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

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,

University of Barcelona (UB)

UNIVERSITAT DE BARCELONA

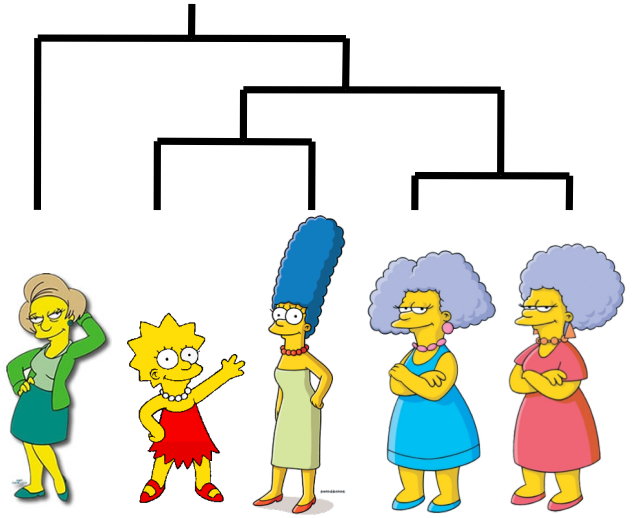
Contents

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)

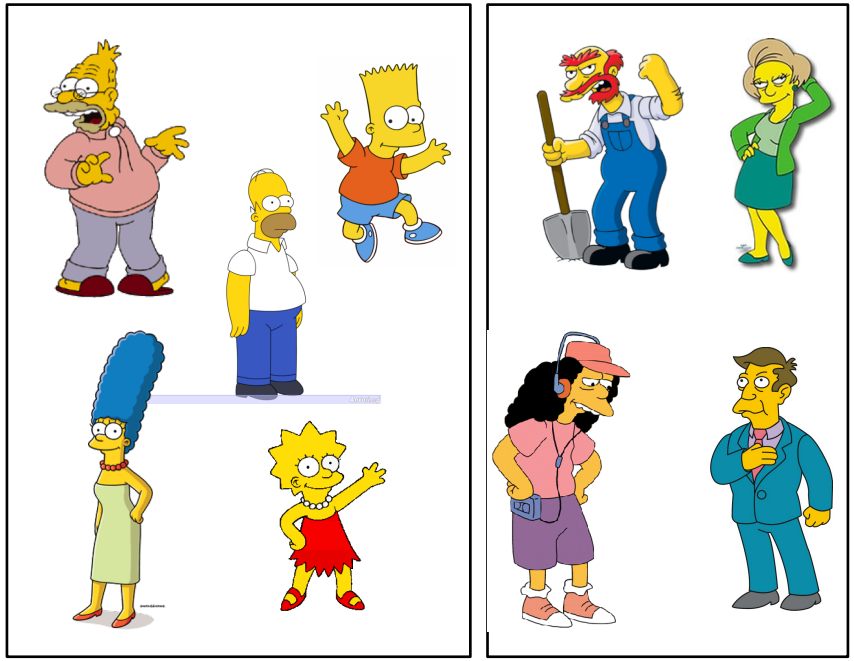
Partitional clustering

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, ***c***:

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

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

Distance Metrics

- **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):

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

UNIVERSITAT DE BARCELONA

K-Means Algorithm

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

1. *Decide on a value for k*
2. *Select k random instances $\{s_1, s_2, \ldots s_k\}$ 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

$s_j = \mu(c_j)$

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

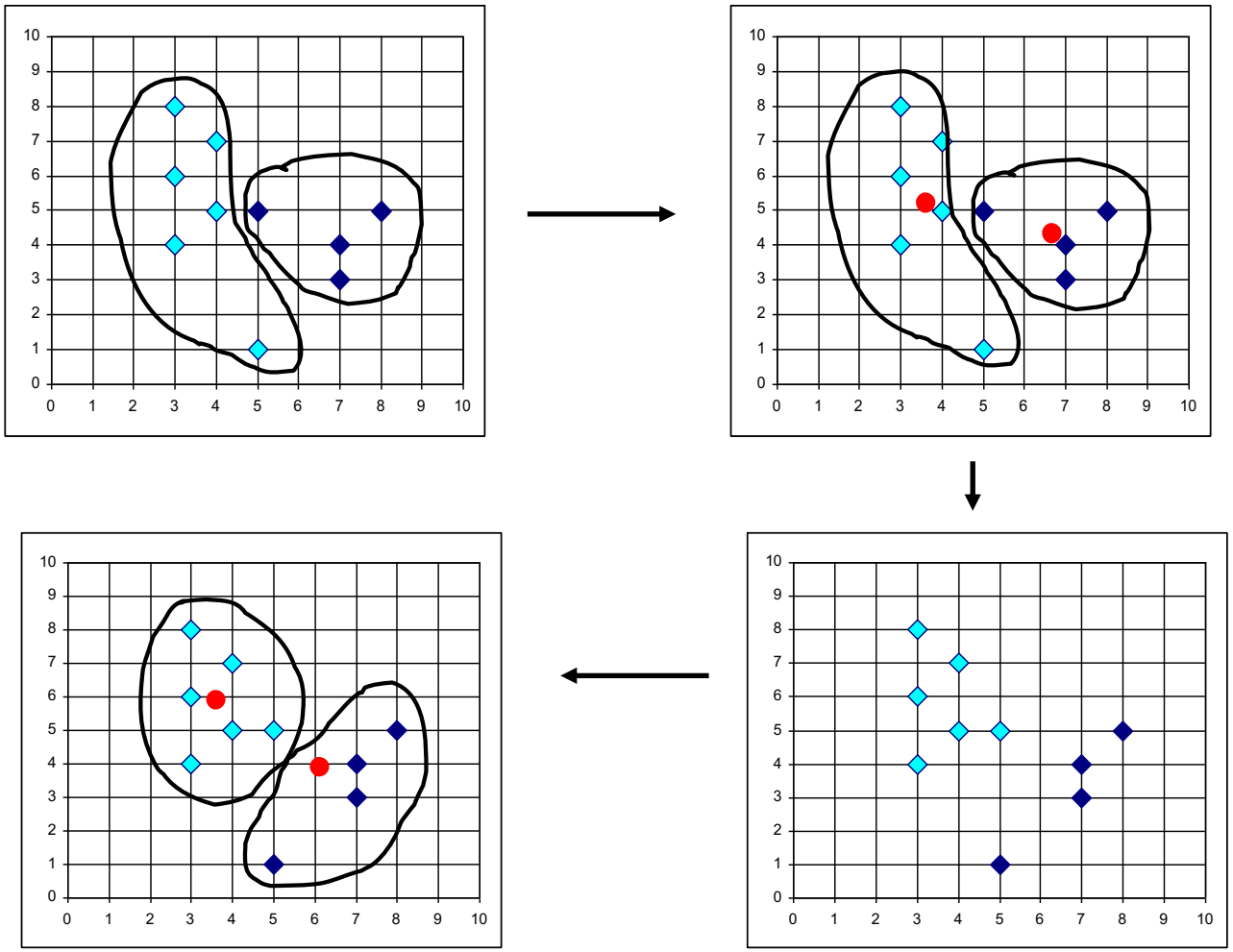
5. *(Until clustering converges or other stopping criterion):*

