Data Science
Expectation-Maximization for Density Modelling

Latent factor estimation

Stéphane Marchand-Maillet

Department of Computer Science

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES

VIPER

Master en Sciences Informatiques - Autumn semester

Table of contents

What is the lecture about?

- ⋆ Understand unsupervised conditional modeling as latent factor estimation
- ⋆ Develop the example of density estimation by Gaussian Mixture models (GMM)
- ⋆ Practice Maximum Likelihood Estimation (MLE)
- ⋆ Understand the alternating principle of Expectation-Maximization for the discovery of latent factors

Reading: [1] (chap 9) and [2] (chap 8.7)

Data modeling

- ⋆ Up until now, we have used an implicit model for the data
  - ◦ Component models → Variance as a criterion on centered data (Normal distribution)
  - ◦ Discriminant models → Variance of projected data for within- and between-class models

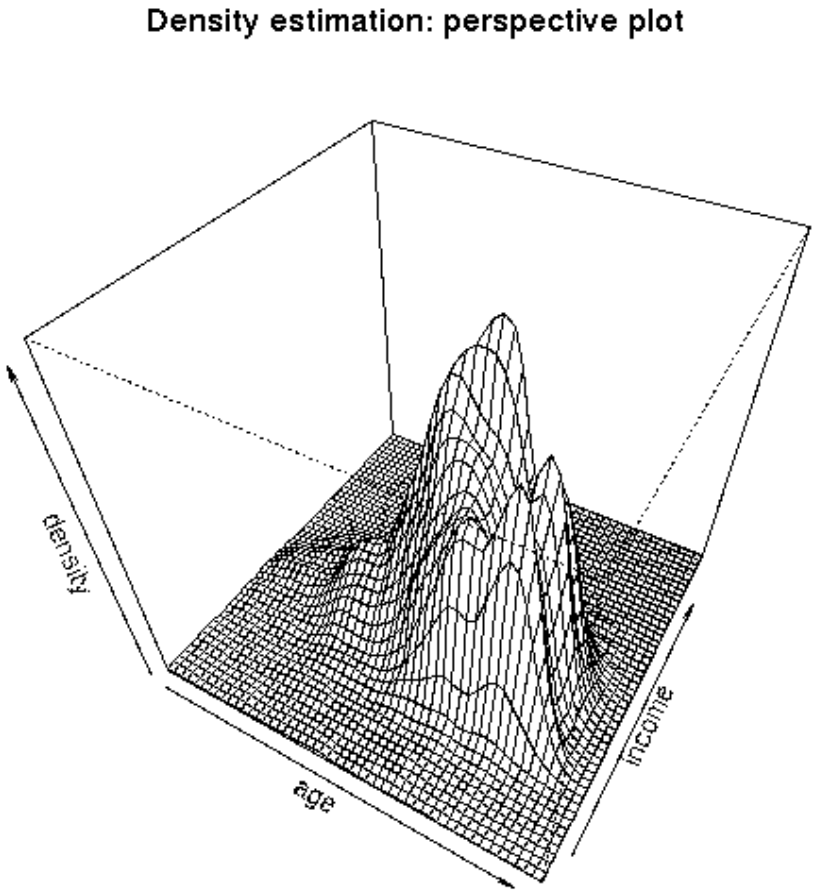
⇒ The distribution is fixed (essentially normal) and we look for its parameters $\boldsymbol{\theta}$ (e.g $\boldsymbol{\theta} = [\boldsymbol{\mu}, \Sigma]$)

- ⋆ Alternatively we can search a model for data density
- ⋆ Let $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) : \Omega \to \mathbb{R}$ be the data density

Density estimation

Methods

- Nearest neighbor methods (kNN)
- Parzen windows, RBF networks
- Histograms
- Mixture models



Mixture models

Definition

⋆ The density $\boldsymbol{f}(\boldsymbol{x})$ is generated by c "basis" functions ϕ (components), from a family $\mathbb{F}$

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{c} \pi_j \phi(\boldsymbol{x}, \boldsymbol{\theta}_j)$$

⋆ $\pi_j \in \mathbb{R}$ are the mixture parameters

⋆ $\phi(\boldsymbol{x}, \boldsymbol{\theta}_j) \in \mathbb{F}$ are functions controlled by parameters $\boldsymbol{\theta}_j$

Assumptions

1. The number of components ($c \in \mathbb{N}^*$) is known
2. The family of functions $\mathbb{F}$ is known
3. Labels (class labels) are unknown

Probabilistic reading

⋆ Density $\boldsymbol{f}(\boldsymbol{x})$ represents a random process where $\boldsymbol{x}$ is drawn from a set of states ω_j with prior probability $\mathbb{P}(\omega_j)$

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{c} \pi_j \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{\theta}_j)$$

$$\boldsymbol{f}(\boldsymbol{x}) = \mathbb{P}(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{j=1}^{c} \mathbb{P}(\omega_j)\, \mathbb{P}(\boldsymbol{x}|\omega_j, \boldsymbol{\theta}_j)$$

We get (by identification):

⋆ $\pi_j = \mathbb{P}(\omega_j)$ so that $\sum_j \pi_j = 1$

⋆ $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{\theta}_j) = \mathbb{P}(\boldsymbol{x}|\omega_j, \boldsymbol{\theta}_j)$ is the conditional probability that $\boldsymbol{x}$ is generated by state ω_j

Reminder: Maximum log-likelihood (MLE)

- ⋆ Given $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ unlabeled samples generated by the mixture $\boldsymbol{f}(\boldsymbol{x}) = \mathbb{P}(\boldsymbol{x}|\boldsymbol{\theta})$.
- ⋆ $\boldsymbol{\theta} \overset{\text{here}}{=} [\pi_j, \boldsymbol{\theta}_j]_{j \in [\![c]\!]}$ are the parameters to infer
- ⋆ Likelihood :

$$\mathbb{P}(\mathcal{X}|\boldsymbol{\theta}) \overset{\text{i.i.d}}{=} \prod_i^N \mathbb{P}(\boldsymbol{x}_i|\boldsymbol{\theta})$$

⇒ Likelihood estimate : $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} \mathbb{P}(\mathcal{X}|\boldsymbol{\theta})$

- ⋆ Log-likelihood

$$\mathbb{L}(\boldsymbol{\theta}, \mathcal{X}) = \sum_{i=1}^N \log \mathbb{P}(\boldsymbol{x}_i|\boldsymbol{\theta}) \overset{\text{here}}{=} \sum_{i=1}^N \log \left[\sum_{j=1}^c \pi_j \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \right]$$

⇒ Maximum log-likelihood estimate (MLE): $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} \mathbb{L}(\boldsymbol{\theta}, \mathcal{X})$

Gaussian mixture

ϕ is a probability density function $\rightarrow$ choosing the (agnostic) normal density family as basis ($\phi \in \mathbb{F} = \{\mathcal{N}(\mu, \Sigma)\}$) seems reasonable:

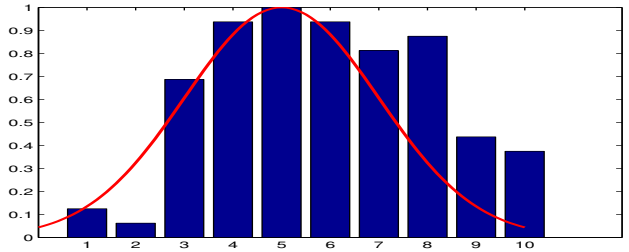
$$f(x) = \sum_{j=1}^{c} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j) \quad \text{i.e} \quad \phi_j(x) := \phi(x, \theta_j) = \mathcal{N}(x|\mu_j, \Sigma_j)$$

$\rightarrow$ Can approximate any density

$\rightarrow$ Enables a linear system of its parameters when maximizing the log-likelihood (MLE)

Basic case : 1 component, $c = 1$

$\star$ $\boldsymbol{\theta} = [\boldsymbol{\mu}, \Sigma]$



$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_i \log e^{-(\boldsymbol{x}_i - \boldsymbol{\mu})^\mathsf{T} \Sigma^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})}$$

$$\Longleftrightarrow$$

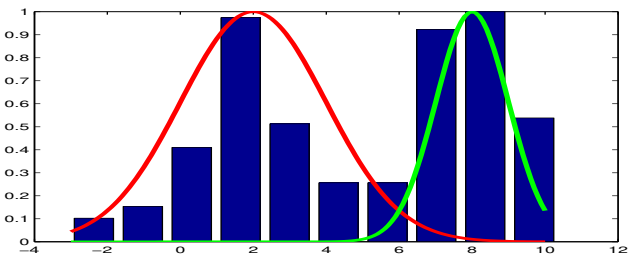
$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})^\mathsf{T} \Sigma^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})$$

$\Rightarrow$ $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_i \boldsymbol{x}_i$ ✆

$\Rightarrow$ $\hat{\Sigma} = \frac{1}{N} \sum_i (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^\mathsf{T}$ ✆

A bit more complex: 2 components

$$\begin{aligned}\boldsymbol{\theta} &= [\pi_1, \boldsymbol{\theta}_1, \pi_2, \boldsymbol{\theta}_2] \\ &= [\pi_1, [\boldsymbol{\mu}_1, \Sigma_1], \pi_2, [\boldsymbol{\mu}_2, \Sigma_2]] \\ &\text{with} \quad \pi_1 + \pi_2 = 1\end{aligned}$$



$$\mathbb{LL}(\boldsymbol{\theta}, \mathcal{X}) = \sum_i^N \log\left[\pi_1 \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_1) + \pi_2 \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_2)\right]$$

$\Rightarrow$ difficult to maximize because of the sum inside the log!

