

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

Motivation

Data modeling
 Gaussian mixture

EM algorithm

Modeling

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 \rightarrow Variance as a criterion on centered data (Normal distribution)
 - Discriminant models \rightarrow Variance of projected data for within- and between-class models

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

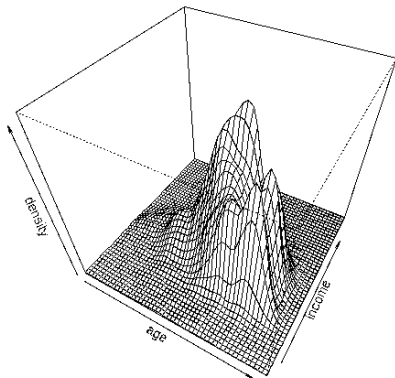
- ★ Alternatively we can search a model for data density
- ★ Let $f_{\theta}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ be the data density

Density estimation

Methods

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

Density estimation: perspective plot



Mixture models

Definition

- ★ The density $f(\mathbf{x})$ is generated by c “basis” functions ϕ (components), from a family \mathbb{F}

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

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

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

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

We get (by identification):

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

Reminder: Maximum log-likelihood (MLE)

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

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

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

- ★ Log-likelihood

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

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

Gaussian mixture

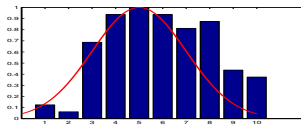
ϕ is a probability density function \rightarrow choosing the (agnostic) normal density family as basis ($\phi \in \mathbb{F} = \{\mathcal{N}(\mu, \Sigma)\}$) seems reasonable:

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

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

Basic case : 1 component, $c = 1$

$$\star \theta = [\mu, \Sigma]$$



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

$$\Longleftrightarrow$$

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

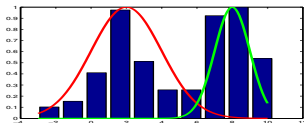
$$\Rightarrow \hat{\mu} = \frac{1}{N} \sum_i \mathbf{x}_i$$

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



A bit more complex: 2 components

$$\begin{aligned}\theta &= [\pi_1, \theta_1, \pi_2, \theta_2] \\ &= [\pi_1, [\mu_1, \Sigma_1], \pi_2, [\mu_2, \Sigma_2]] \\ &\text{with } \pi_1 + \pi_2 = 1\end{aligned}$$



$$\mathbb{L}(\theta, \mathcal{X}) = \sum_i^N \log [\pi_1 \phi(\mathbf{x}_i, \theta_1) + \pi_2 \phi(\mathbf{x}_i, \theta_2)]$$

- ⇒ difficult to maximize because of the sum inside the \log !
- ⇒ Solution : iterative 2 steps (E-M) algorithm to maximize \mathbb{L}
- **Expectation-Maximization** (E-M) algorithm

Estimation (2 components)

★ What is missing here is the assignment of \mathbf{x}_i to one of the 2 components ϕ_j

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

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

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

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

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

Expectation (E-step)

$$\theta = [\pi_1, \theta_1, \pi_2, \theta_2] = [\pi_1, [\mu_1, \Sigma_1], \pi_2, [\mu_2, \Sigma_2]]$$

- ★ 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 ϕ_j (parameterized by θ_j^0):

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

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

- ★ γ_{ij} is the **responsibility**.
 - ★ It is the likelihood that \mathbf{x}_i is (purely) modeled by component ϕ_j
- $\Rightarrow \gamma_{ij}$ represents the part of \mathbf{x}_i that is modeled (generated) by component ϕ_j

Responsibility and soft-assignment

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

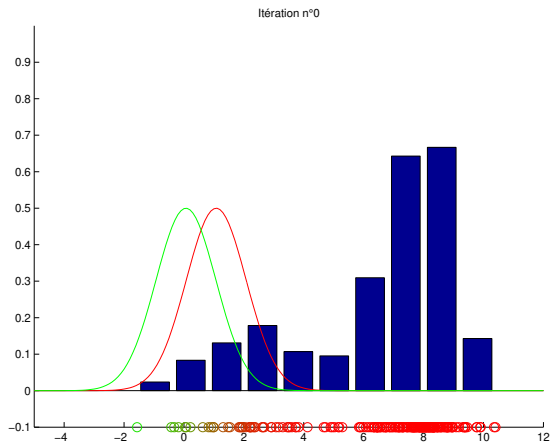
- ★ Note: $\gamma_{i1} + \gamma_{i2} = 1$
- ★ Given the responsibility of every data (γ_{ij}), we can estimate the parameters of every component by (weighted) maximum likelihood:

$$\begin{aligned}\hat{\mu}_1 &= \sum_{i=1}^N \frac{\gamma_{i1}}{\sum_{k=1}^N \gamma_{k1}} \mathbf{x}_i & \hat{\Sigma}_1 &= \sum_{i=1}^N \frac{\gamma_{i1}}{\sum_{k=1}^N \gamma_{k1}} (\mathbf{x}_i - \hat{\mu}_1)(\mathbf{x}_i - \hat{\mu}_1)^T \\ \hat{\mu}_2 &= \sum_{i=1}^N \frac{\gamma_{i2}}{\sum_{k=1}^N \gamma_{k2}} \mathbf{x}_i & \hat{\Sigma}_2 &= \sum_{i=1}^N \frac{\gamma_{i2}}{\sum_{i=1}^N \gamma_{i2}} (\mathbf{x}_i - \hat{\mu}_2)(\mathbf{x}_i - \hat{\mu}_2)^T\end{aligned}$$

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

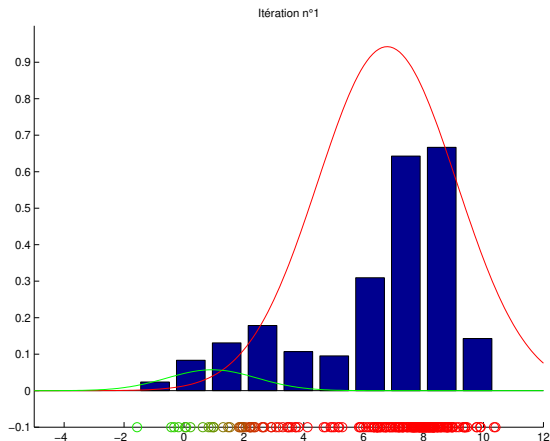
Example

Successive iterations of E- and M-steps



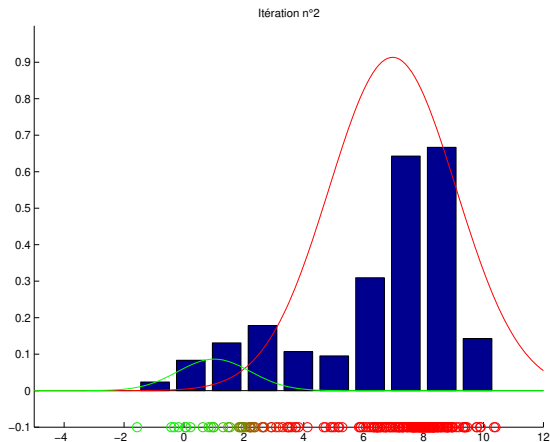
Example

Successive iterations of E- and M-steps



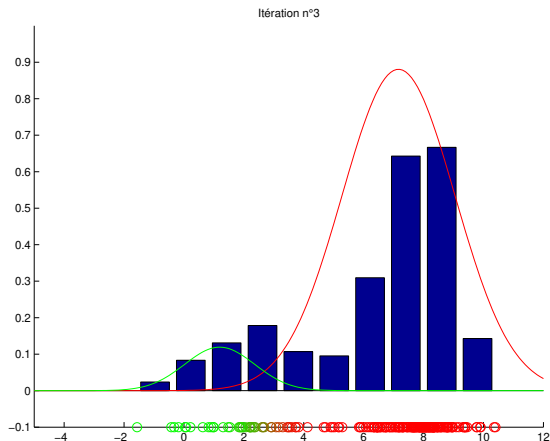
Example

Successive iterations of E- and M-steps



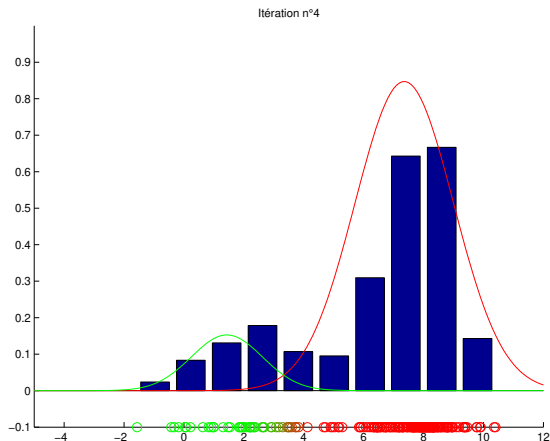
Example

Successive iterations of E- and M-steps



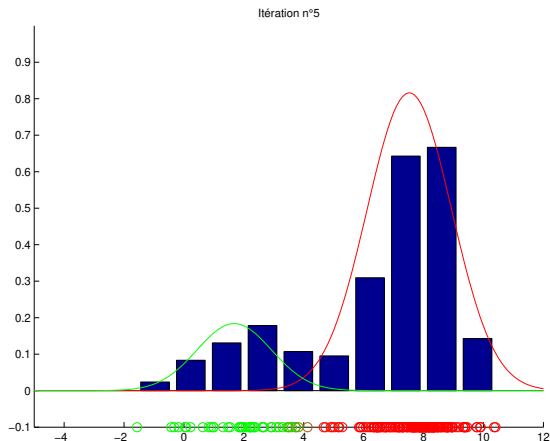
Example

Successive iterations of E- and M-steps



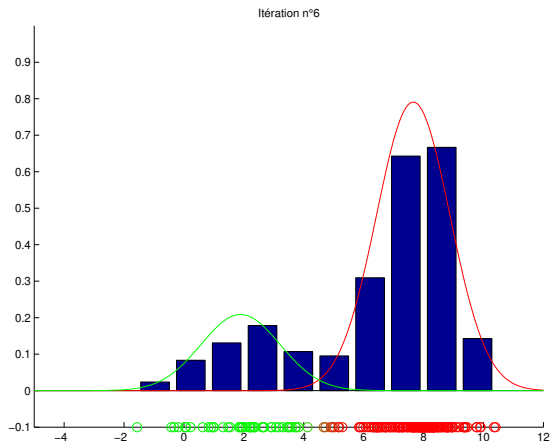
Example

Successive iterations of E- and M-steps



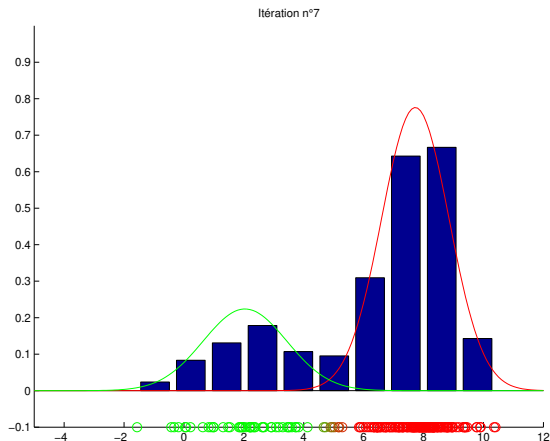
Example

Successive iterations of E- and M-steps



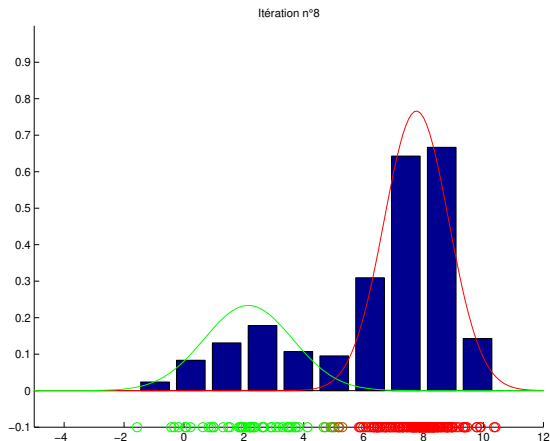
Example