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 *I* iterations: O(*Iknm*).
- 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 distance of every point to its corresponding cluster** centroid

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

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

Comments on the *K-Means* Method

- **Strength**
  - *Relatively efficient*: *O*(*tkn*), where *n* is # instances, k is # clusters, and *t* is # iterations. Normally, *k*, *t* << *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*

Seed Choice

- 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

Extension of k-Means (towards hierarchical clustering): Bisecting k-Means

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 an 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

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

$$A\colon \mu_A = [\mu_A(x_1)\, \mu_A(x_2) \ldots \mu_A(x_n)]$$

$$\mu_A(x_1) = .5 \quad \mu_A(x_2) = .8 \quad \mu_A(x_3) = .2$$
$$\mu_A(x_4) = .3 \quad \mu_A(x_5) = .2$$

Or equivalently

$$\{x_1, x_2, x_3, x_4, x_5\}$$
$$\boldsymbol{\mu_A = [0.5, 0.8, 0.2, 0.3, 0.2]}$$

$$\{x_1, x_2, x_3, x_4, x_5\}$$
$$\boldsymbol{\mu_B = [0.5, 0.2, 0.8, 0.7, 0.8]}$$

Fuzzy Clustering

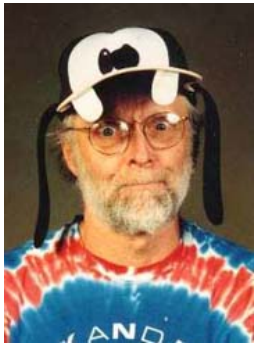
- **"linguistic" terms** like "*much*", "*much more*", "*less*", "*more or less*", "*more than*" and others can use in fuzzy clustering

- 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

Fuzzy C-Means Clustering (FCM)

- 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



Prof. Bezdek

- Several approximations
  - **FCM**: Fuzzy C-Means Clustering (Bezdek, 1981)
  - **PCM**: Possibililistic C-Means Clustering (Krishnapuram - Keller, 1993)
  - **FPCM**: Fuzzy Possibililistic C-Means (N. Pal - K. Pal - Bezdek, 1997)

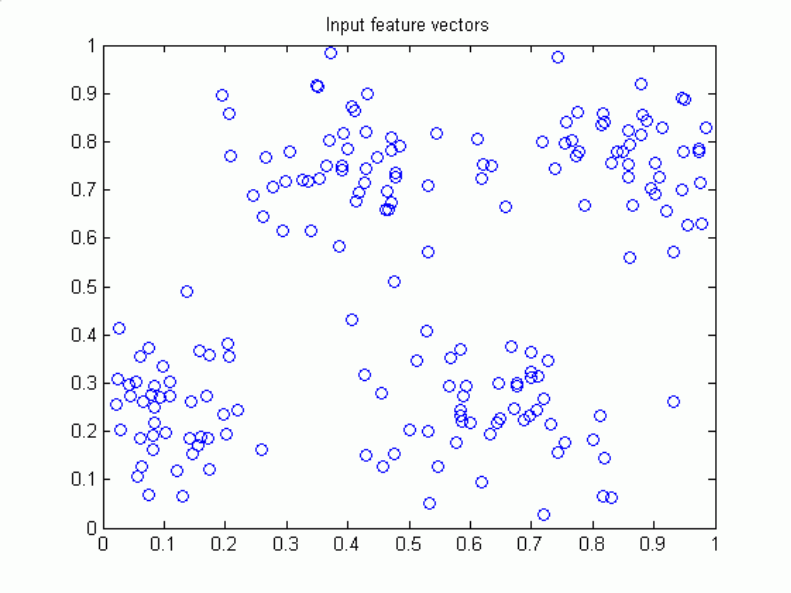
Fuzzy C-Means Clustering

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

Fuzzy C-Means Clustering

features
p = 2

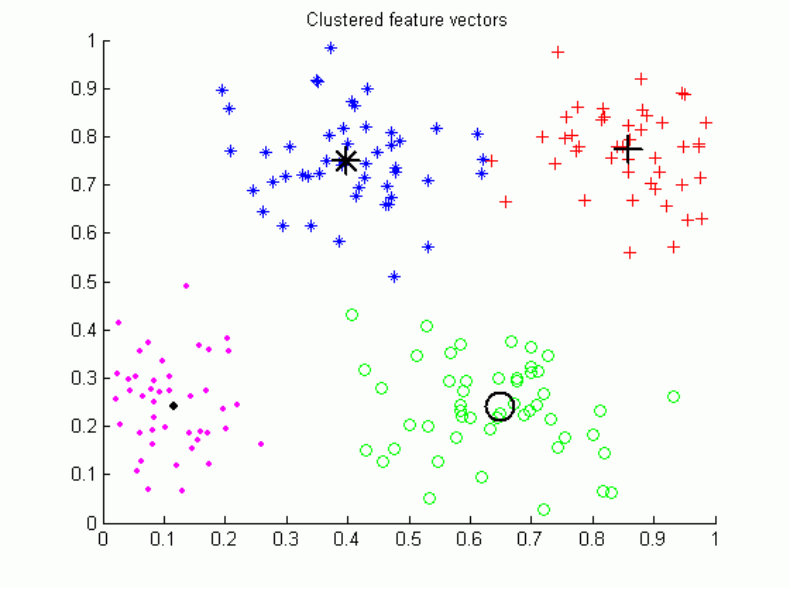
instances
X
n = 188



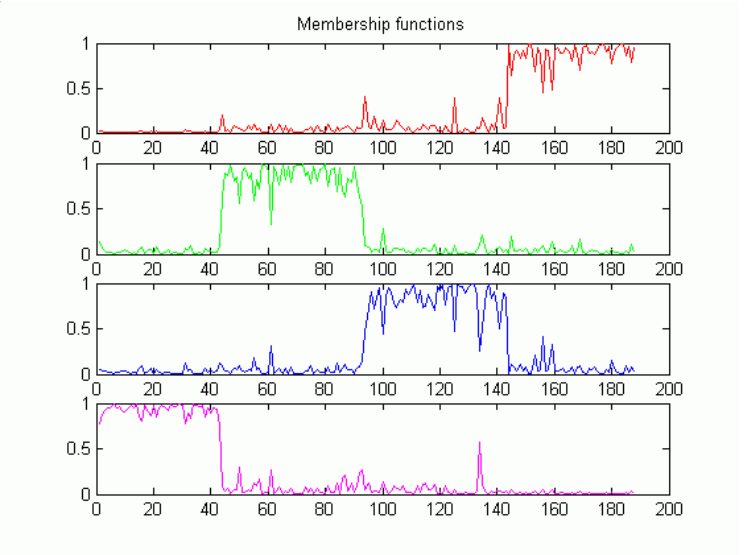
Sample Illustration

U

c = 4
number of clusters



Rows of U (Membership Functions)