$\Rightarrow$ Solution : iterative 2 steps (E-M) algorithm to maximize $\mathbb{LL}$

$\rightarrow$ Expectation-Maximization (E-M) algorithm

Estimation (2 components)

- ⋆ What is missing here is the assignment of $\boldsymbol{x}_i$ to one of the 2 components ϕ_j

$\Rightarrow$ If we knew it, we would treat the problem as twice 1 component

$\Rightarrow$ We introduce (unobserved) latent variables : the (binary) assignment $\delta_{ij} \in \{\text{true}, \text{false}\}$ of every $\boldsymbol{x}_i$ to component ϕ_j:

$$\boldsymbol{x}_i \sim \phi_1 \Rightarrow \delta_{i1} = \text{true}, \delta_{i2} = \text{false}$$
$$\boldsymbol{x}_i \sim \phi_2 \Rightarrow \delta_{i1} = \text{false}, \delta_{i2} = \text{true}$$

⚠ But we can only estimate $\delta_{ij} \Rightarrow$ EM strategy

Expectation (E-step)

$$\boldsymbol{\theta} = [\pi_1, \boldsymbol{\theta}_1, \pi_2, \boldsymbol{\theta}_2] = [\pi_1, [\boldsymbol{\mu}_1, \Sigma_1], \pi_2, [\boldsymbol{\mu}_2, \Sigma_2]]$$

- ⋆ Assume we know an initial value for $\boldsymbol{\theta}^0 = [\pi_1^0, \boldsymbol{\theta}_1^0, \pi_2^0, \boldsymbol{\theta}_2^0]$
- ⋆ We can infer the (statistical) contribution γ_{ij} of every data $\boldsymbol{x}_i$ to every component ϕ_j (parameterized by $\boldsymbol{\theta}_j^0$):

$$\gamma_{ij}(\boldsymbol{\theta}^0) = \mathbb{P}[\delta_{ij} = \text{true} | \boldsymbol{\theta}^0, \mathcal{X}]$$

$$\gamma_{i1}(\boldsymbol{\theta}^0) = \frac{\pi_1^0 \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_1^0)}{\pi_1^0 \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_1^0) + \pi_2^0 \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_2^0)}$$

- ⋆ γ_{ij} is the responsibility.
- ⋆ It is the likelihood that $\boldsymbol{x}_i$ is (purely) modeled by component ϕ_j

$\Rightarrow$ γ_{ij} represents the part of $\boldsymbol{x}_i$ that is modeled (generated) by component ϕ_j

Responsibility and soft-assignment

$\Rightarrow$ We can use γ_{ij} to determine δ_{ij} $\Rightarrow$ $\boldsymbol{x}_i$ can be assigned to either ϕ_1 or ϕ_2 (maximum vote $\rightarrow$ binarization)

$\Rightarrow$ K-means-type hard-assignment $\rightarrow$ each data is assigned to one and only one component (cluster)

EM is "softer": a data contributes (via γ_{ij}) to several components (density modes). EM computes a soft-assignment with

$$\sum_j \gamma_{ij} = 1 \quad \forall i$$

Maximization (M-step)

- ⋆ Note: $\gamma_{i1} + \gamma_{i2} = 1$
- ⋆ Given the responsibility of every data (γ_{ij}), we can estimate the parameters of every component by (weighted) maximum likelihood:

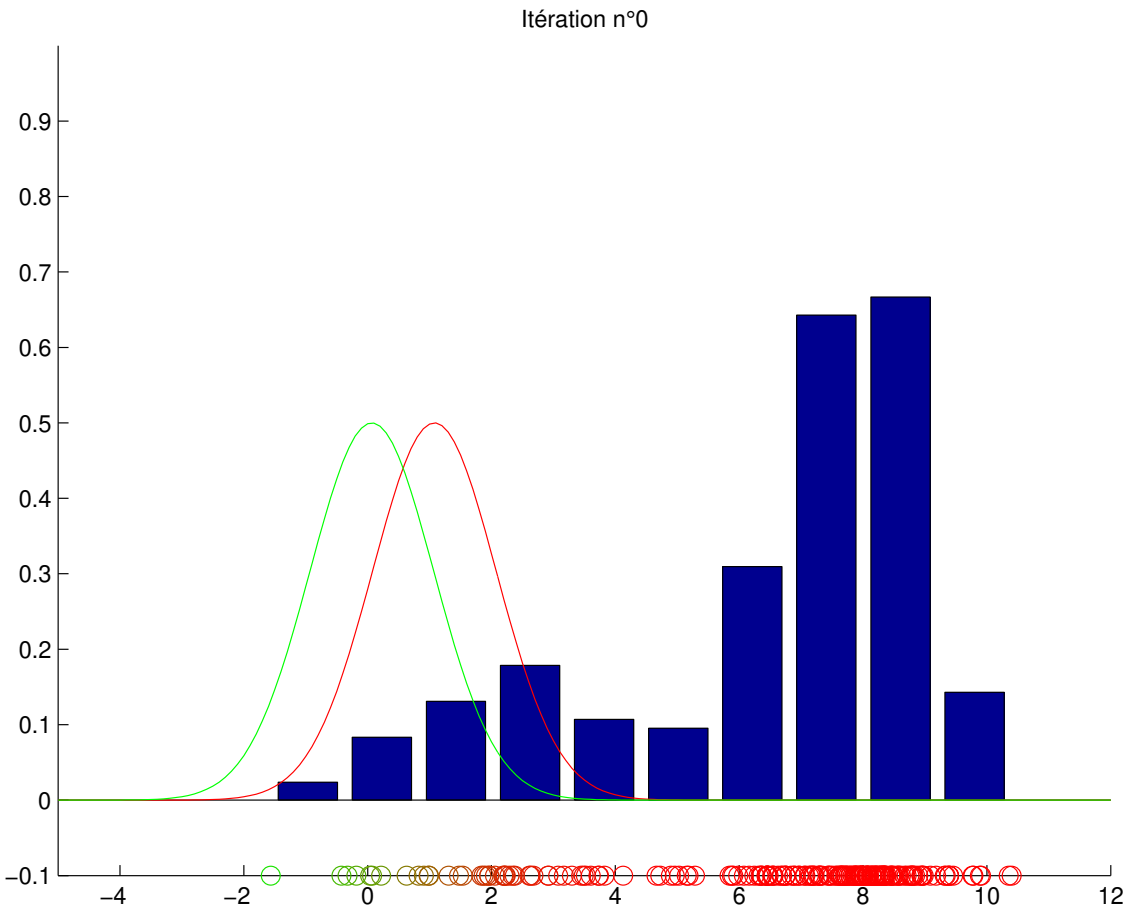
$$\hat{\boldsymbol{\mu}}_1 = \sum_{i=1}^{N} \frac{\gamma_{i1}}{\sum_{k=1}^{N} \gamma_{k1}} \boldsymbol{x}_i \qquad \hat{\boldsymbol{\Sigma}}_1 = \sum_{i=1}^{N} \frac{\gamma_{i1}}{\sum_{k=1}^{N} \gamma_{k1}} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_1)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_1)^{\mathsf{T}}$$

$$\hat{\boldsymbol{\mu}}_2 = \sum_{i=1}^{N} \frac{\gamma_{i2}}{\sum_{k=1}^{N} \gamma_{k2}} \boldsymbol{x}_i \qquad \hat{\boldsymbol{\Sigma}}_2 = \sum_{i=1}^{N} \frac{\gamma_{i2}}{\sum_{i=1}^{N} \gamma_{i2}} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_2)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_2)^{\mathsf{T}}$$

- ⋆ Weight for mixture j: $\pi_j = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ij}$ $\qquad (\Rightarrow \text{Ⓟ} \sum_j \pi_j = 1)$

