Data Science
Expectation-Maximization for Density Modelling

Latent factor estimation

Stéphane Marchand-Maillet

Department of Computer Science

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES

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

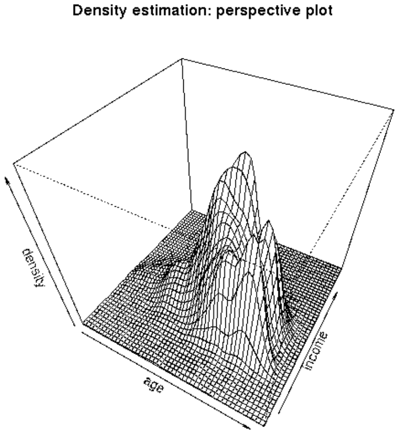
⇒ The distribution is fixed (essentially normal) and we look for its parameters θ (e.g $\theta = [\boldsymbol{\mu}, \boldsymbol{\Sigma}]$)

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

Density estimation

Methods

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



Mixture models

Definition

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

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

- $\pi_j \in \mathbb{R}$ are the mixture parameters
- $\boldsymbol{\phi}(x, \theta_j) \in \mathbb{F}$ are functions controlled by parameters θ_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 $f(x)$ represents a random process where x is drawn from a set of states ω_j with prior probability $\mathbb{P}(\omega_j)$

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

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

We get (by identification):

- ⋆ $\pi_j = \mathbb{P}(\omega_j)$ so that $\sum_j \pi_j = 1$
- ⋆ $\phi(x, \theta_j) = \mathbb{P}(x|\omega_j, \theta_j)$ is the conditional probability that x is generated by state ω_j

Reminder: Maximum log-likelihood (MLE)

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

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

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

- Log-likelihood

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

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

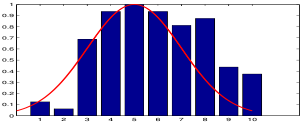
Gaussian mixture

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:

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

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

Basic case : 1 component, c = 1



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

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_i \log e^{-(x_i-\mu)^T \mathbf{\Sigma}^{-1}(x_i-\mu)}$$

$$\Longleftrightarrow$$

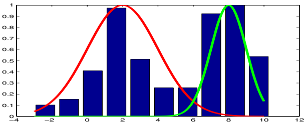
$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_i (x_i-\mu)^T \mathbf{\Sigma}^{-1}(x_i-\mu)$$

$\Rightarrow$ $\hat{\mu} = \frac{1}{N}\sum_i x_i$ φ

$\Rightarrow$ $\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_i (x_i-\hat{\mu})(x_i-\hat{\mu})^T$ φ

A bit more complex: 2 components

$$\theta = [\pi_1, \theta_1, \pi_2, \theta_2]$$
$$= [\pi_1, [\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1], \pi_2, [\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2]]$$
with $\pi_1 + \pi_2 = 1$



$$\mathbb{L}(\theta, \mathcal{X}) = \sum_i^N \log\left[\pi_1 \boldsymbol{\phi}(x_i, \theta_1) + \pi_2 \boldsymbol{\phi}(x_i, \theta_2)\right]$$

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

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

$\rightarrow$ 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 $\boldsymbol{\phi}_j$

⇒ If we knew it, we would treat the problem as twice 1 component

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

$$\boldsymbol{x}_i \sim \boldsymbol{\phi}_1 \Rightarrow \delta_{i1} = \texttt{true}, \delta_{i2} = \texttt{false}$$
$$\boldsymbol{x}_i \sim \boldsymbol{\phi}_2 \Rightarrow \delta_{i1} = \texttt{false}, \delta_{i2} = \texttt{true}$$

⚠ But we can only estimate δ_{ij} ⇒ EM strategy

Expectation (E-step)

$\theta = [\pi_1, \theta_1, \pi_2, \theta_2] = [\pi_1, [\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1], \pi_2, [\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2]]$