Fuzzy C-Means Clustering

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

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

$$1 \le m \le \infty$$

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

- Degree of fuzzification $m \geq 1$

Fuzzy C-Means Clustering

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

$$1 \le m \le \infty$$

- 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 $(1, \infty)$

Fuzzy C-Means Clustering

GOAL: Minimizing Objective Function

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

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

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

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

Note: It is the Center of Gravity

Fuzzy C-Means Clustering

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 $E_t = \| V_t - V_{t-1} \|$, ← 1-norm
  - Termination threshold (Typ. 0.01) (0 < ε)

Fuzzy C-Means Clustering

Iterative FCM algorithm

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

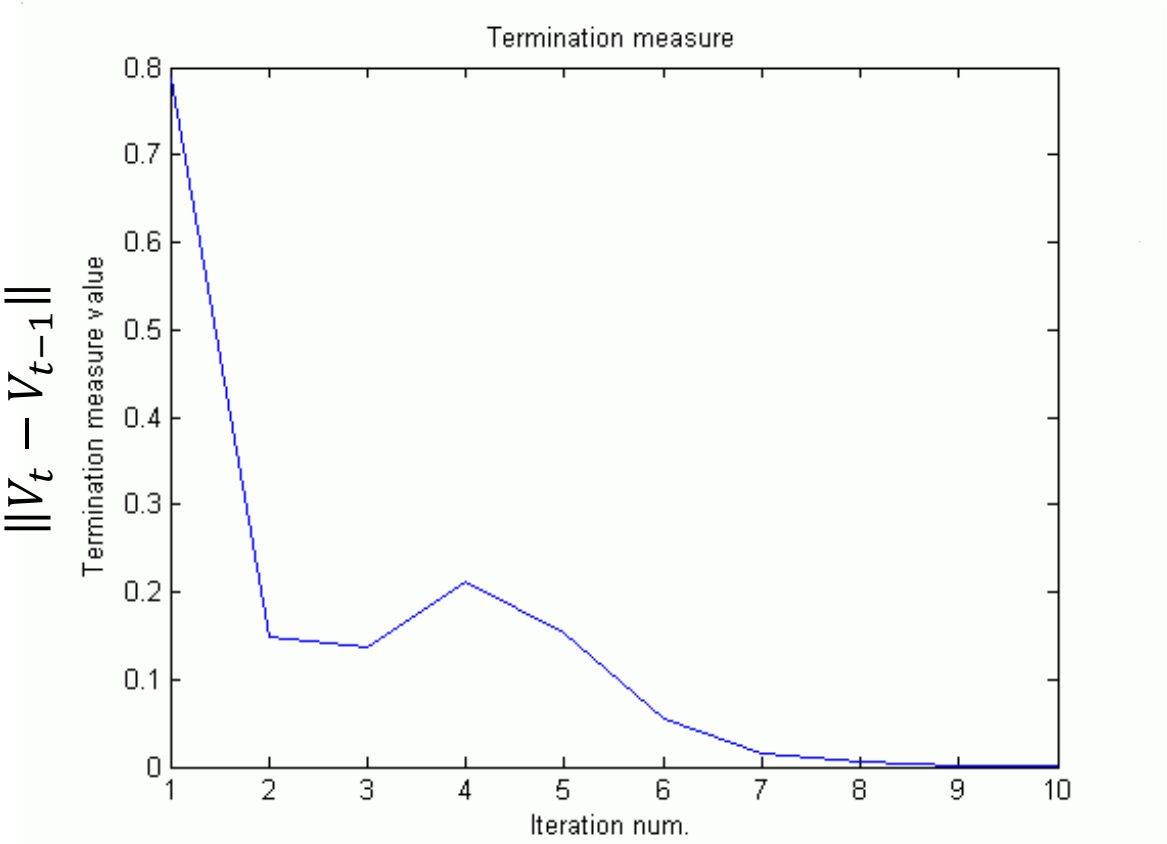
```
t ← 0
REPEAT
    t ← t +1
    Compute matrix U_t   (Eq.1)
    Compute associated clusters centers V_t (Eq.2)
UNTIL ( t = T   or ‖ V_t − V_{t−1} ‖ ≤ ε )
(U,V) ← (U_t,V_t)
```

Fuzzy C-Means Clustering

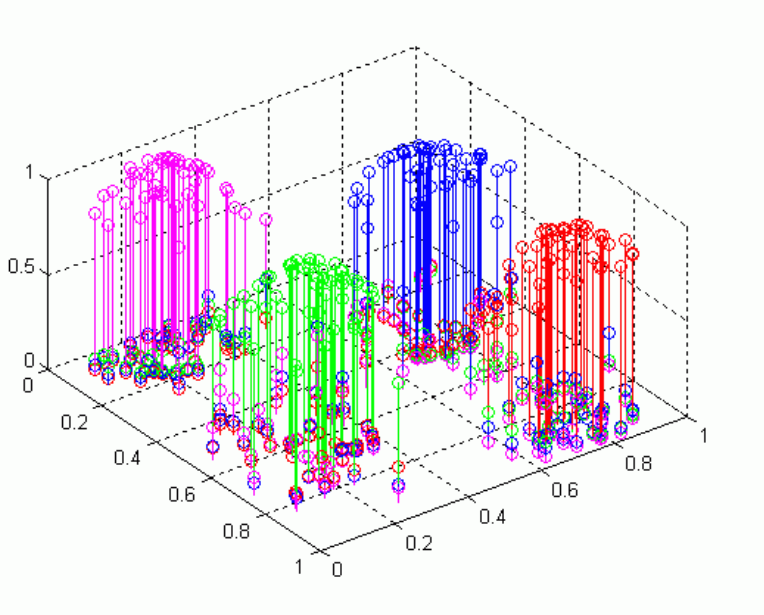
Sample Termination Measure Plot

Termination Measure Values

$m = 2.0$



Final Membership Degrees



Note that when advancing iterations, the error is decreasing

Example Fuzzy clusters to Crisp clusters

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

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

Cl_1 : [0.6 0.7 0.3 0.1 0.4 0.2 0.1]
Cl_2 : [0.1 0.1 0.3 0.5 0.1 0.7 0.1]
Cl_3 : [0.3 0.2 0.4 0.4 0.5 0.1 0.8]

Example Fuzzy clusters to Crisp clusters

Fuzzy Clusters

Cl_1 : [0.6 0.7 0.3 0.1 0.4 0.2 0.1]
Cl_2 : [0.1 0.1 0.3 0.5 0.1 0.7 0.1]
Cl_3 : [0.3 0.2 0.4 0.4 0.5 0.1 0.8]