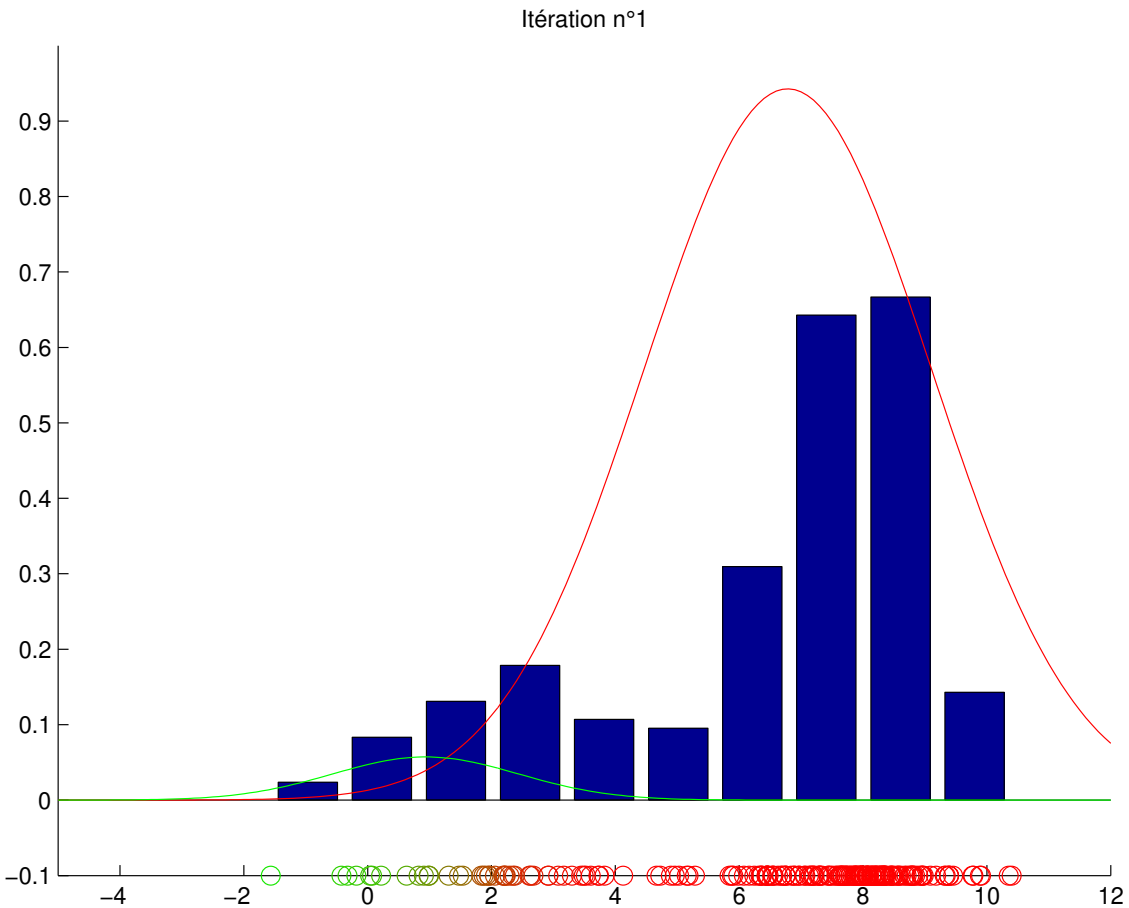
Example

Successive iterations of E- and M-steps



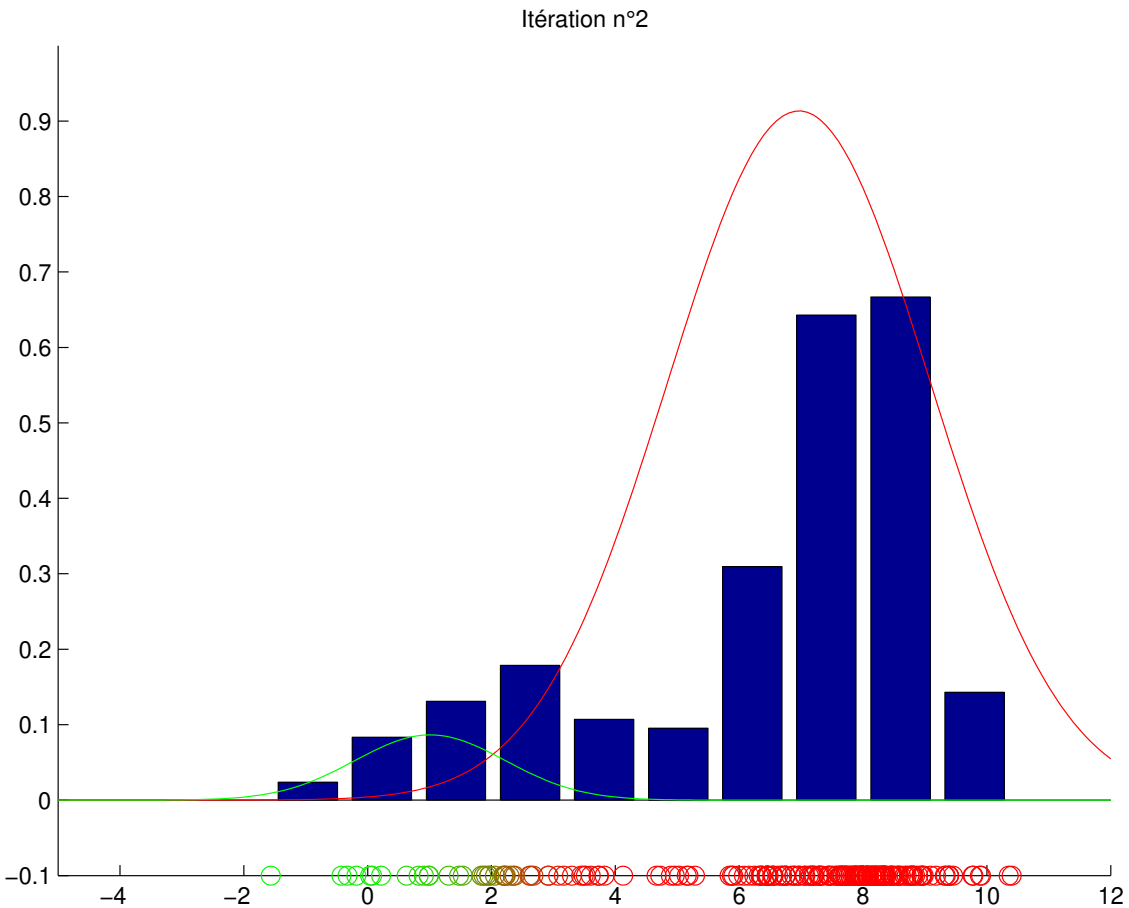
Example

Successive iterations of E- and M-steps



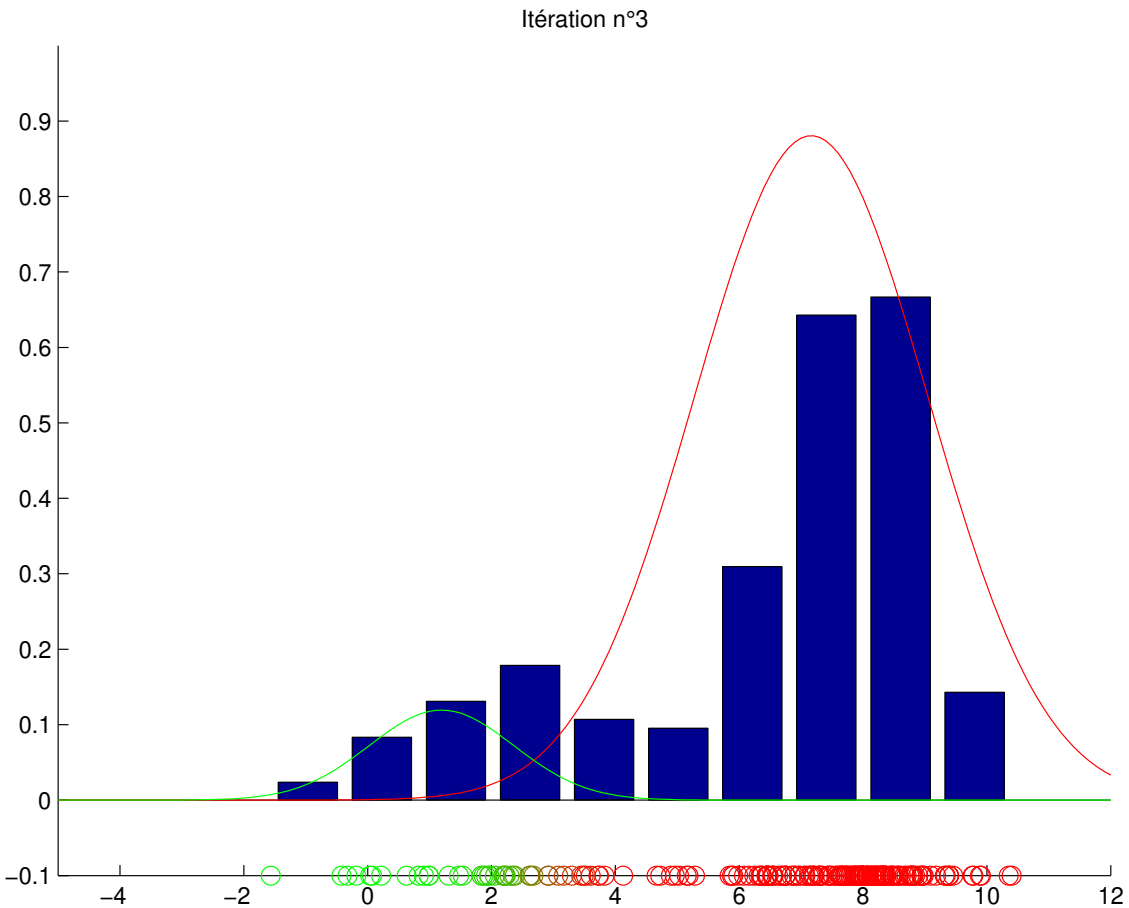
Example

Successive iterations of E- and M-steps



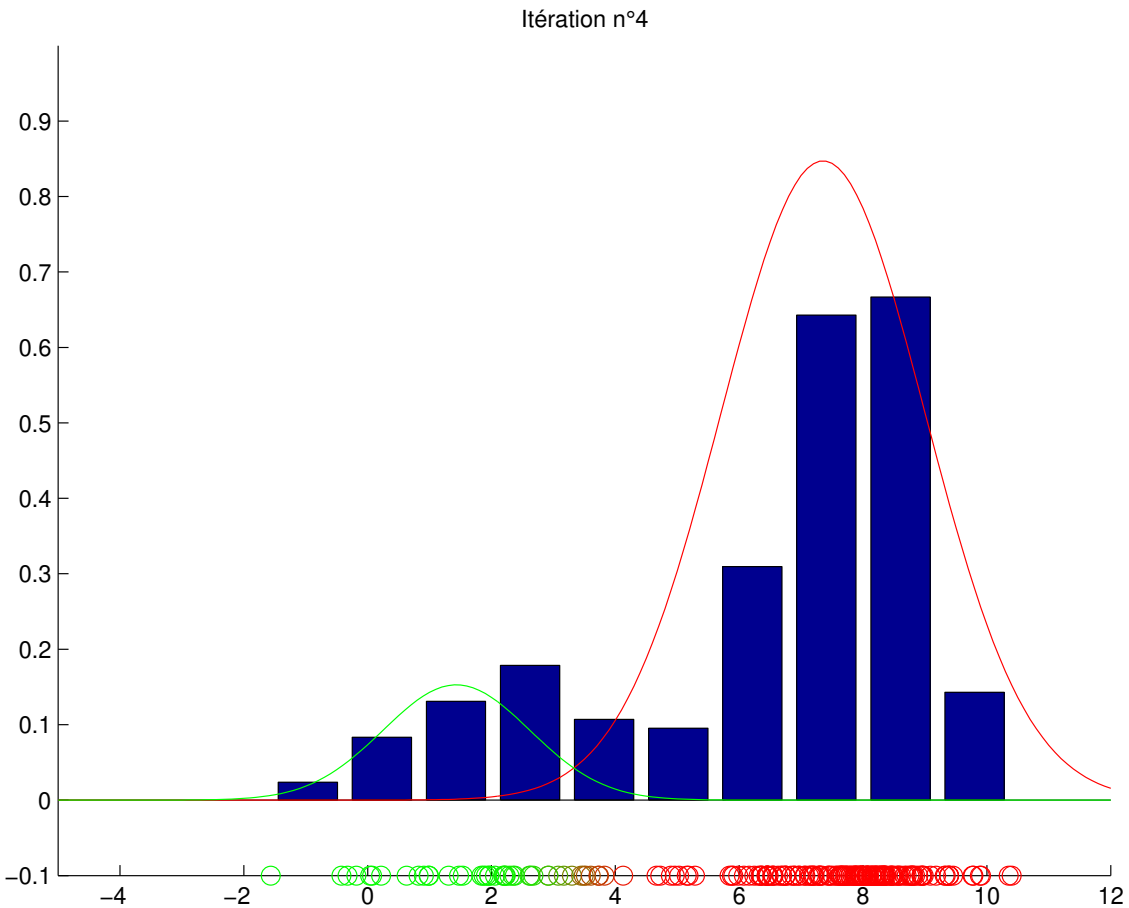
Example

Successive iterations of E- and M-steps



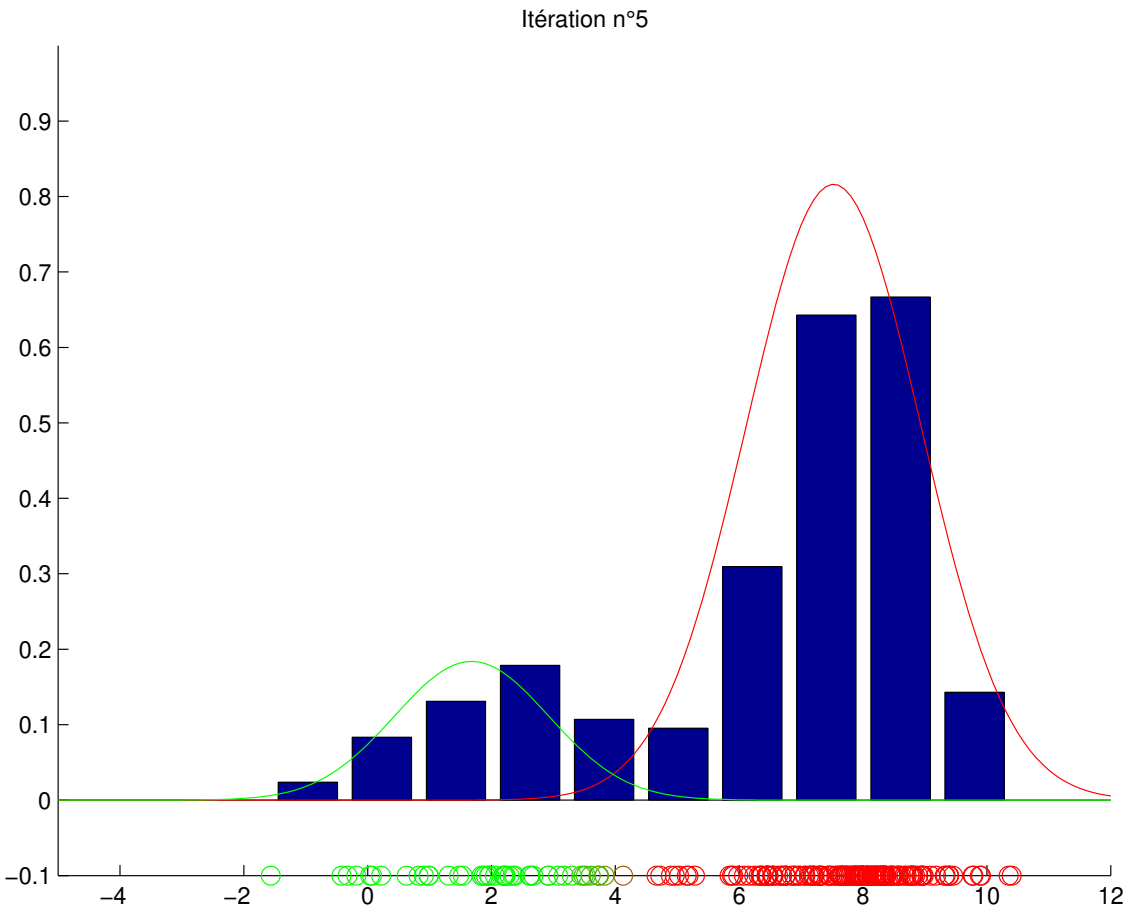
Example

Successive iterations of E- and M-steps



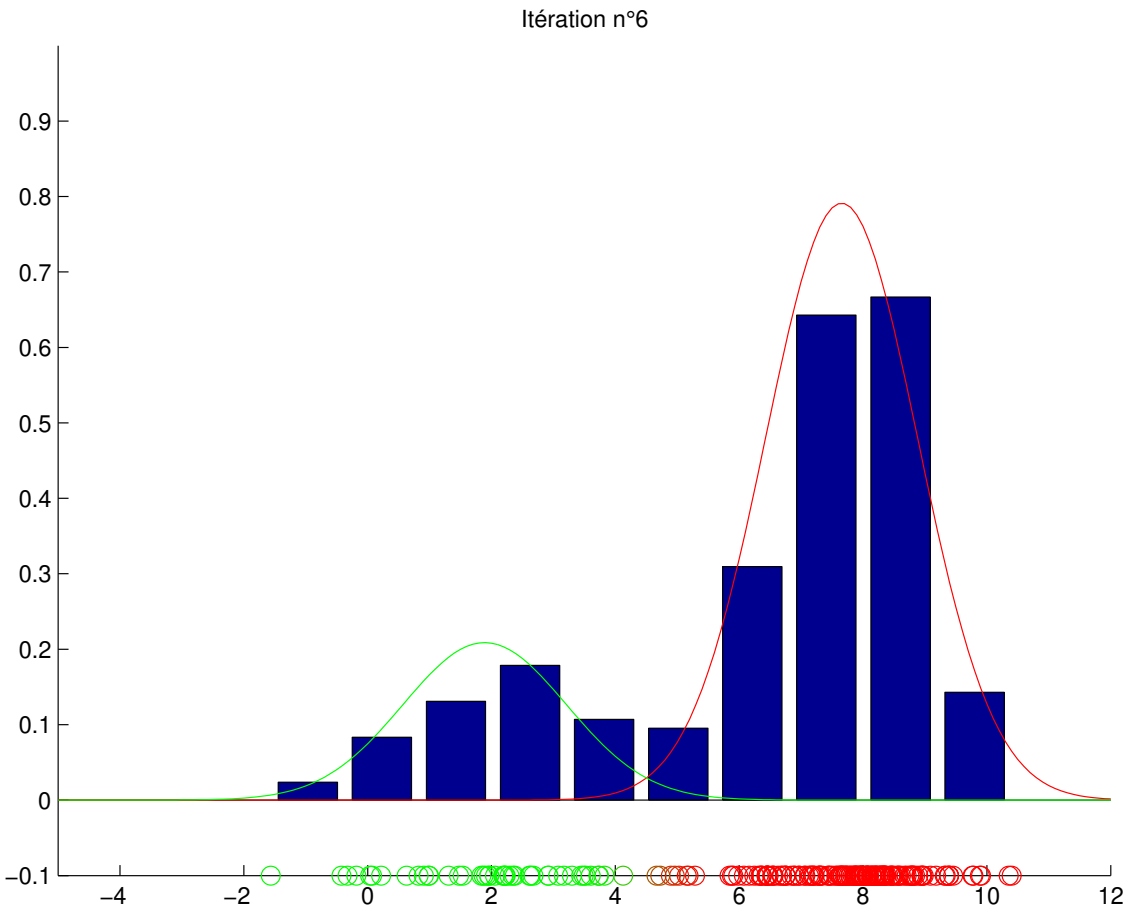
Example

Successive iterations of E- and M-steps



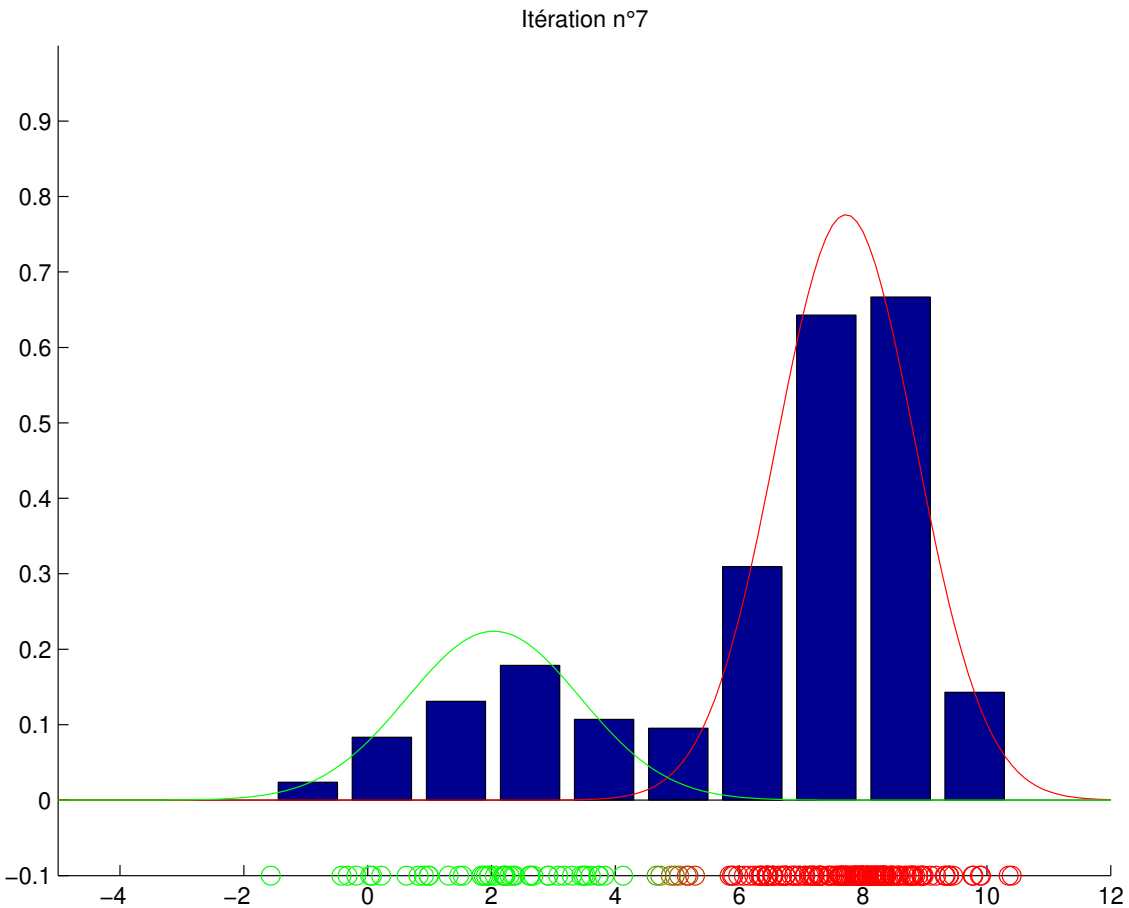
Example

Successive iterations of E- and M-steps



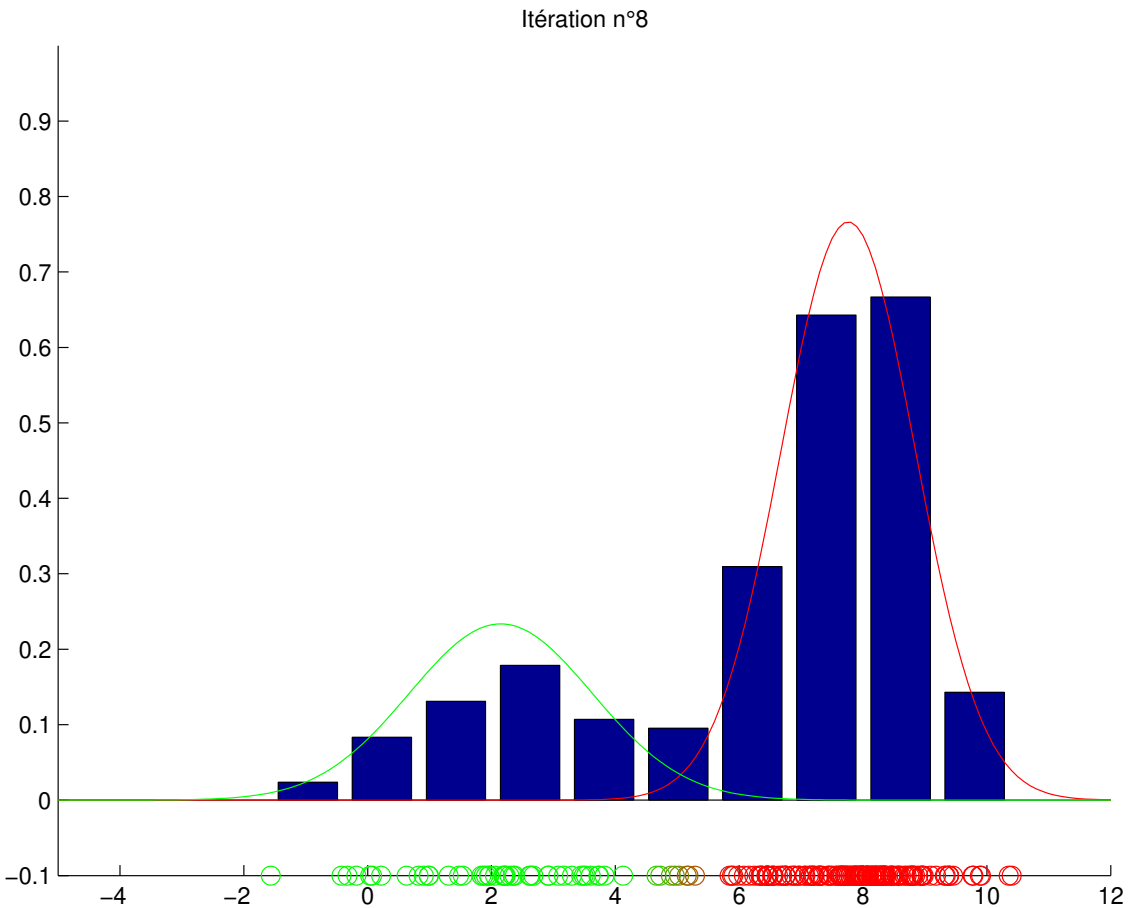
Example

Successive iterations of E- and M-steps



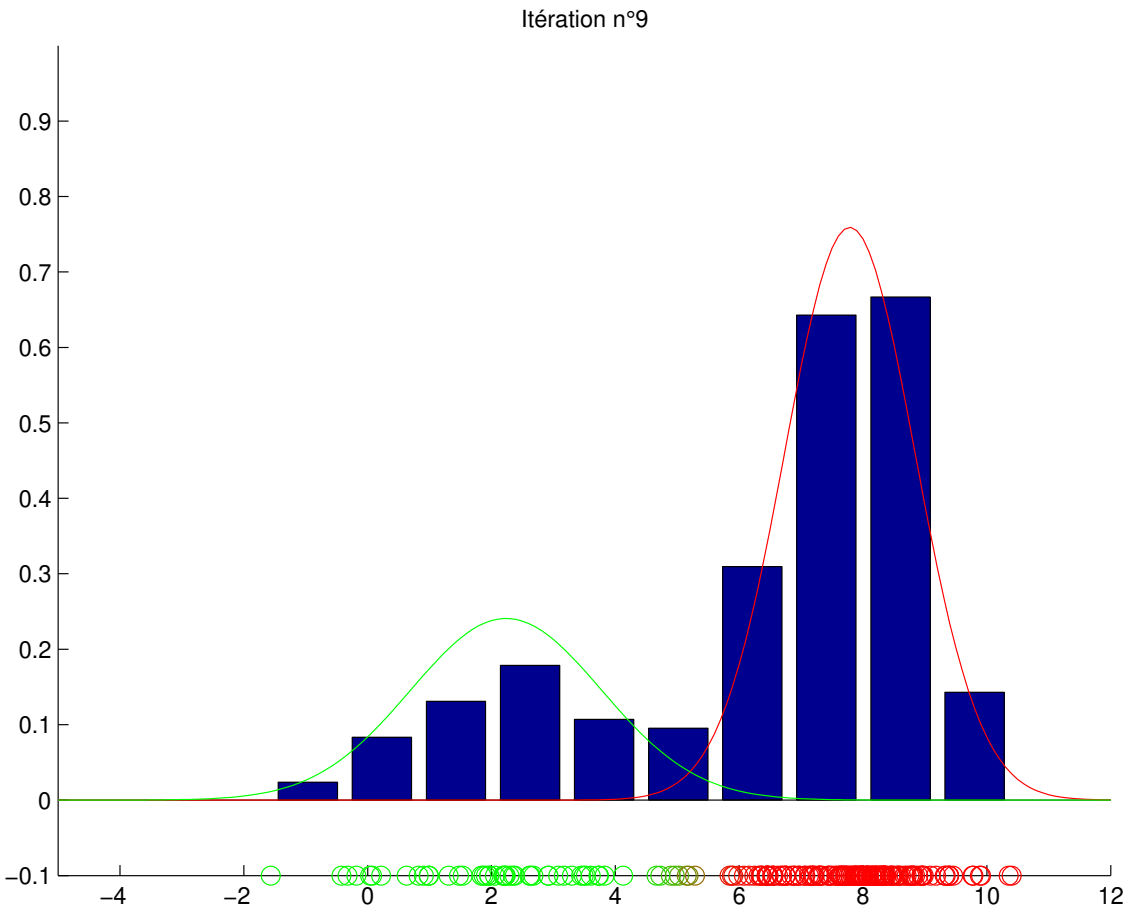
Example

Successive iterations of E- and M-steps



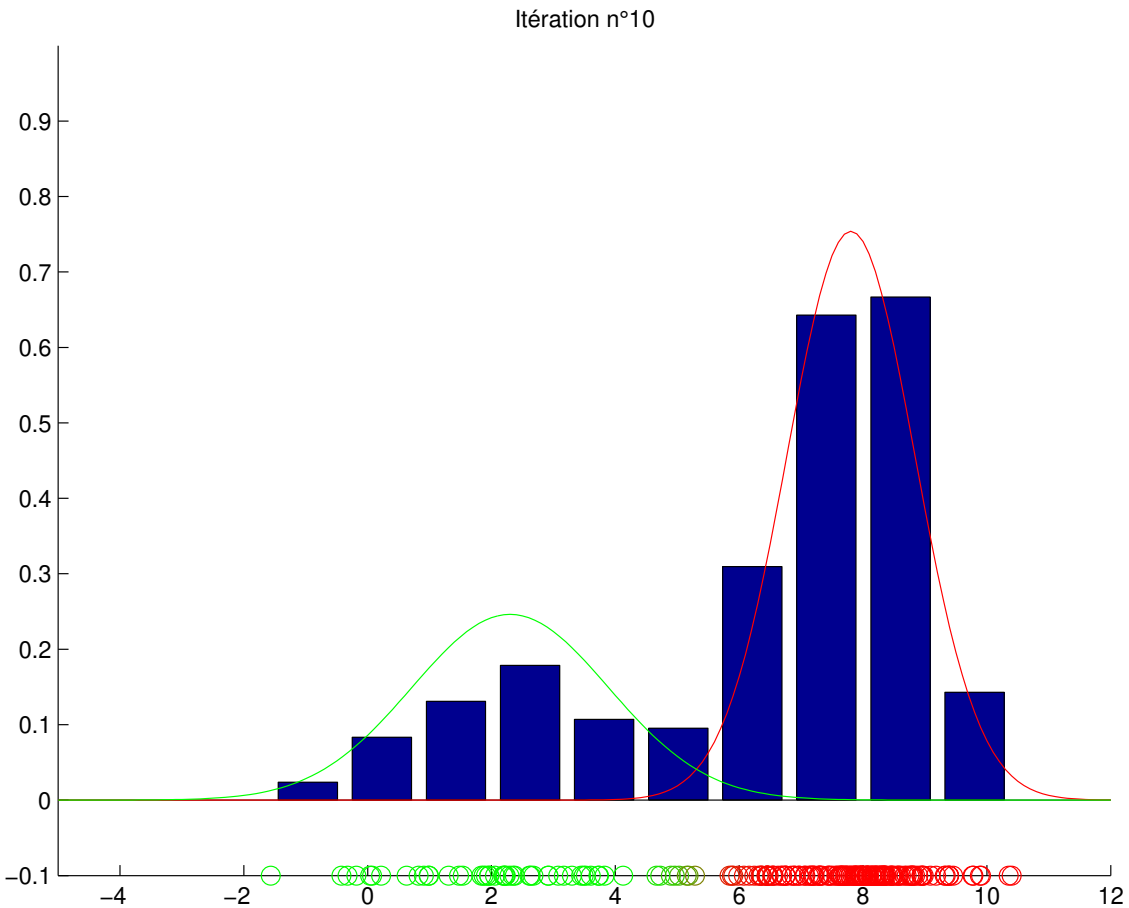
Example

Successive iterations of E- and M-steps



Example

Successive iterations of E- and M-steps



Results

- True parameters

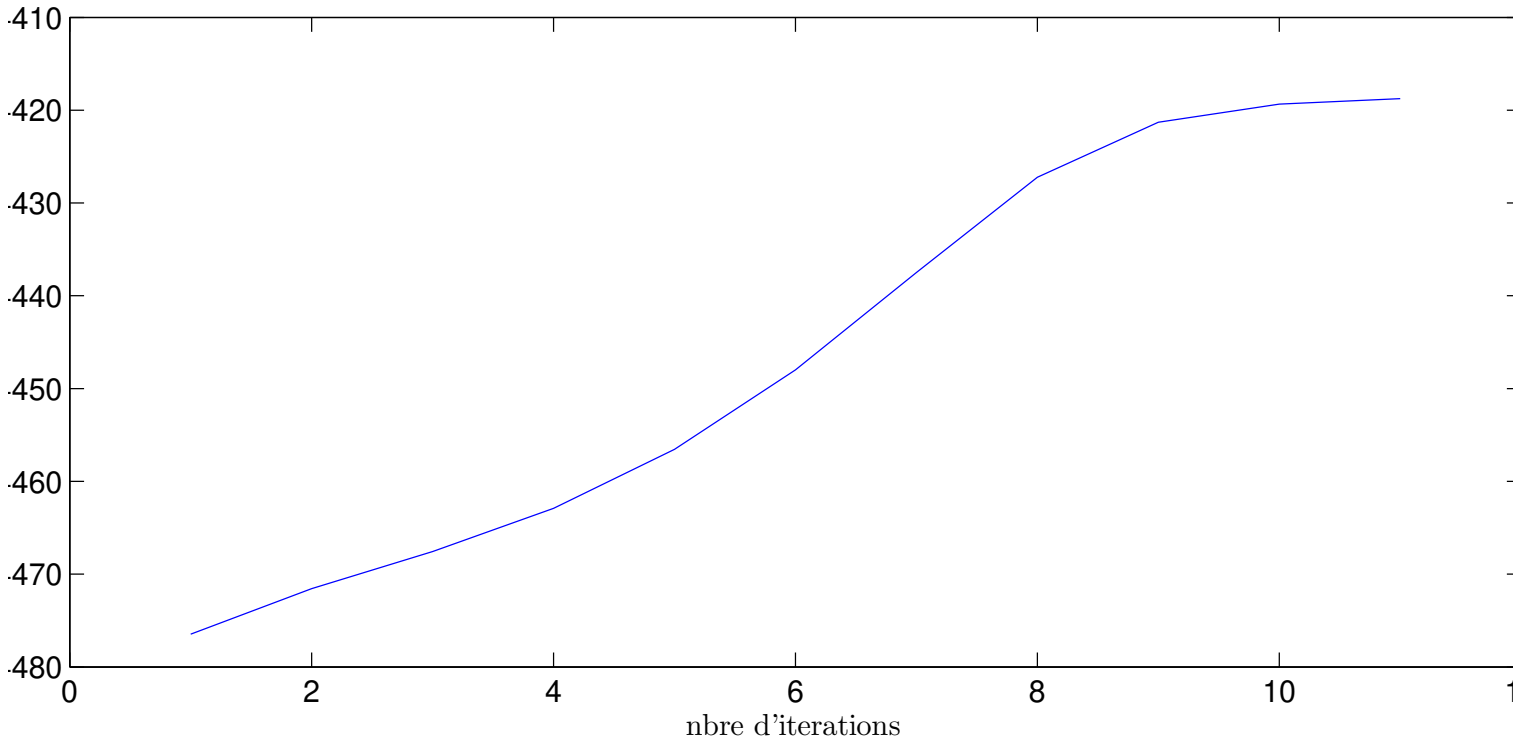