- ⋆ Assume we know an initial value for $\theta^0 = [\pi_1^0, \theta_1^0, \pi_2^0, \theta_2^0]$
- ⋆ We can infer the (statistical) contribution γ_{ij} of every data $\mathbf{x}_i$ to every component $\boldsymbol{\phi}_j$ (parameterized by θ_j^0):

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

$$\gamma_{i1}(\theta^0) = \frac{\pi_1^0 \boldsymbol{\phi}(\mathbf{x}_i, \theta_1^0)}{\pi_1^0 \boldsymbol{\phi}(\mathbf{x}_i, \theta_1^0) + \pi_2^0 \boldsymbol{\phi}(\mathbf{x}_i, \theta_2^0)}$$

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

⇒ γ_{ij} represents the part of $\mathbf{x}_i$ that is modeled (generated) by component $\boldsymbol{\phi}_j$

Responsibility and soft-assignment

- $\Rightarrow$ We can use γ_{ij} to determine δ_{ij} $\Rightarrow$ $\mathbf{x}_i$ can be assigned to either $\boldsymbol{\phi}_1$ or $\boldsymbol{\phi}_2$ (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:

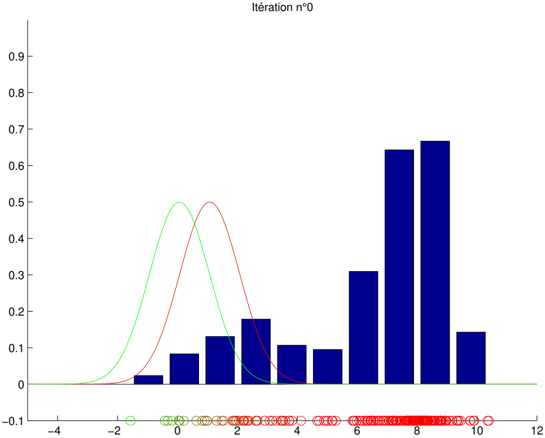
$$\hat{\boldsymbol{\mu}}_1 = \sum_{i=1}^{N} \frac{\gamma_{i1}}{\sum_{k=1}^{N} \gamma_{k1}} \boldsymbol{x}_i \qquad \hat{\boldsymbol{\Sigma}}_1 = \sum_{i=1}^{N} \frac{\gamma_{i1}}{\sum_{k=1}^{N} \gamma_{k1}} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_1)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_1)^T$$