Successive iterations of E- and M-steps



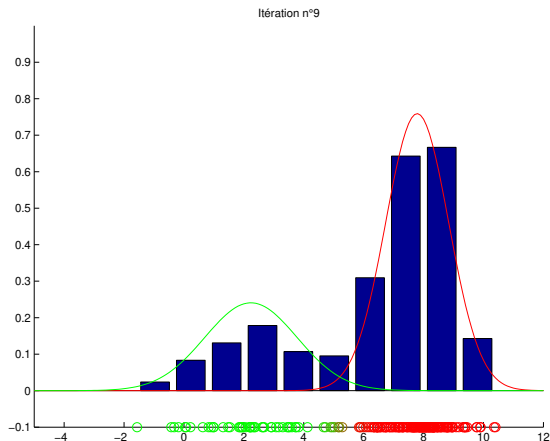
Example

Successive iterations of E- and M-steps



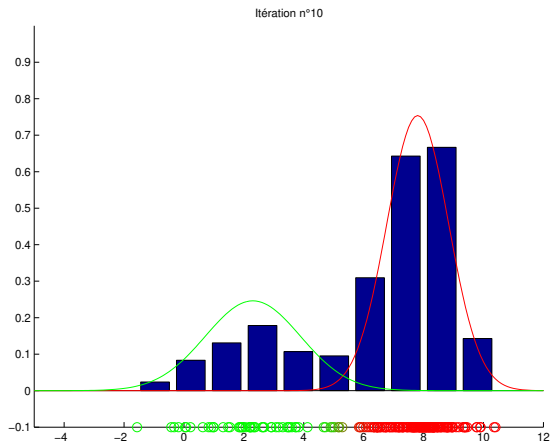
Example

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

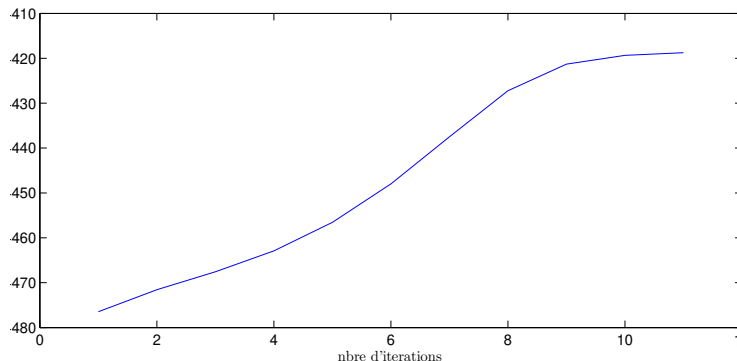
0.76	2.17	1.56	7.91	1.06
------	------	------	------	------

20 iterations

0.76	2.15	1.98	8.01	0.98
------	------	------	------	------

Iterations

Alternate cycle Expectation-Maximization → 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 :

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

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

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

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

EM algorithm, c components

1. Initial $\theta^0 = \{\pi_j^0, \mu_j^0, \Sigma_j^0\}_{j \in \llbracket c \rrbracket}$
 - In general: $\pi_j = 1/c$, μ_j is chosen at random and $\Sigma_k = \text{Id}$
 - Alternative: use K-means as initialization
2. **E-step** : compute responsibilities for every data $i \in \llbracket N \rrbracket$ and every component $j \in \llbracket c \rrbracket$

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

3. **M-step** Estimations of mixture parameters



