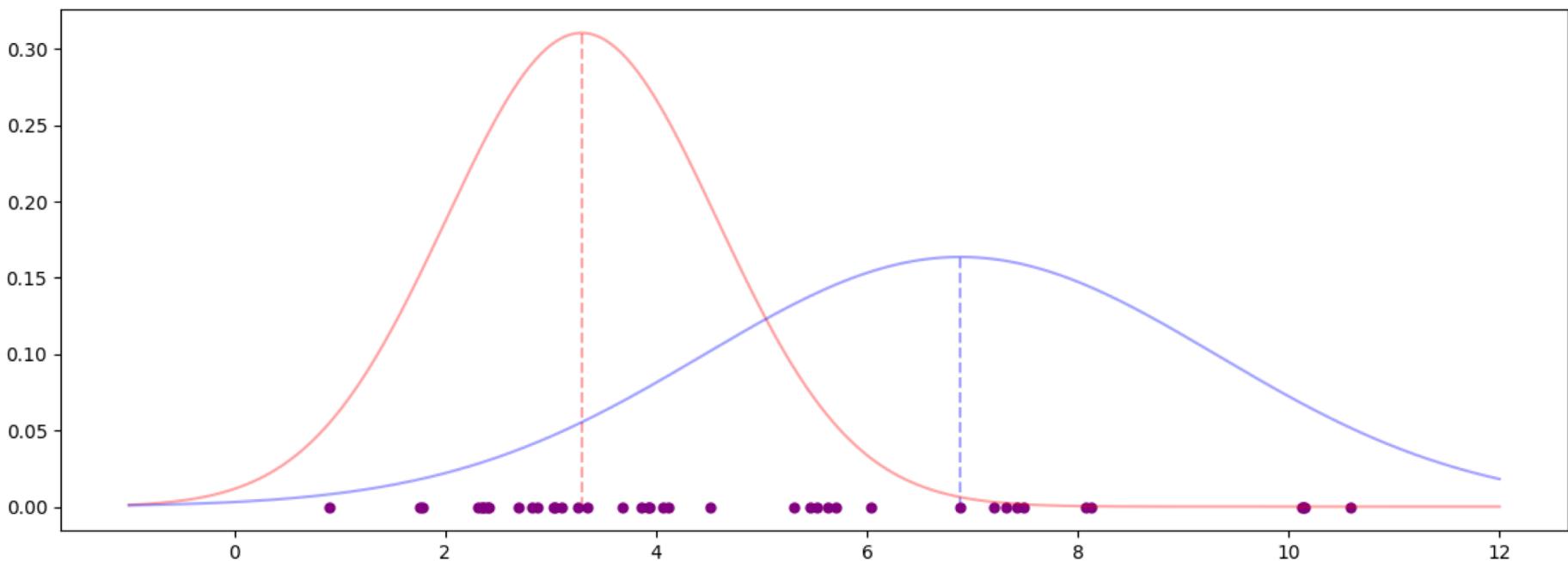
You will see in the next slides how they evolve during different iterations