π_1	$\boldsymbol{\mu}_1$	Σ_1	$\boldsymbol{\mu}_2$	Σ_2
0.75	2	2	8	1

- Estimated parameters

$\hat{\pi}_1$	$\hat{\boldsymbol{\mu}}_1$	$\hat{\Sigma}_1$	$\hat{\boldsymbol{\mu}}_2$	$\hat{\Sigma}_2$
10 iterations				
0.76	2.17	1.56	7.91	1.06
20 iterations				
0.76	2.15	1.98	8.01	0.98

Iterations

Alternate cycle Expectation-Maximization $\rightarrow$ increases the likelihood of the data w.r.t mixture model



The process is iterated until convergence, i.e when the likelihood of the data does not change (much)

Limitations

- ⋆ Hill-climbing $\Rightarrow$ depends on initial parameters

$\Rightarrow$ sensitive to local maxima

$\Rightarrow$ Initialization strategies (e.g k-means or k-means $++$)

- ⋆ Potential slow convergence, depending on the distributions

c-component mixtures

Generalization with :

$$\delta_{i1}, \delta_{i2}, \ldots, \delta_{ij}, \ldots, \delta_{ic}$$

$\delta_{ij} =$ true if data $\boldsymbol{x}_i$ is generated by component ϕ_j ($\delta_{ij} =$ false otherwise)

$\Rightarrow$ Responsibility γ_{ij} : expectation of δ_{ij} over all the components