$$\mu_j = \frac{\sum_{i=1}^N \gamma_{ij} \mathbf{x}_i}{\sum_i \gamma_{ij}}; \quad \Sigma_j = \frac{\mathbf{X}_j \Gamma_j \mathbf{X}_j^T}{\text{Tr}(\Gamma_j)}; \quad \pi_j = \frac{\sum_i \gamma_{ij}}{N}$$

with $\Gamma_j = \text{diag}(\gamma_{1j}, \dots, \gamma_{Nj})$, \mathbf{X}_j centered on μ_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. 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 → over-parameterized

Shape of the covariance matrix

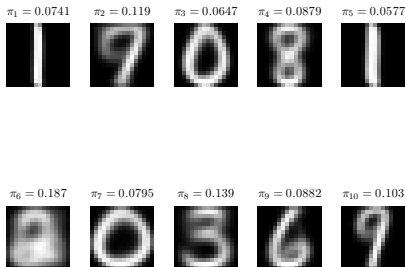
Over-parameterized problem

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

Matrix parameterization

- ★ Spherical models $\Sigma = \sigma \mathbf{I}$ → 1 parameter
- ★ Diagonal models $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_D)$ → D parameters
- ★ Full models $\Sigma \in \mathbb{R}^{D \times D}$ → $O(D^2)$ parameters

Character recognition



More complex models → 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

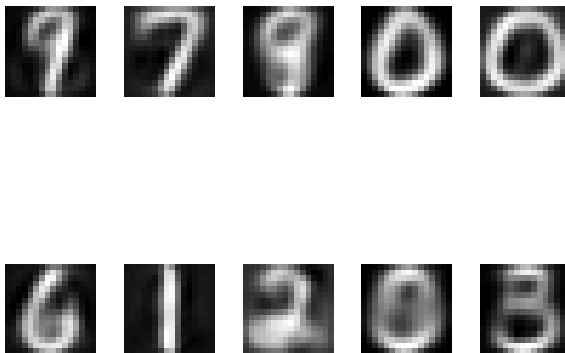
EM within the space of the 10 first PC



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

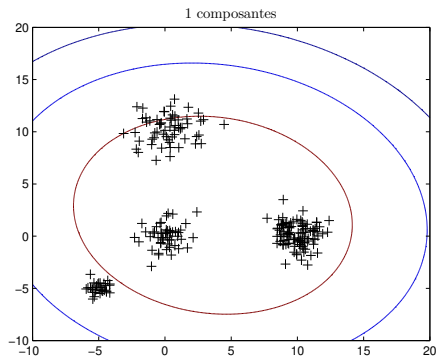
- ★ The larger c , the less points may be assigned to every component (in average)
- ★ Search for **parsimonious** models, i.e small number of parameters to estimate