Crisp Clusters from Fuzzy Clusters

Cl_1 : [1 1 0 0 0 0 0]
Cl_2 : [0 0 0 1 0 1 0]
Cl_3 : [0 0 1 0 1 0 1]

Example Fuzzy clusters to Crisp clusters

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

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

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

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

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

$$Cl_3 : \{\; x_3 \quad x_5 \quad x_7 \;\}$$

Crisp Clusters

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

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

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

Fuzzy C-Means Clustering

- **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)

Fuzzy C-Means Clustering

Optimal Number of Clusters

- Performance Index

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

Average of all feature vectors

$$\overline{x} = \frac{1}{n} \sum_{k=1}^{n} x_k$$

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

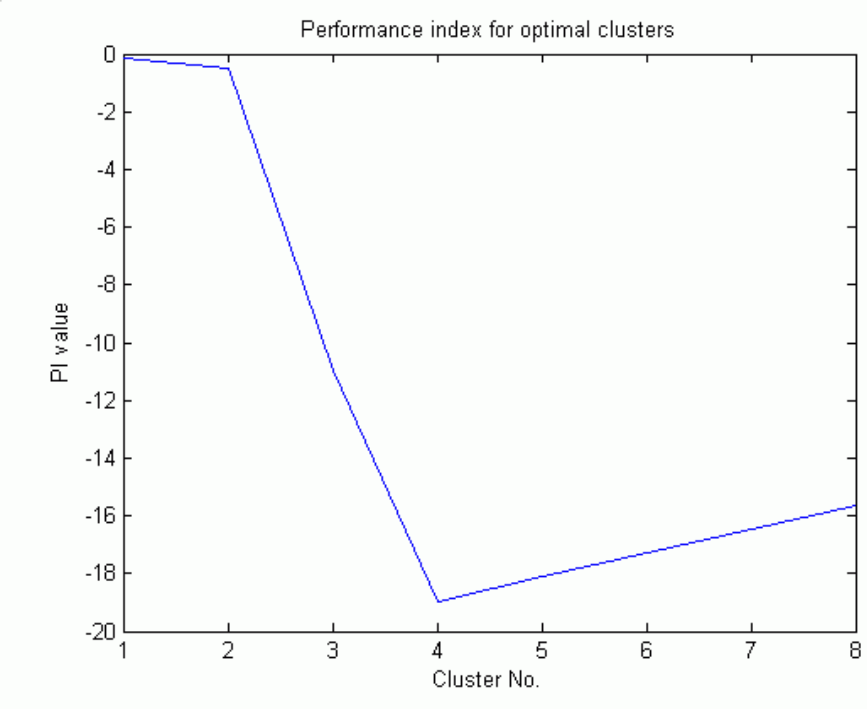
Sum of the within fuzzy cluster fluctuations (small value for optimal c)

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

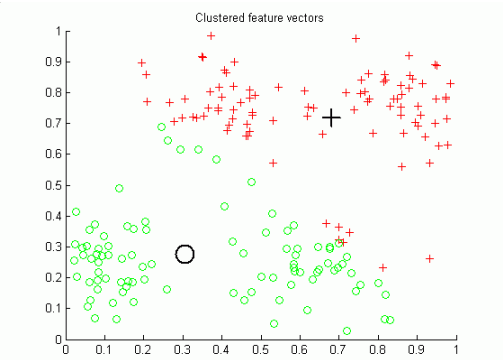
Fuzzy C-Means Clustering

Optimal Cluster No. (Example)

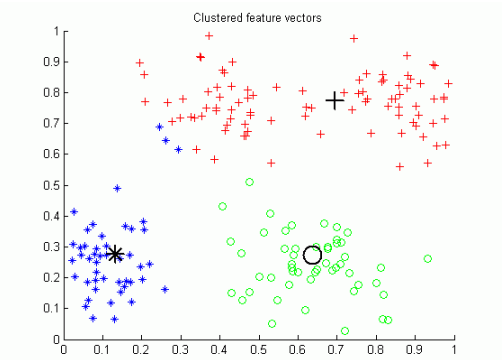
Performance index for optimal clusters (is minimum for c = 4)



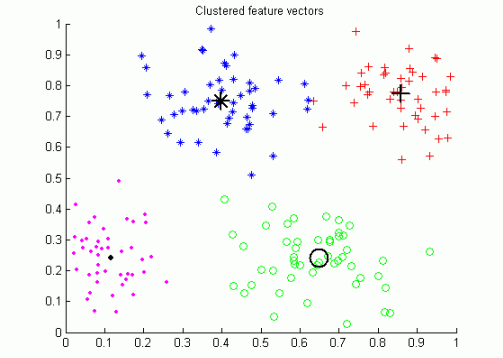
c = 2



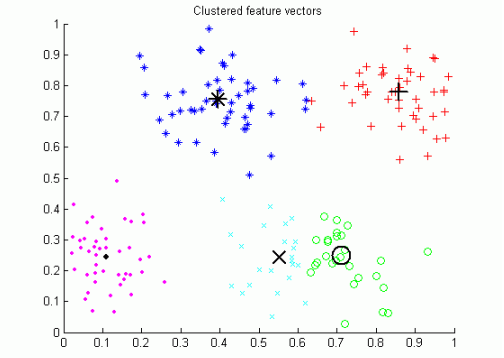
c = 3



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

UNIVERSITAT DE BARCELONA

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: 32-57J.*
- ***C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", 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-Maximitzation (EM)

Look at the video

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

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

Applications of EM

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

Expectation Maximization

- **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

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

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)

Expectation Maximization (EM)

- Probabilistic method for soft clustering
  - It uses probabilities instead of distances!
- Direct method that assumes *k* clusters: $\{c_1, c_2, \ldots c_k\}$
- 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 θ 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, θ, 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: $\Sigma_i log(p_A Pr(x_i|A) + p_B Pr(x_i|B))$

Estimating parameters from weighted objects:

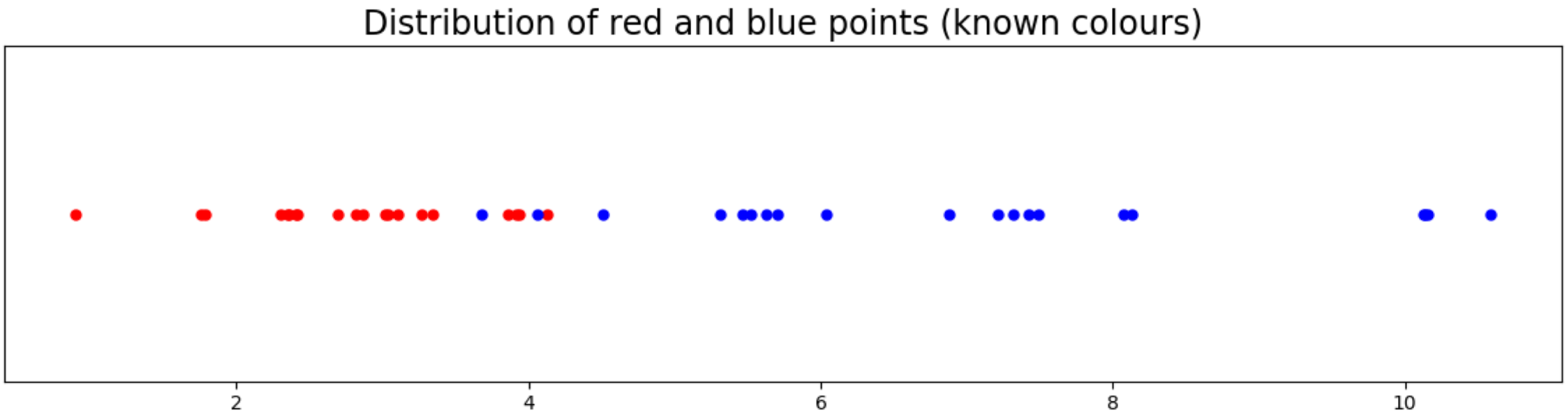
Mean of cluster A: $\mu_A = \frac{w_1 x_1 + w_2 x_2 + ... + w_n x_n}{w_1 + w_2 + ... + w_n}$

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

UNIVERSITAT DE BARCELONA

EM example

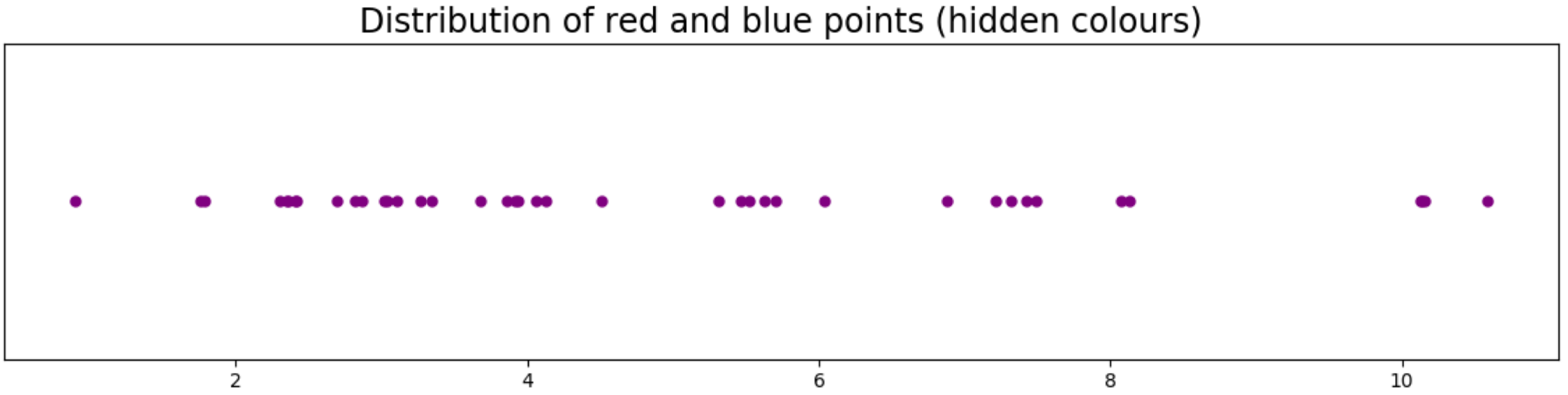
- Suppose we are given two sets of samples, red and blue, drawn from two different normal distributions.
- Our goal will be to find the mean and standard deviation for each group of points.
- Just so it's clear what we are working with, let's plot these red and blue groups



UNIVERSITAT DE BARCELONA

EM example

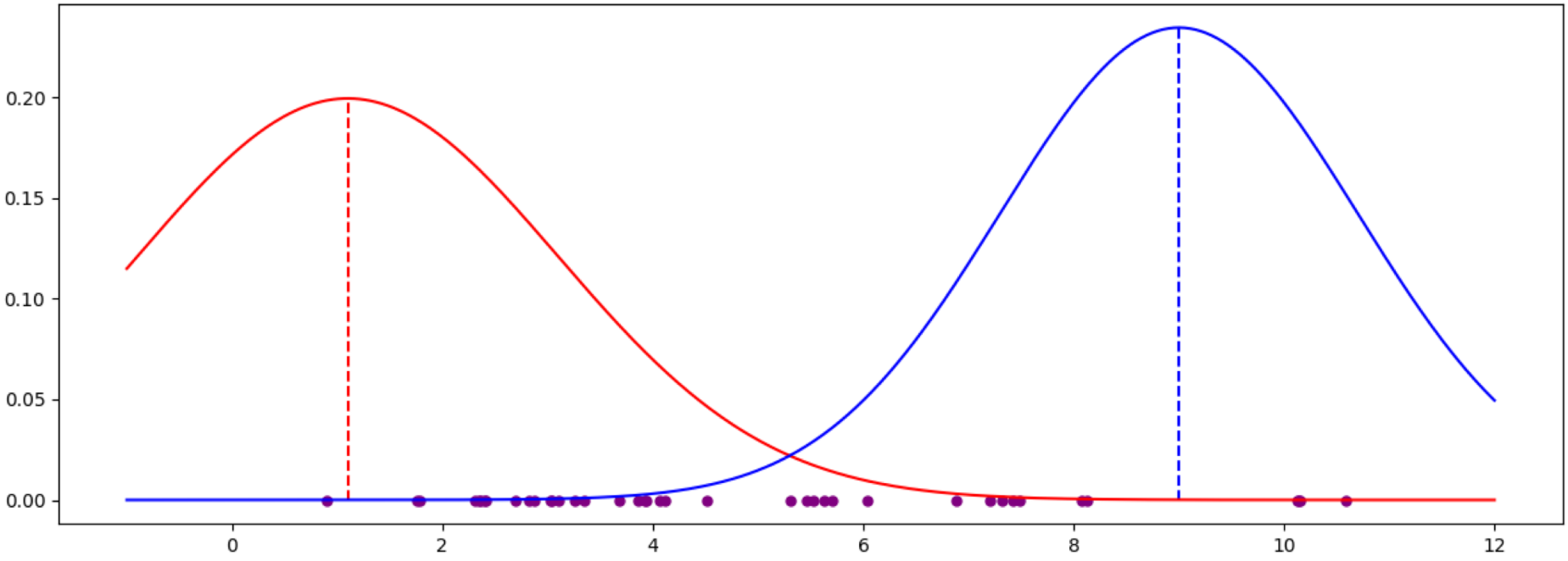
- Now suppose we paint every point purple. Now we have *hidden variables*.
- We know each point is really *either* red or blue, but the actual colour is not known to us. As such, we don't which values to put into the formulae for the mean and standard deviation.
- How can we estimate the most likely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