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

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

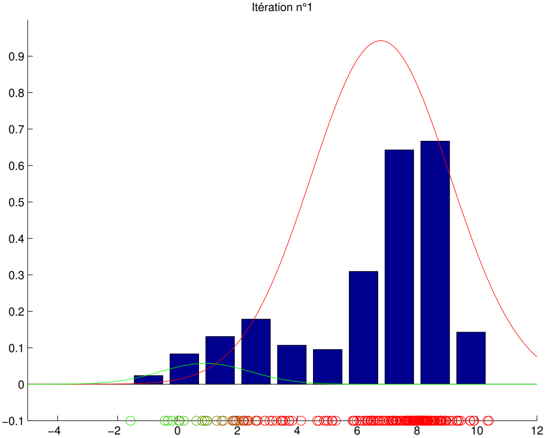
Example

Successive iterations of E- and M-steps



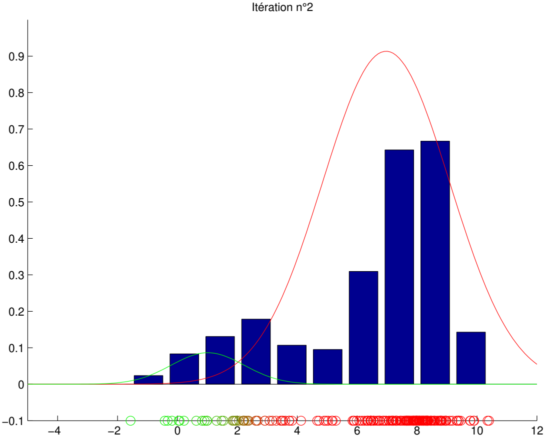
Example

Successive iterations of E- and M-steps



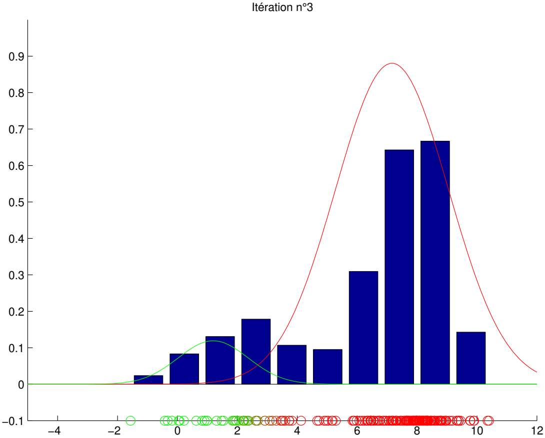
Example

Successive iterations of E- and M-steps



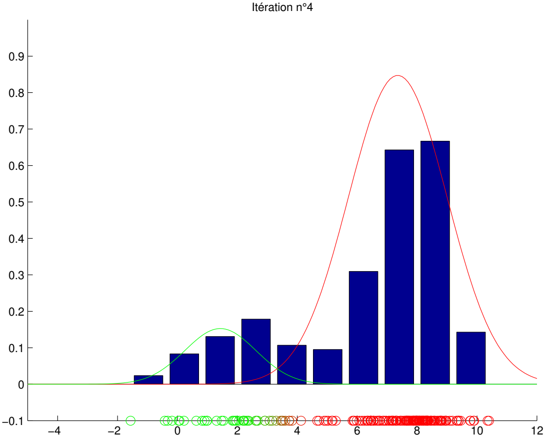
Example

Successive iterations of E- and M-steps



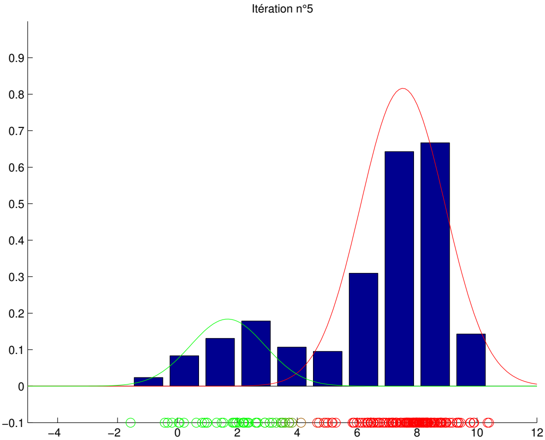
Example

Successive iterations of E- and M-steps



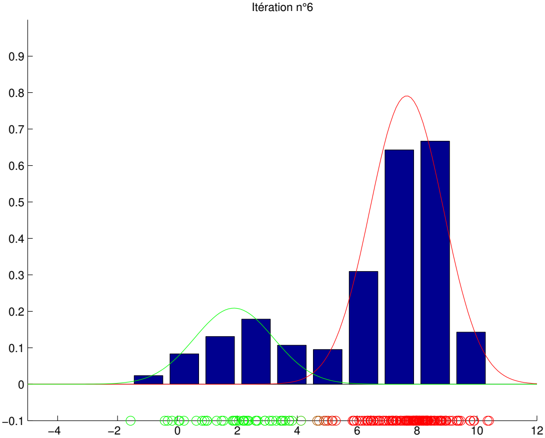
Example

Successive iterations of E- and M-steps



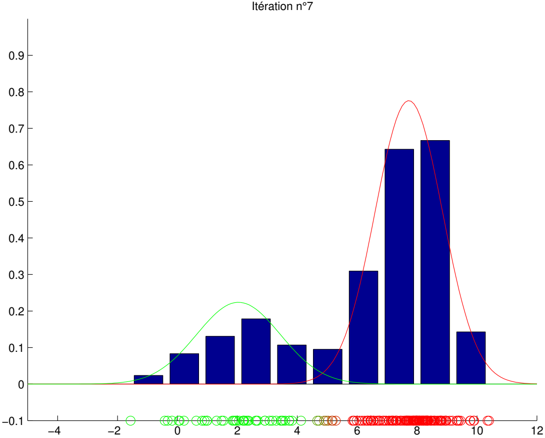