Bayesian Information Criterion (BIC)

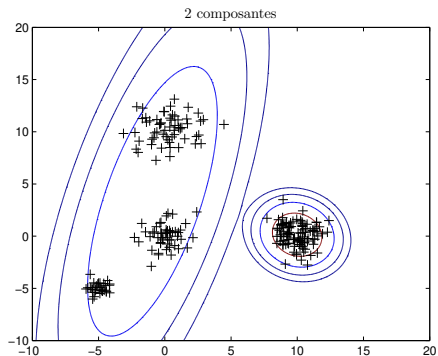
- ★ The larger c is, the better the estimate of l
- ★ Trade likelihood against complexity

$$\text{BIC}(\theta, \mathcal{X}) = -2\mathbb{L}(\theta, \mathcal{X}) + |\theta| \log(N) \quad \hat{\theta} \stackrel{\text{here}}{=} \underset{\theta}{\operatorname{argmin}} \text{BIC}(\theta, \mathcal{X})$$

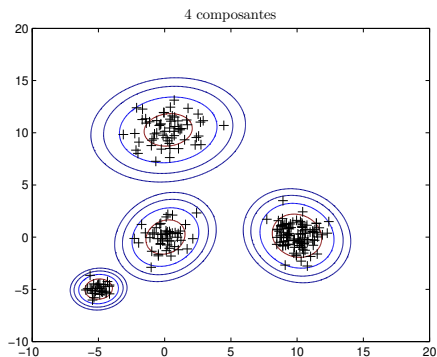
Example: 4-component mixture



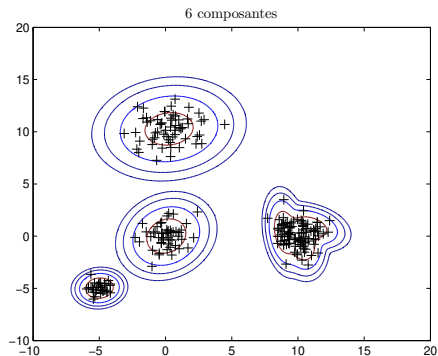
Example: 4-component mixture



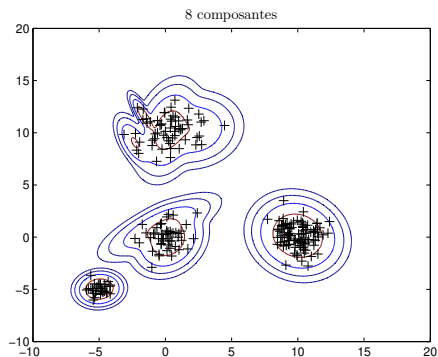
Example: 4-component mixture



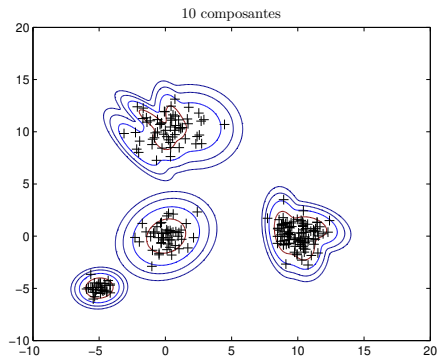
Example: 4-component mixture



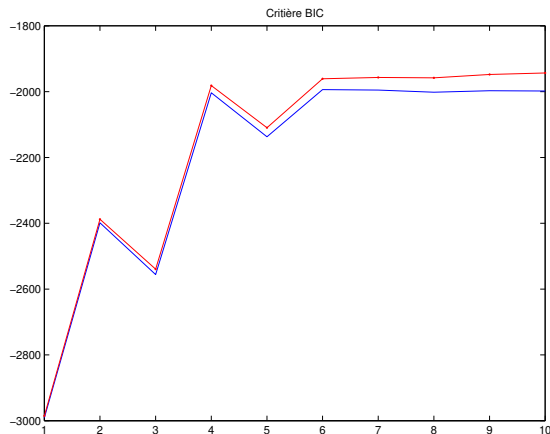
Example: 4-component mixture



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 \mathbf{x}_i is associated to class j for which $\mathbb{P}(C = j|\mathbf{x}_i) \approx \pi_j \phi_j(\mathbf{x}_i)$ 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 between GMM and **K-means**
- ★ What are hard- and soft-assignments?

Ⓢ It is strongly advised 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).