Parameters $\boldsymbol{\theta} = [\pi_j, \boldsymbol{\theta}_j]_{j \in [\![c]\!]} = [\pi_j, [\boldsymbol{\mu}_j, \Sigma_j]]_j$ to estimate

EM algorithm, c components

1. Initial $\boldsymbol{\theta}^0 = \{\pi_j^0, \boldsymbol{\mu}_j^0, \Sigma_j^0\}_{j \in [\![c]\!]}$
   - In general: $\pi_j = 1/c$, $\boldsymbol{\mu}_j$ is chosen at random and $\Sigma_k = \mathbf{Id}$
   - Alternative: use k-means as initialization
2. E-step : compute responsibilities for every data $i \in [\![N]\!]$ and every component $j \in [\![c]\!]$

$$\gamma_{ij} = \frac{\pi_j \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\sum_{k=1}^{c} \pi_k \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_k)}$$

3. M-step Estimations of mixture parameters

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} \gamma_{ij} \boldsymbol{x}_i}{\sum_i \gamma_{ij}}; \quad \Sigma_j = \frac{\boldsymbol{X}_j \Gamma_j \boldsymbol{X}_j^\top}{\mathrm{Tr}(\Gamma_j)}; \quad \pi_j = \frac{\sum_i \gamma_{ij}}{N}$$

with $\Gamma_j = \mathrm{diag}(\gamma_{1j}, \ldots, \gamma_{Nj})$, $\boldsymbol{X}_j$ centered on $\boldsymbol{\mu}_j$