UNIVERSITAT DE BARCELONA

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

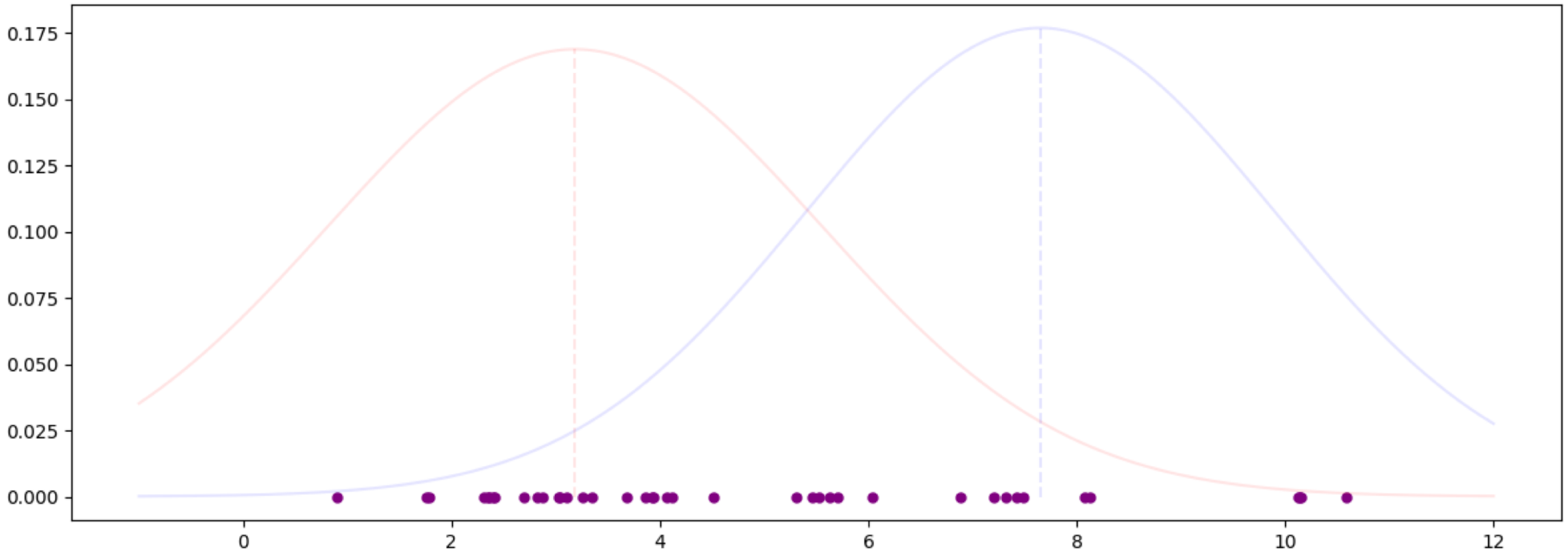
1. Start with initial estimates of the mean and standard deviation for the red and blue groups
2. Check how likely each (mean, standard deviation) estimate is to produce each sequence each of the data points (using the likelihood function)
3. Produce a weighting for each (mean, stand deviation) pair 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 **Maximisation step**)
5. Repeat steps 2-4 until each parameter estimate has converged, or a set number of iterations has been reached