Example

Successive iterations of E- and M-steps



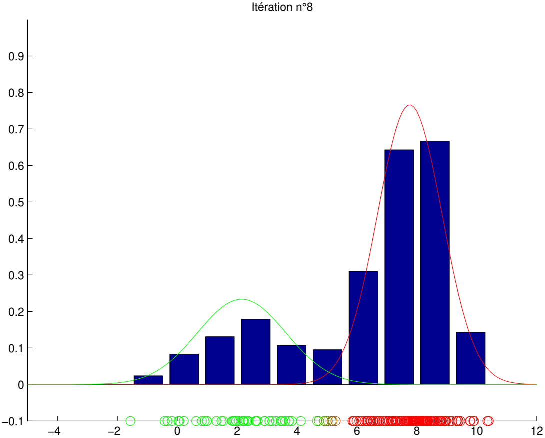
Example

Successive iterations of E- and M-steps



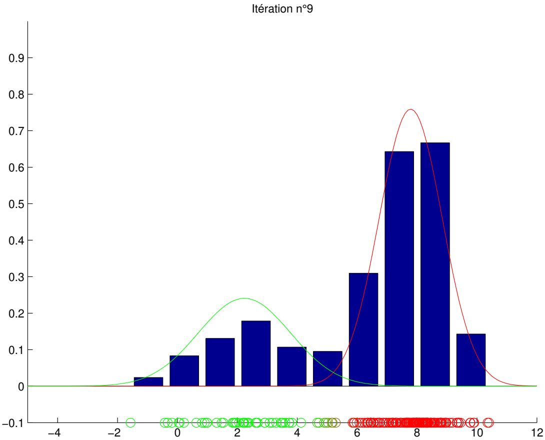
Example

Successive iterations of E- and M-steps



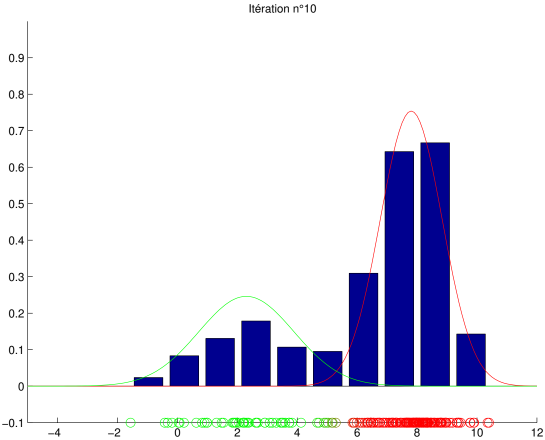
Example

Successive iterations of E- and M-steps



Example

Successive iterations of E- and M-steps



Results φ

- True parameters

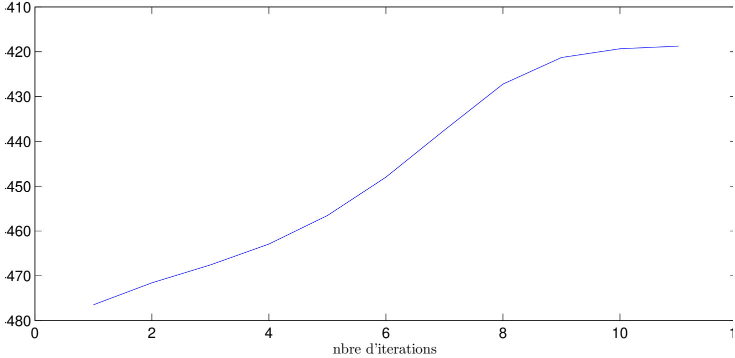
π_1	μ_1	Σ_1	μ_2	Σ_2
0.75	2	2	8	1

- Estimated parameters

$\hat{\pi}_1$	$\hat{\mu}_1$	$\hat{\Sigma}_1$	$\hat{\mu}_2$	$\hat{\Sigma}_2$
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 ⇒ depends on initial parameters