4. Iterate 2. and 3. until convergence

Modeling

- ⋆ The *a priori* parametrization of the mixture changes the convergence

$\Rightarrow$ Parameters

1. c: number of components
2. Σ_j: the shape of covariance matrices (diagonal, full, parameterized)

- ⋆ Too flexible or too rigid models mean wrong or no convergence...
- ⋆ Number of variables : $D \times D \times c + 2 \times c$: if N low, D large and c large $\rightarrow$ over-parameterized

Shape of the covariance matrix

Over-parameterized problem

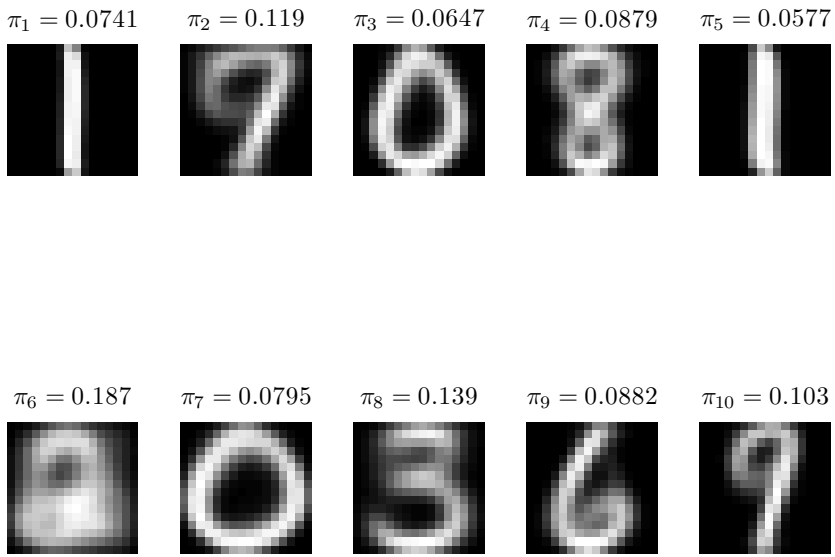
- ⋆ $\Sigma \in \mathbb{R}^{D \times D}$
- ⋆ e.g: character recognition $\boldsymbol{x}_i \in \mathbb{R}^{256}$

$\Rightarrow$ Needs to estimate $256^2 \times c$ parameters for the covariance (given about 7000 data points)!

Matrix parameterization

- ⋆ Spherical models $\Sigma = \sigma \mathbf{Id}$ $\rightarrow 1$ parameter
- ⋆ Diagonal models $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_D)$ $\rightarrow D$ parameters
- ⋆ Full models $\Sigma \in \mathbb{R}^{D \times D}$ $\rightarrow O(D^2)$ parameters