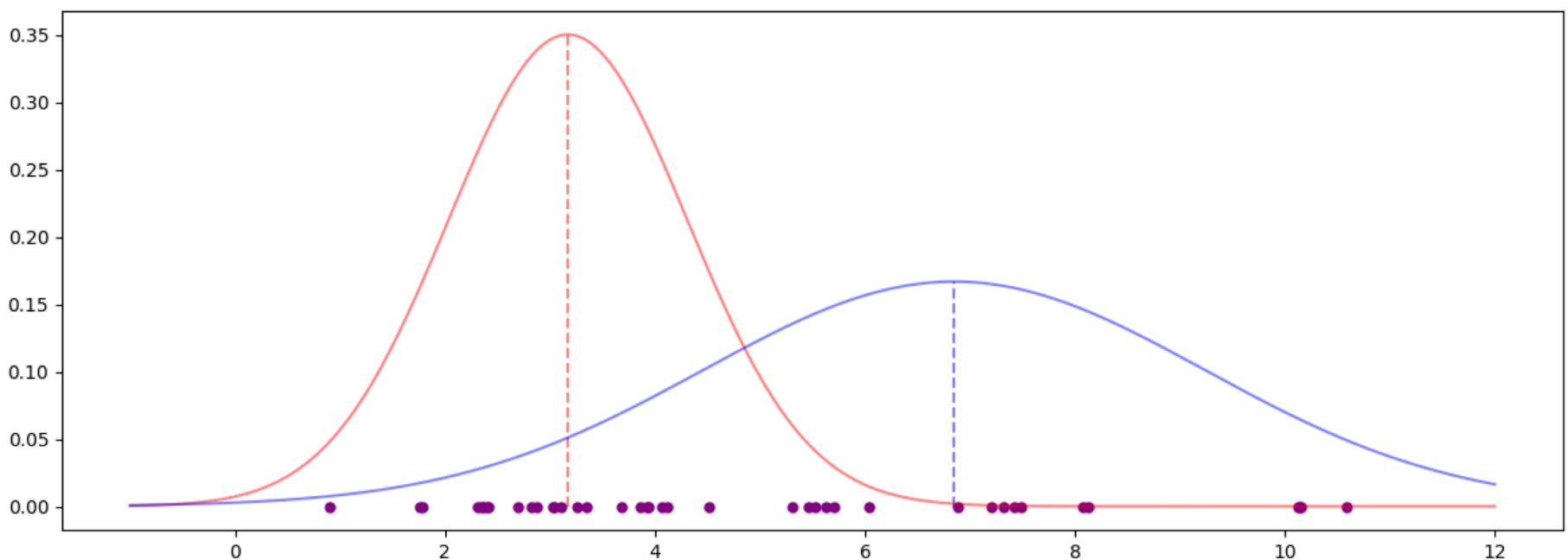
# EM example



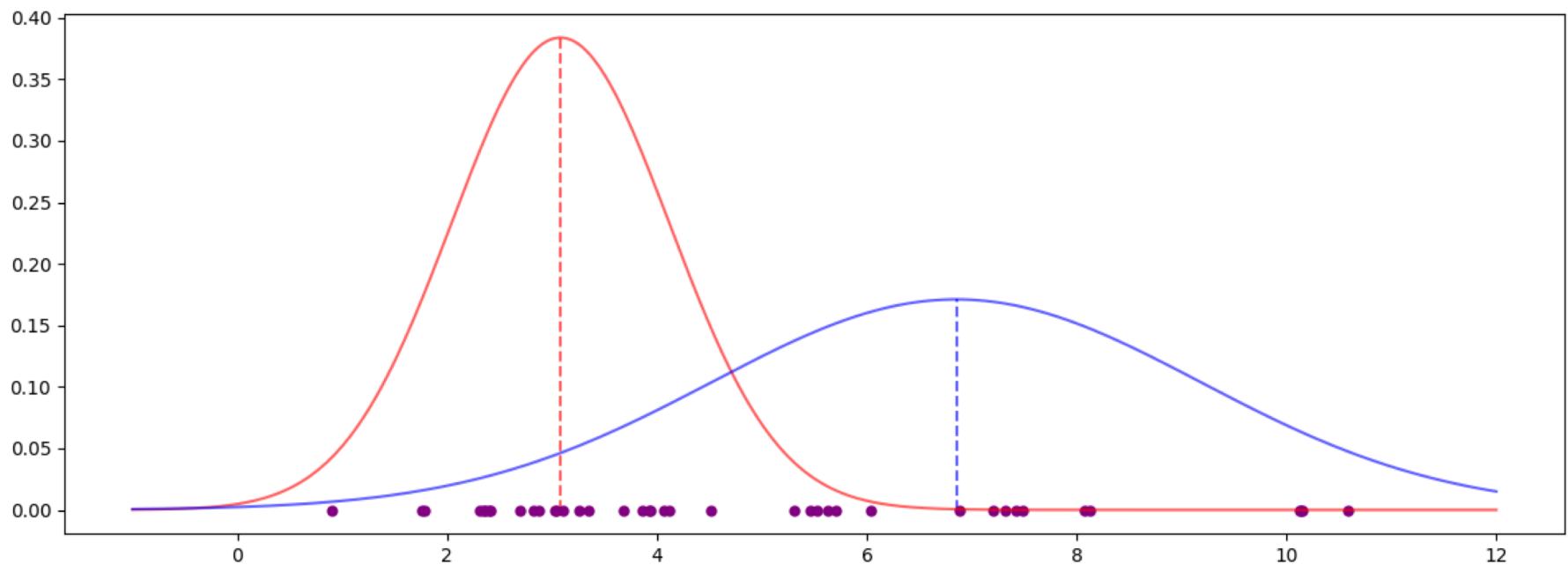
# EM example



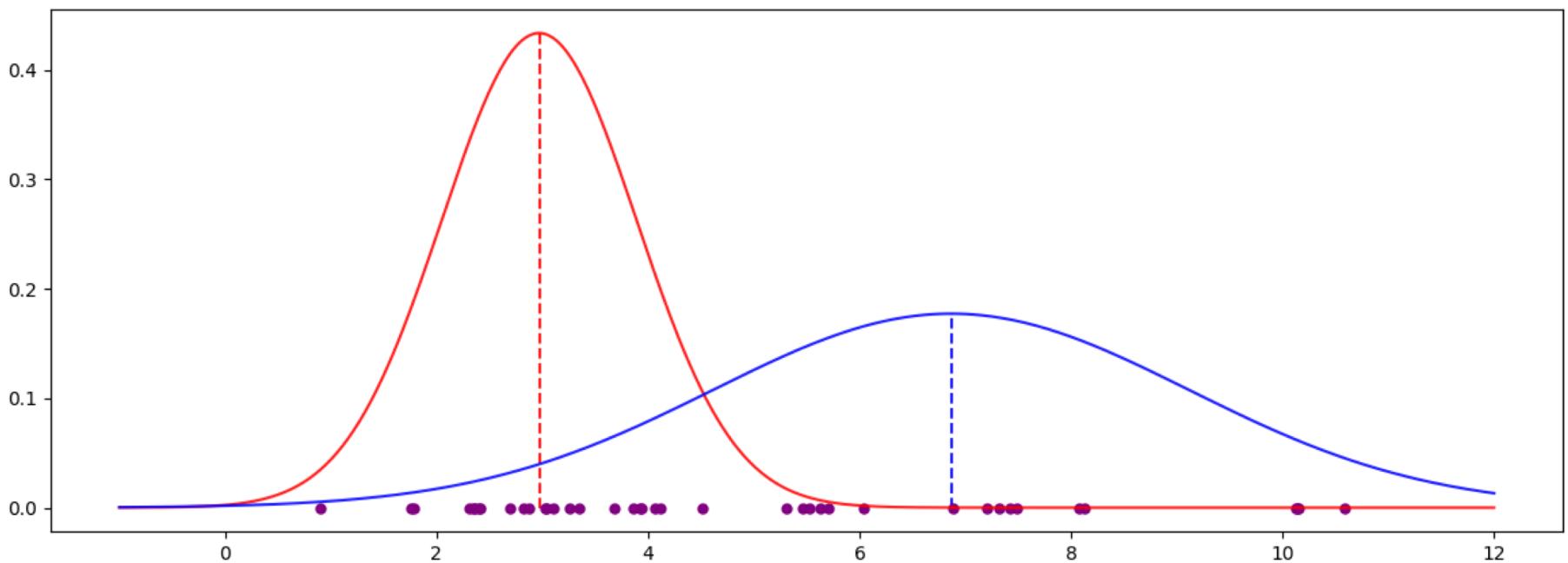
# EM example



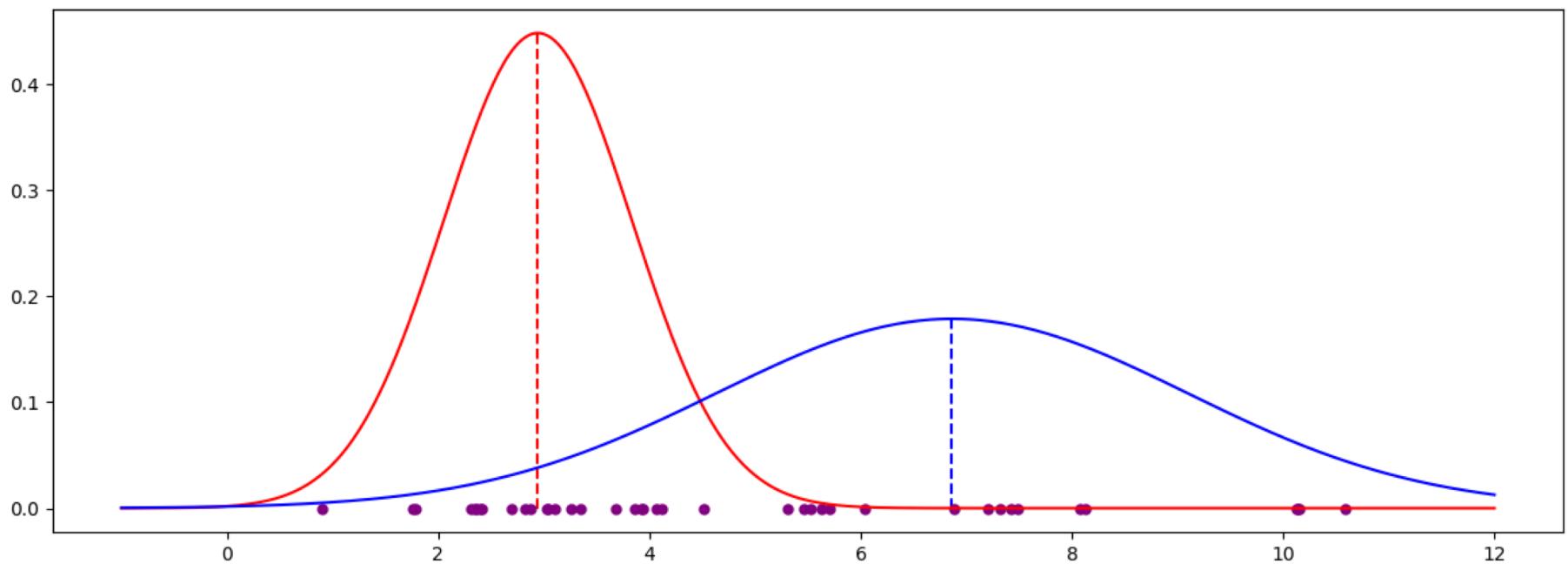
# EM example



# EM example



# EM example



# Summary of clustering

- Clustering groups objects in a cluster if they are similar (or related) to one another and different from (or unrelated to) the objects in other groups
- Clustering can be hard or soft
- There are a variety of approaches, including:
  - Hierarchical clustering: HAC
  - Partitional clustering: k-Means, Fuzzy c-Means
  - Model-based clustering: Expectation Maximization

## Week 3

# Course. Introduction to Machine Learning

## Theory 3. Introduction to unsupervised learning and Cluster Analysis (Part II)

Dr. Maria Salamó Llorente

[maria.salamo@ub.edu](mailto:maria.salamo@ub.edu)

Dept. Mathematics and Informatics,  
Faculty of Mathematics and Informatics,  
University of Barcelona (UB)