UNIVERSITAT DE BARCELONA

EM example

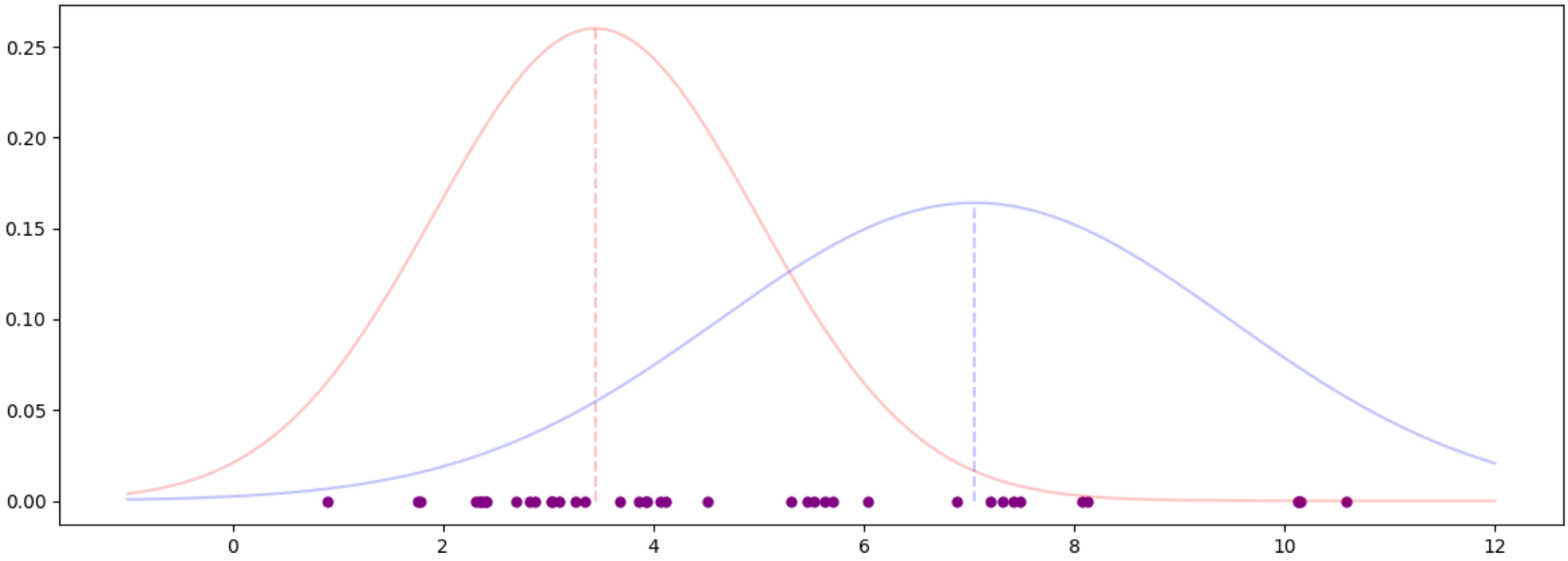
This is the first distribution

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

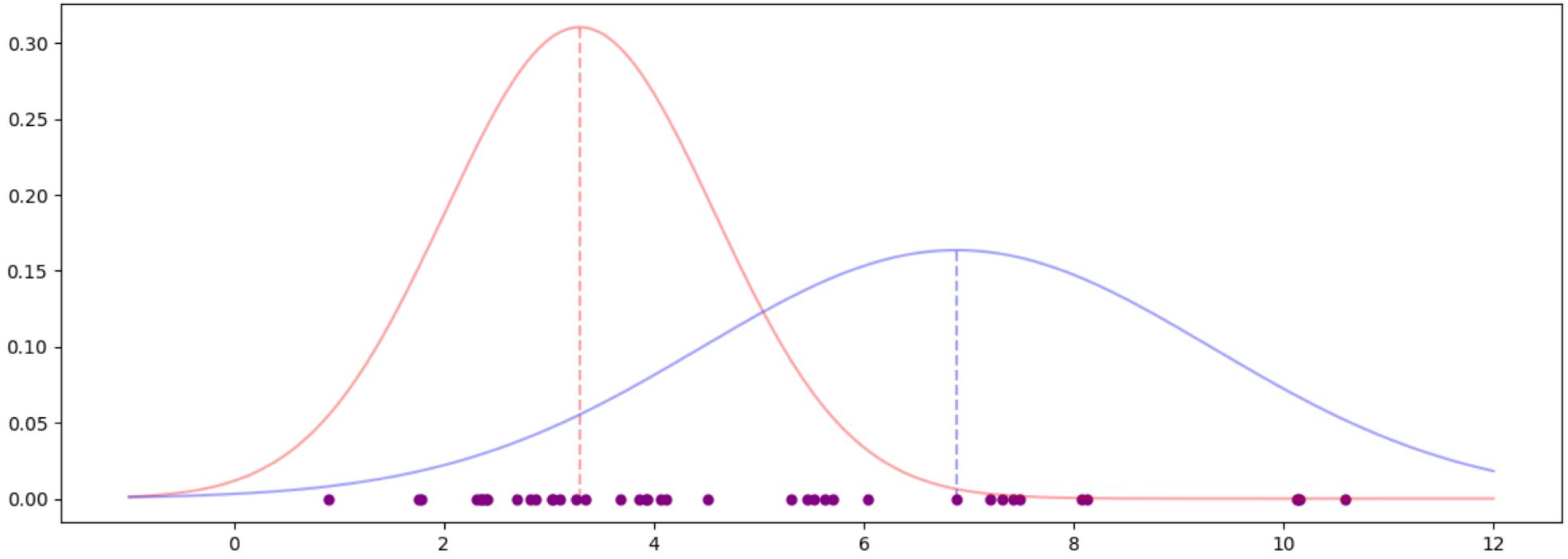


UNIVERSITAT DE BARCELONA

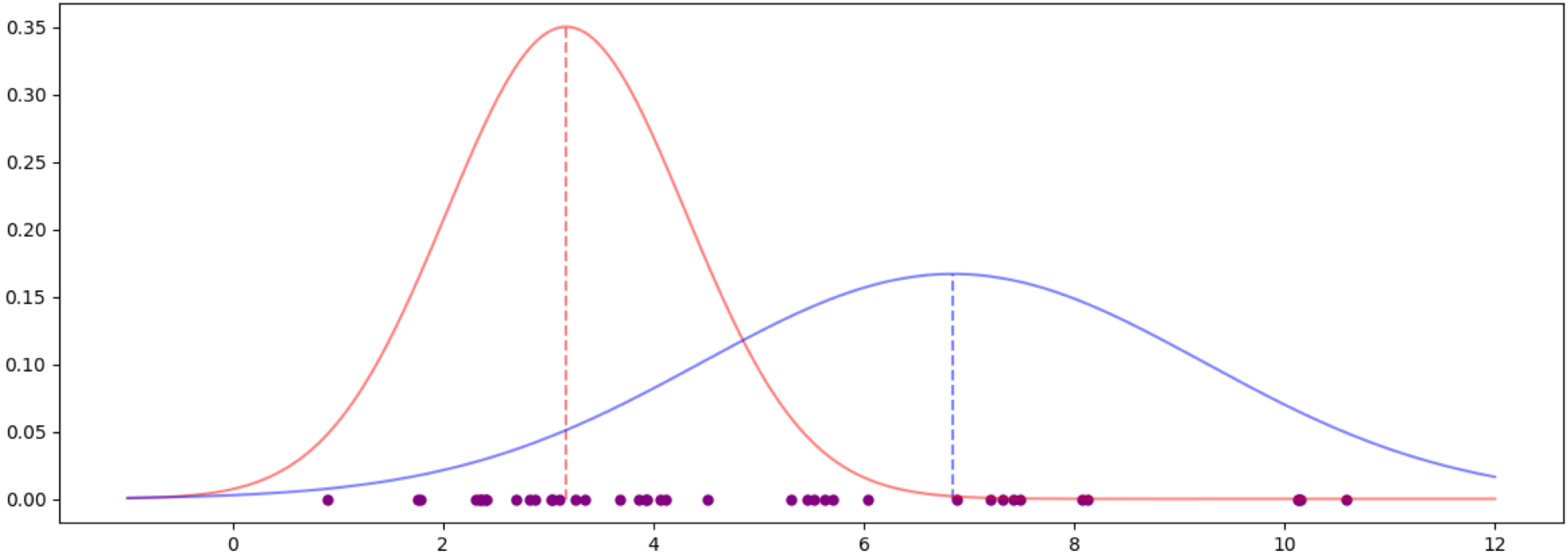
EM example