⇒ sensitive to local maxima

⇒ Initialization strategies (e.g K-means or K-means ++)

- ⋆ Potential slow convergence, depending on the distributions

c-component mixtures

Generalization with :

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

$\delta_{ij} = \texttt{true}$ if data x_i is generated by component ϕ_j ($\delta_{ij} = \texttt{false}$ otherwise)

$\Rightarrow$ Responsibility γ_{ij} : expectation of δ_{ij} over all the components
Parameters $\theta = [\pi_j, \theta_j]_{j \in [\![c]\!]} = [\pi_j, [\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j]]_j$ to estimate

EM algorithm, c components

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

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

3. M-step Estimations of mixture parameters φ

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} \gamma_{ij} \boldsymbol{x}_i}{\sum_i \gamma_{ij}}; \quad \boldsymbol{\Sigma}_j = \frac{\mathbf{X}_j \Gamma_j \mathbf{X}_j^T}{\mathsf{Tr}(\Gamma_j)}; \quad \pi_j = \frac{\sum_i \gamma_{ij}}{N}$$

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

4. Iterate 2. and 3. until convergence

Modeling

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

⇒ Parameters

1. c: number of components
2. $\mathbf{\Sigma}_j$: 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

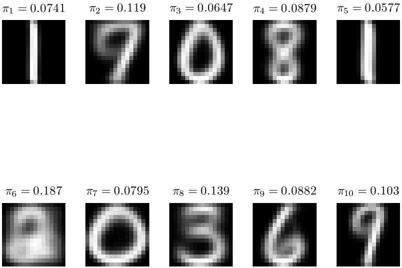
Over-parameterized problem

- ⋆ $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$
- ⋆ e.g: character recognition $\boldsymbol{x}_i \in \mathbb{R}^{256}$
- ⇒ Needs to estimate $256^2 \times c$ parameters for the covariance (given about 7000 data points)!

Matrix parameterization

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

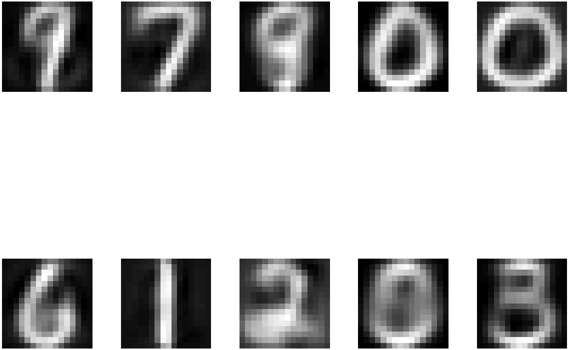
Recall : 50 principal components reconstruct 90% of the signal

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

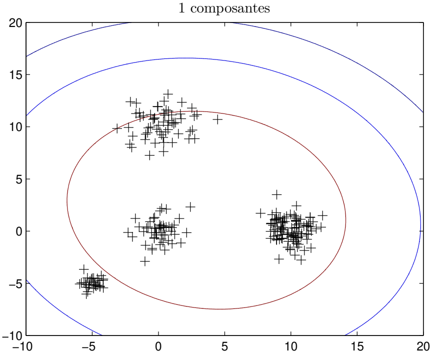
- The larger c, the less points may be assigned to every component (in average)
- Search for parsimonious models, i.e small number of parameters to estimate

Bayesian Information Criterion (BIC)