Character recognition



More complex models $\rightarrow$ no convergence since p is too large

Pre-processing : using PCA to reduce the dimension

Recall : 50 principal components reconstruct 90% of the signal

EM within the space of the 50 first PC

Pre-processing : using PCA to reduce the dimension

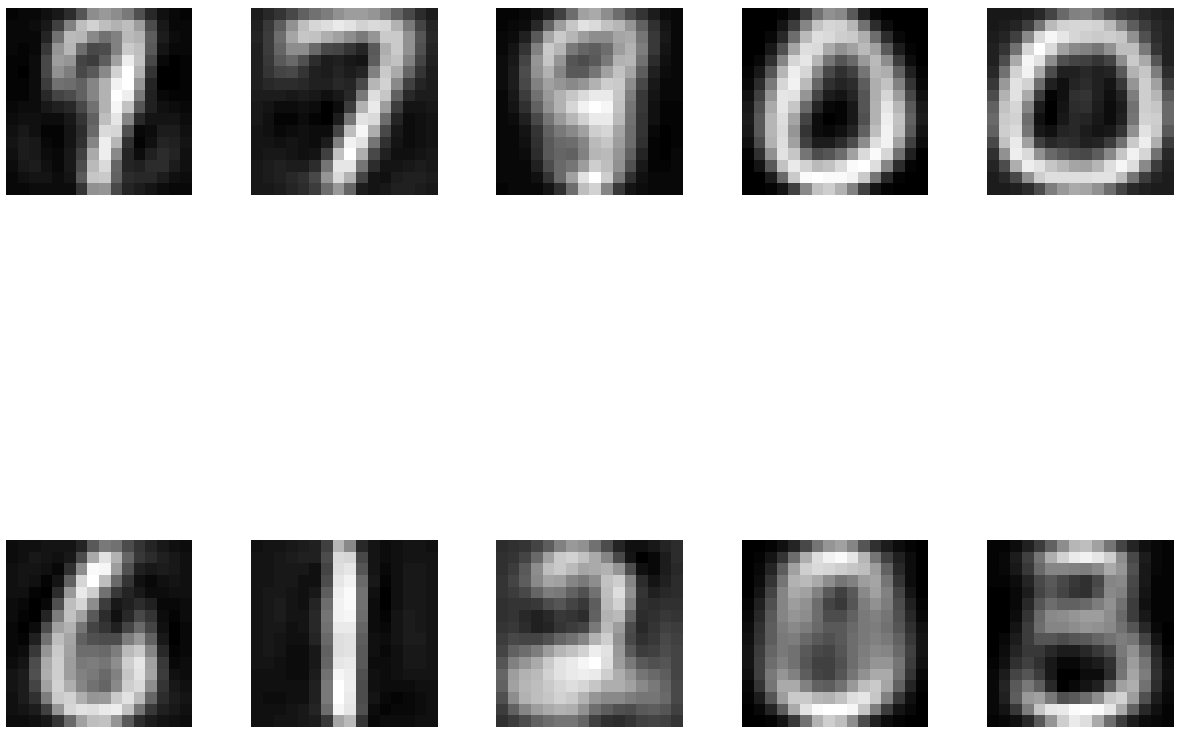
Recall : 50 principal components reconstruct 90% of the signal

EM within the space of the 10 first PC

Pre-processing : using PCA to reduce the dimension

Recall : 50 principal components reconstruct 90% of the signal

EM within the space of the 2 first PC



Number of components

Parsimony

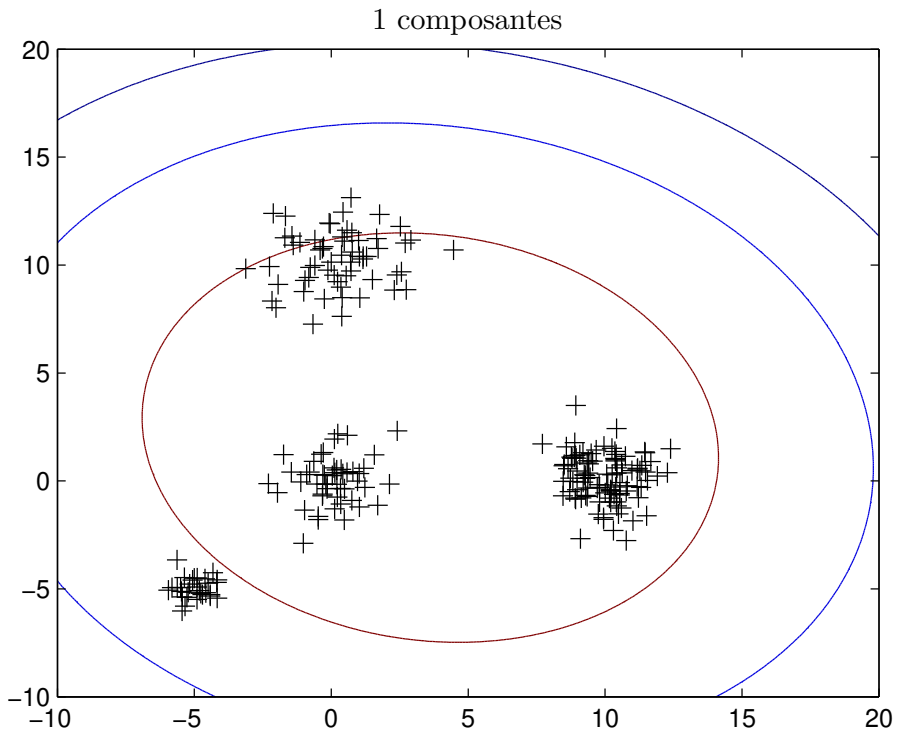
- ⋆ The larger c, the less points may be assigned to every component (in average)
- ⋆ Search for parsimonious models, i.e small number of parameters to estimate

Bayesian Information Criterion (BIC)

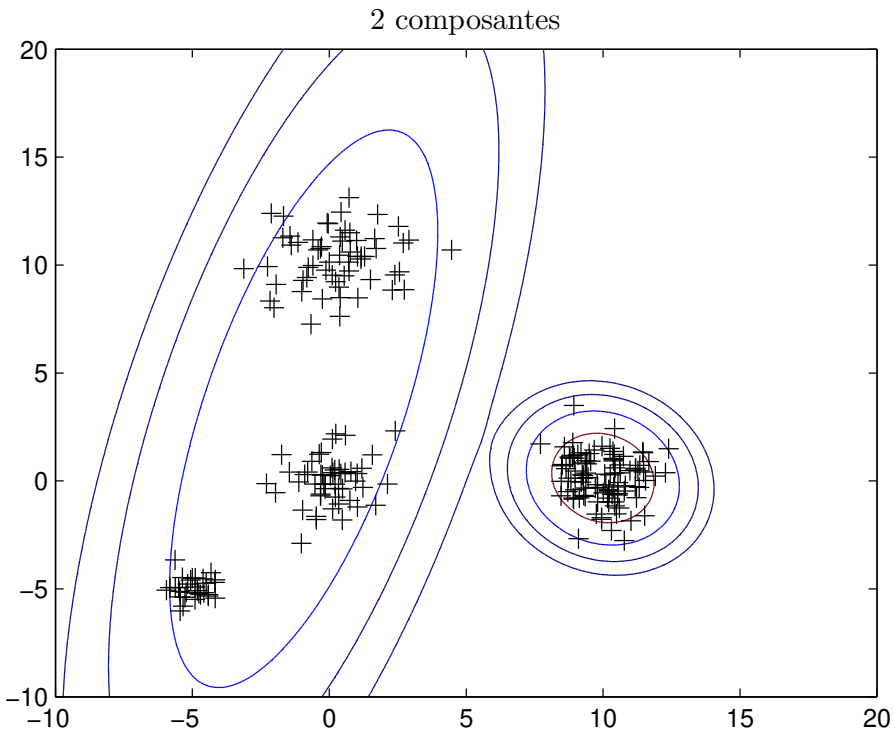
- ⋆ The larger c is, the better the estimate of l
- ⋆ Trade likelihood against complexity

$$\text{BIC}(\boldsymbol{\theta}, \mathcal{X}) = -2\mathbb{LL}(\boldsymbol{\theta}, \mathcal{X}) + |\boldsymbol{\theta}| \log(N) \qquad \hat{\boldsymbol{\theta}} \overset{\text{here}}{=} \underset{\boldsymbol{\theta}}{\text{argmin}}\, \text{BIC}(\boldsymbol{\theta}, \mathcal{X})$$