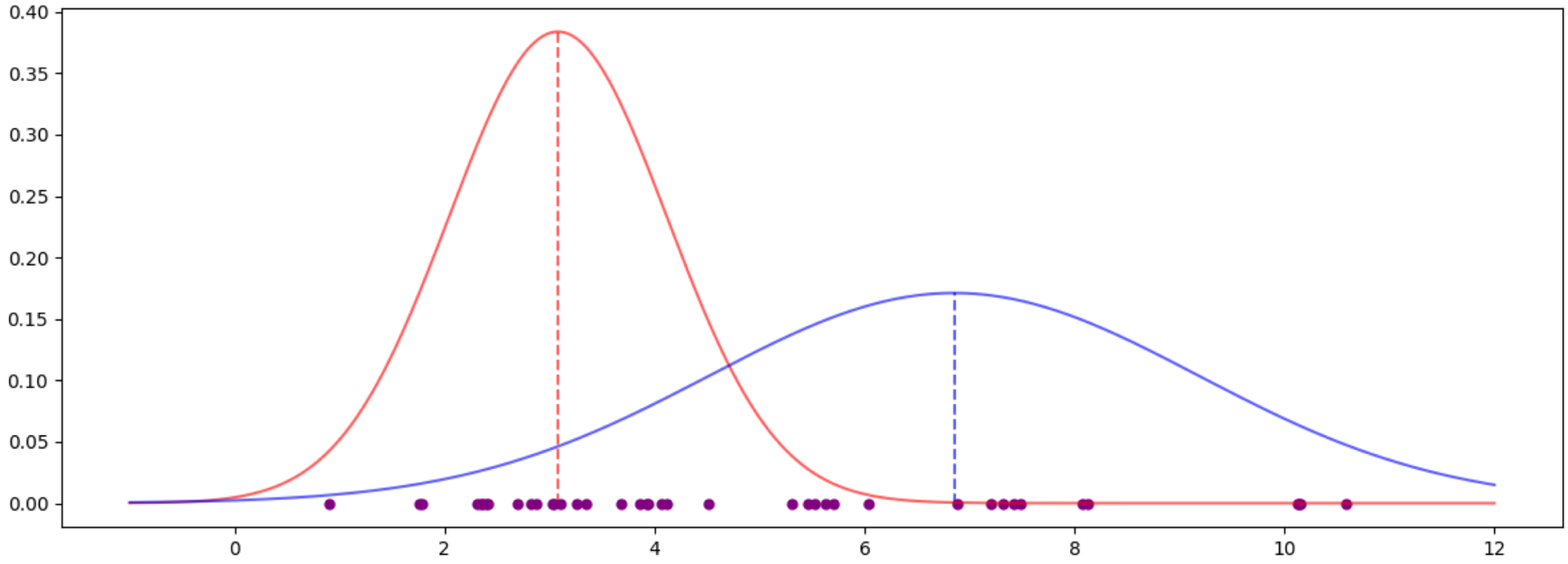
EM example



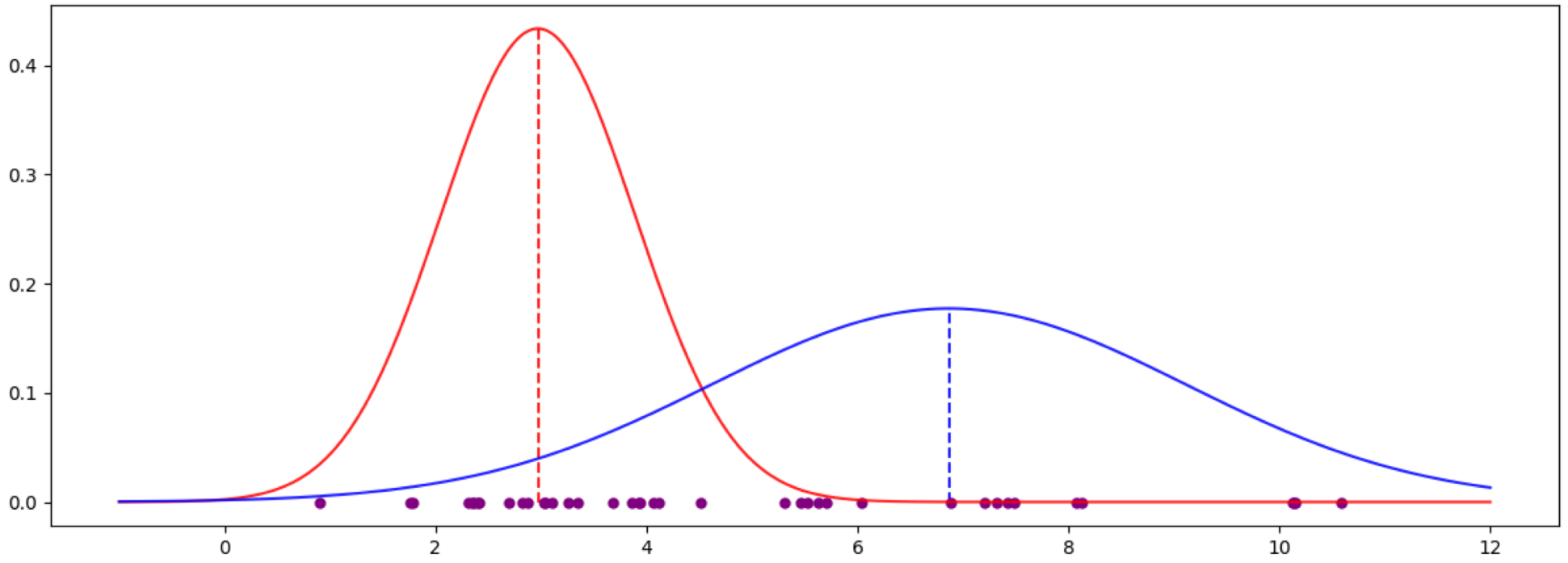
EM example



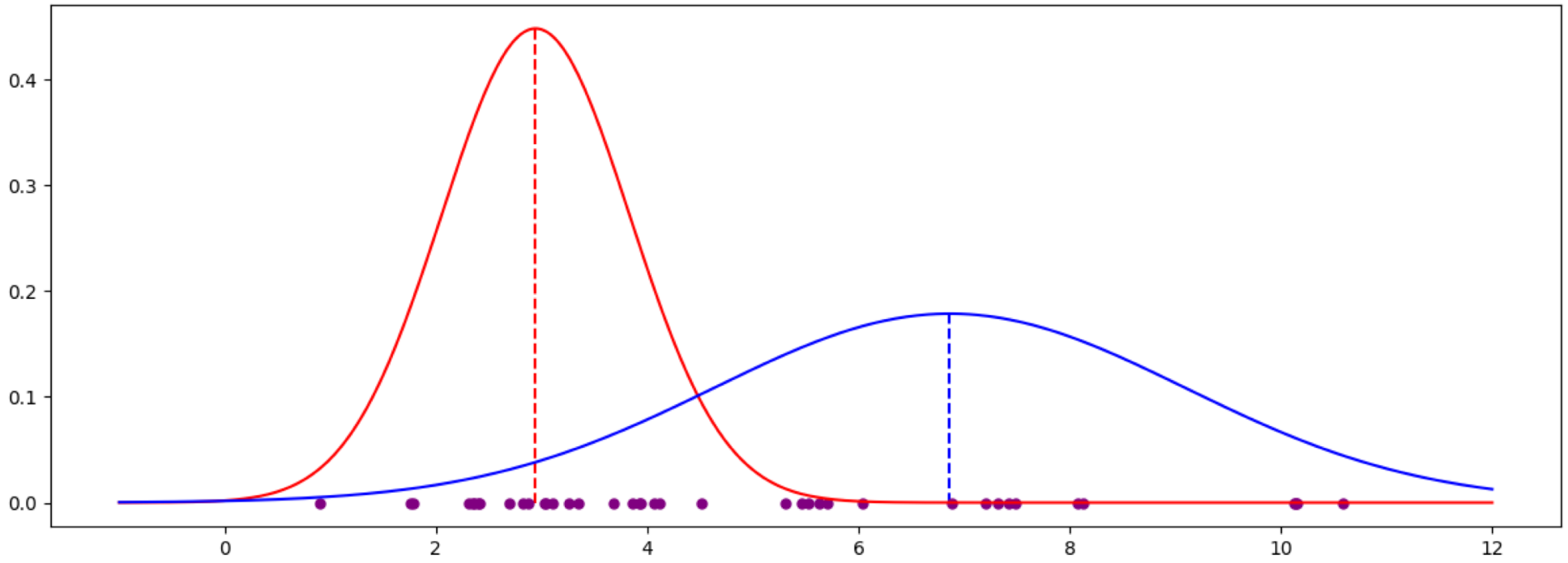
EM example



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

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,

University of Barcelona (UB)