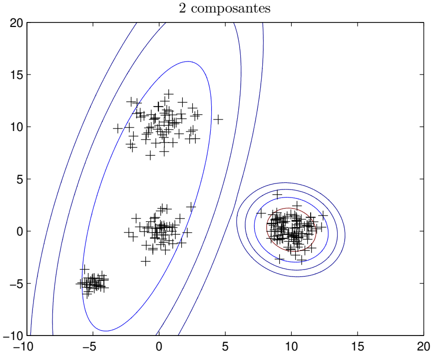
- The larger c is, the better the estimate of l
- Trade likelihood against complexity

$$\mathrm{BIC}(\theta, \mathcal{X}) = -2\mathbb{L}(\theta, \mathcal{X}) + |\theta| \log(N) \qquad \hat{\theta} \overset{\text{here}}{=} \operatorname*{argmin}_{\theta} \mathrm{BIC}(\theta, \mathcal{X})$$

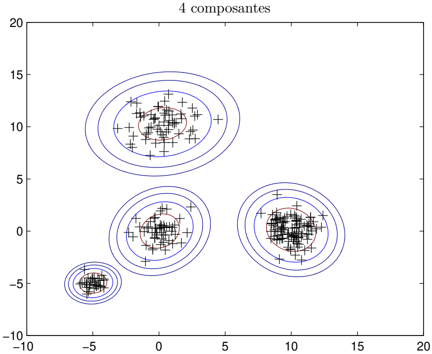
Example: 4-component mixture



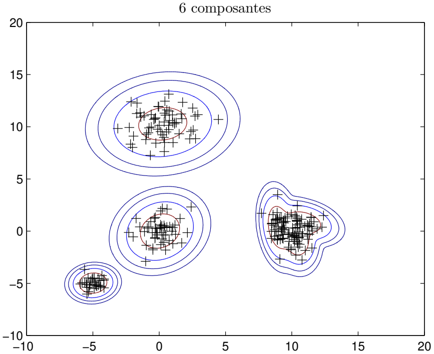
Example: 4-component mixture



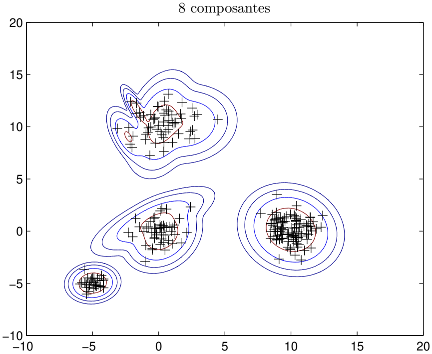
Example: 4-component mixture



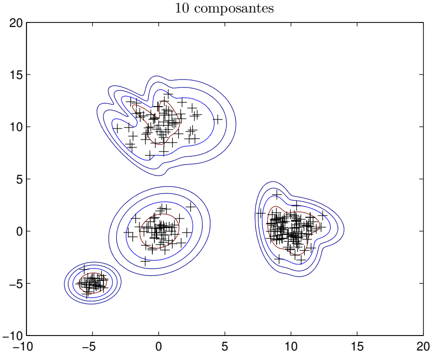
Example: 4-component mixture



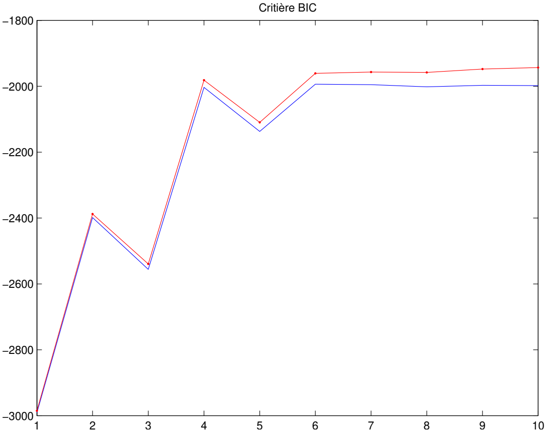
Example: 4-component mixture



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 x_i is associated to class j for which $\mathbb{P}(C = j|x_i) \approx \pi_j \phi_j(x_i)$ 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?

φ 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).

License