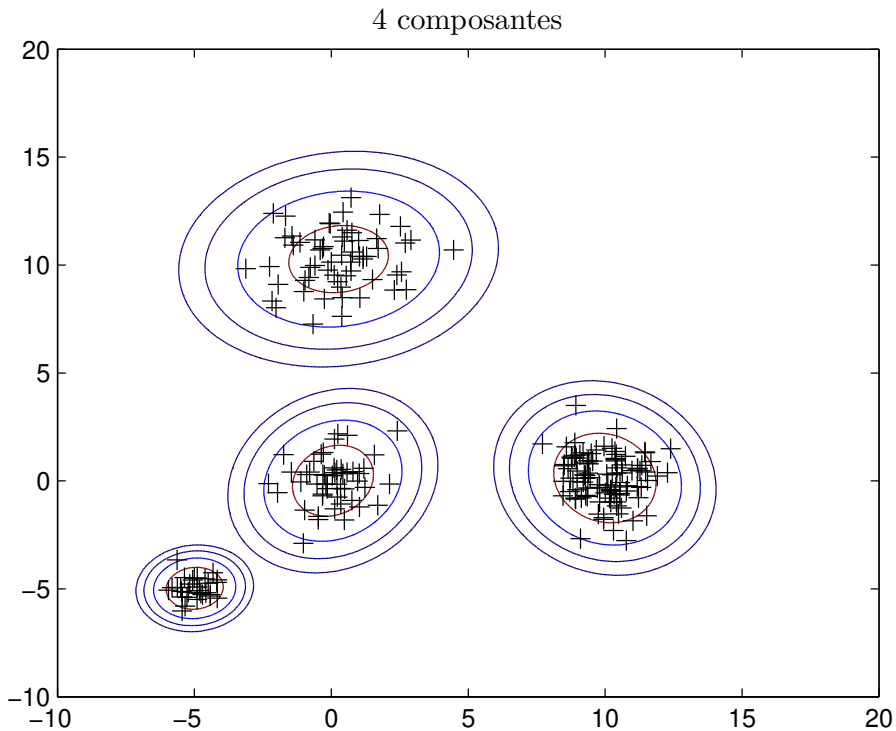
Example: 4-component mixture



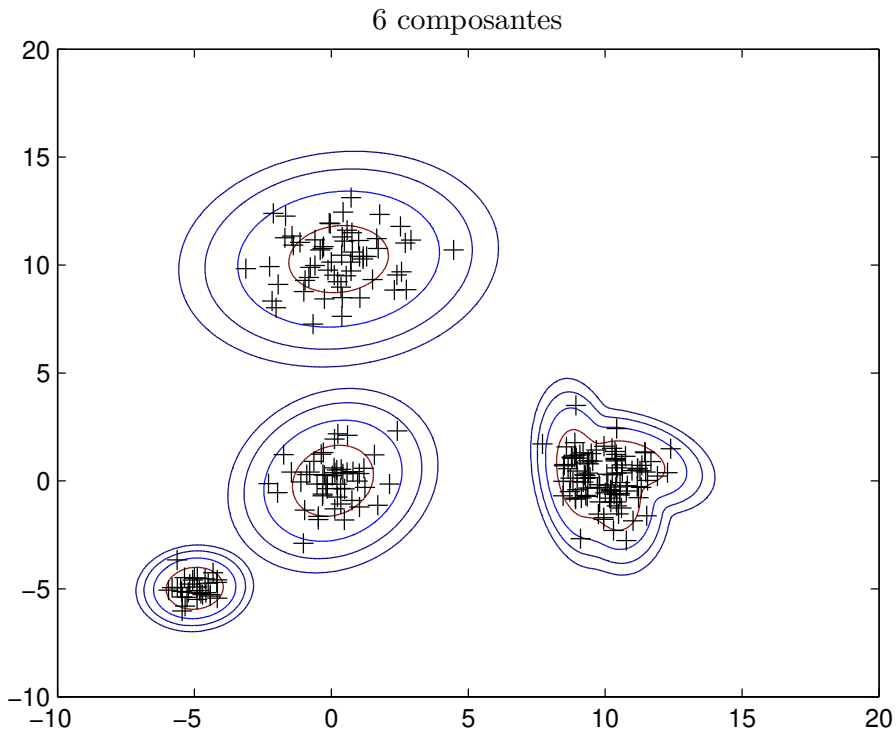
Example: 4-component mixture



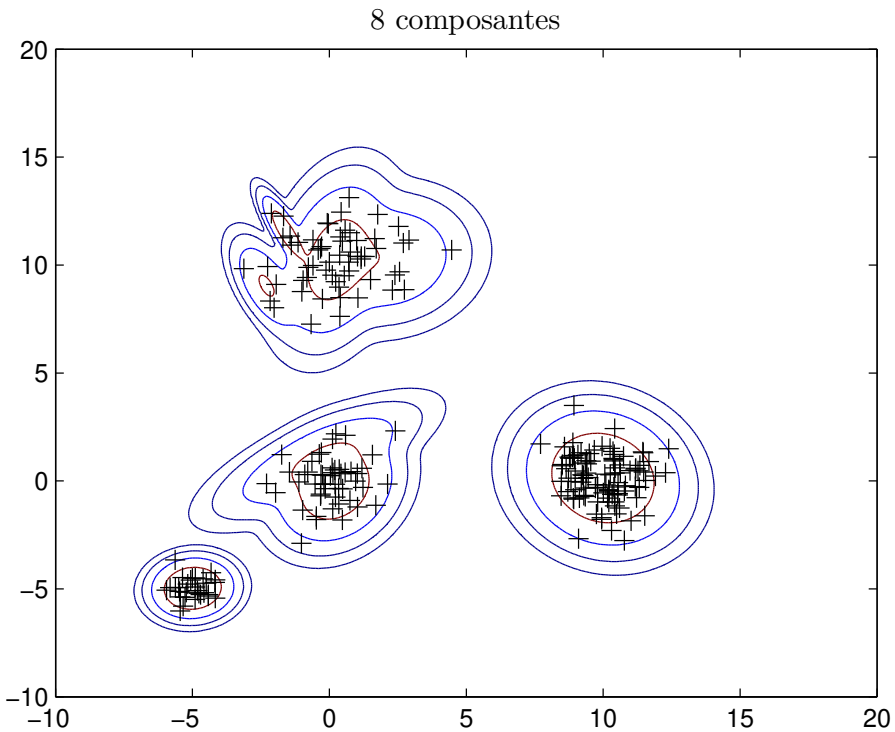
Example: 4-component mixture



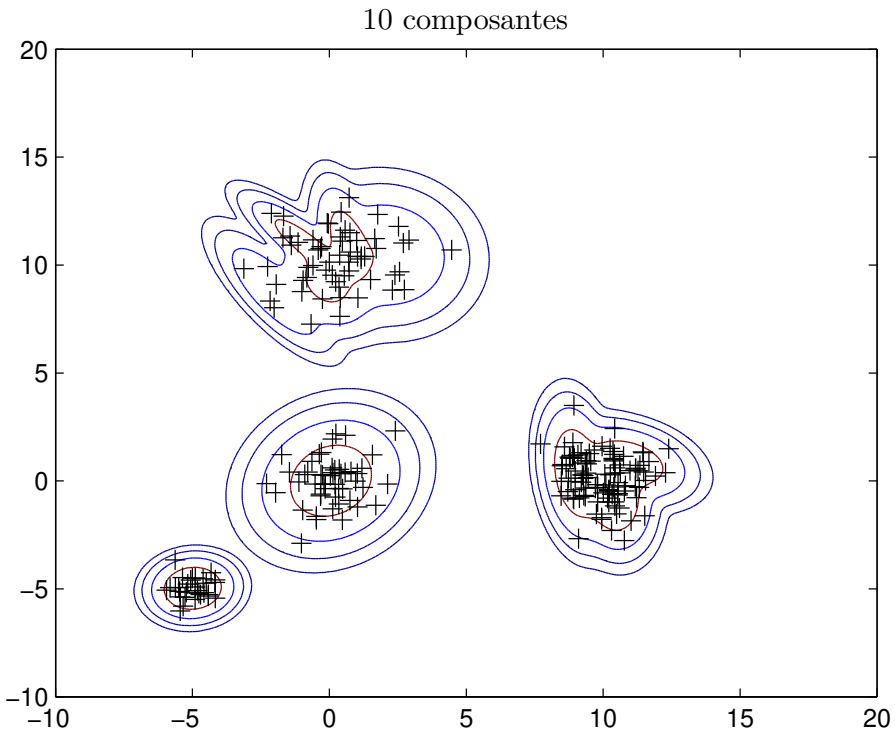
Example: 4-component mixture



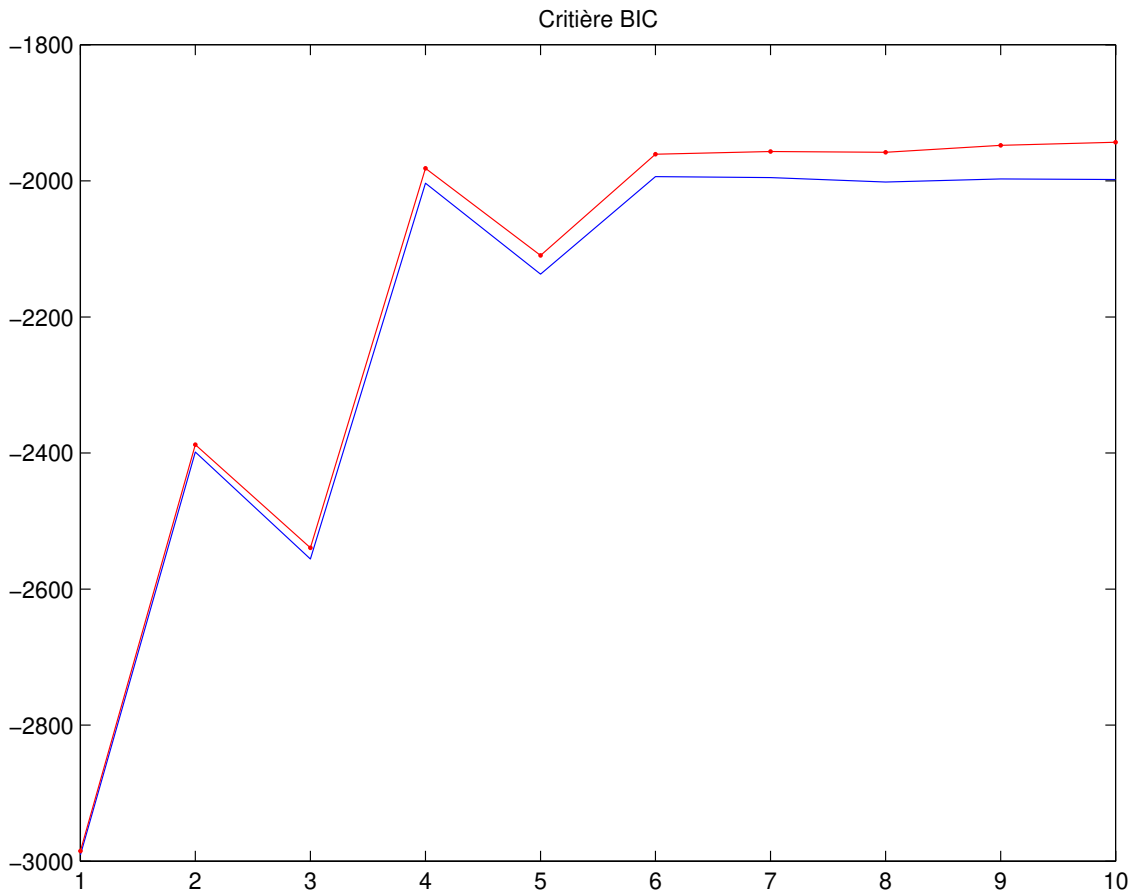
Example: 4-component mixture



Example: 4-component mixture



Example (contd)



- ⋆ Need to test all models
- ⋆ Depends on convergence
- ⋆ *Fine tuning* by hand!

Summary

Gaussian mixtures

- ⋆ The Gaussian mixture model generalizes the assumption underlying PCA and LDA
- ⋆ Explicit density modeling and estimation
- ⋆ Also MLE classification (unsupervised): if components are classes, data $\boldsymbol{x}_i$ is associated to class j for which $\mathbb{P}(C = j|\boldsymbol{x}_i) \approx \pi_j \phi_j(\boldsymbol{x}_i)$ is maximized among all classes

EM algorithm

- ⋆ Iterative algorithm to maximize the (log-)likelihood
- ⋆ Principle used in many other scenarios
- ⋆ Based on the definition of unobserved (latent) variables (δ_{ij})
- ⋆ Probabilistic Latent Semantic Analysis (pLSA) $\rightarrow$ EM where the hidden variables are the latent concepts

Example questions [mostly require formal – mathematical – answers]

- What is a Gaussian Mixture model (GMM)?
- Why can it be viewed as conditional modeling?
- How can we use it for generating data?
- What is the responsibility? How to interpret it?
- Why is EM for GMM a MLE?
- How to interpret the E-step?
- How to interpret the M-step?
- Discuss the relationship betwen GMM and k-means
- What are hard- and soft-assignments?

It is strongly adviced to develop the algebra contained in this chapter

References I

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. (available online).

[2] Kevin P. Murphy. *Probabilistic Machine Learning: an Introduction*. MIT Press, 2022. (available online).