Data Science
Static data analysis

Stéphane Marchand-Maillet

Master en Sciences Informatiques - Semestre d'Automne

Data modelling

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

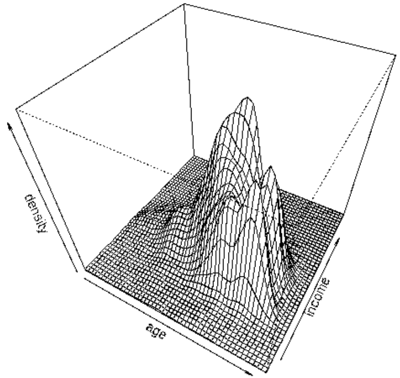
⇒ The distribution is fixed (essentially normal) and we look for its parameters (μ, σ)

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

Density estimation

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

Density estimation: perspective plot



Mixture models

Definition

- ⋆ The density $f(x)$ is generated by c “basis” functions (components)

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

- ⋆ π_j are the mixture parameters
- ⋆ $\phi(x, \boldsymbol{\theta}_j)$ are functions controled by parameters $\boldsymbol{\theta}_j$

Hypotheses

1. The number of components (c) is known
2. The family of functions ϕ is known
3. Labels (classes) are unknown

Probabilist reading

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

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

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

- ⋆ We get $\pi_j = P(\omega_j)$ and $\sum_j \pi_j = 1$
- ⋆ $\phi(x, \boldsymbol{\theta}_j)$ is the conditional probabilty that x is generated by ω_j

Gaussian mixture

ϕ is a probability density function. Chosing the (agnostic) normal law as basis $\mathcal{N}(\mu, \Sigma)$ seems reasonable:

$$f(x) = \sum_{j=1}^{c} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$$

- ⋆ Can approximate any density
- ⋆ Enables a linear system of its parameters by maximising the log-likelihood

Maximum log-likelihood (ML)

- Given $\Omega = \{x_1, \ldots, x_N\}$ unlabeled samples generated by the mixture $f(x) = p(x|\boldsymbol{\theta})$.
- $\boldsymbol{\theta} = \{\pi_j, \mu_j, \Sigma_j\}$ atre the parameters to infer.
- Likelihood :

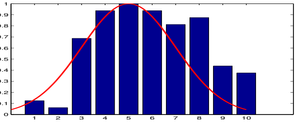
$$p(\Omega|\boldsymbol{\theta}) = \prod_{i}^{N} p(x_i|\boldsymbol{\theta})$$

- Estimation : $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\Omega|\boldsymbol{\theta})$
- or maximimum log-likelihood

$$l(\boldsymbol{\theta}, \Omega) = \sum_{i=1}^{N} \log p(x_i|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{c} \pi_j \phi(x_i, \boldsymbol{\theta}_j) \right]$$

Basic case : 1 component, $c = 1$

- ⋆ $\boldsymbol{\theta} = \{\mu, \Sigma\}$



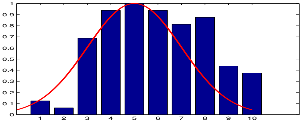
$$\max_{\boldsymbol{\theta}} \sum_i \log e^{-(x_i-\mu)^\mathsf{T}\Sigma^{-1}(x_i-\mu)}$$

$$\Longleftrightarrow$$

$$\min_{\boldsymbol{\theta}} \sum_i (x_i-\mu)^\mathsf{T}\Sigma^{-1}(x_i-\mu)$$

Basic case : 1 component, $c = 1$

- $\boldsymbol{\theta} = \{\mu, \Sigma\}$



$$\max_{\boldsymbol{\theta}} \sum_i \log e^{-(x_i-\mu)^\mathsf{T}\Sigma^{-1}(x_i-\mu)}$$

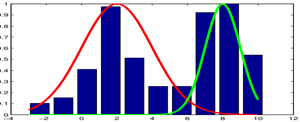
$$\iff$$

$$\min_{\boldsymbol{\theta}} \sum_i (x_i - \mu)^\mathsf{T}\Sigma^{-1}(x_i - \mu)$$

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

A bit more complex: 2 components

$$\boldsymbol{\theta} = \{\pi, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$$
$$= \{\pi_1, \pi_2, \mu_1, \Sigma_1, \mu_2, \Sigma_2\}$$



$$l(\boldsymbol{\theta}, \Omega) = \sum_i^N \log\left[(1-\pi)\phi(x_i, \boldsymbol{\theta}_1) + \pi\phi(x_i, \boldsymbol{\theta}_2)\right]$$

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

$\Rightarrow$ Solution : iteratif 2 steps (E-M) algorithm to maximise l

$\rightarrow$ Expectation-Maximisation (EM) algorithm

EM algorithm (2 components)

- ⋆ The unknown here is the assignement x_i to one of the 2 components ϕ_i
- ⇒ If we knew it, we would treat the problem as twice 1 component
- ⋆ The EM algorithm introduces unknown variables : the assignement $\Delta_i \in \{0, 1\}$ of every x_i to one component :

$$x_i \backsim \phi_1 \text{ if } \Delta_i = 0, x_i \backsim \phi_2 \text{ if } \Delta_i = 1$$

Expectation (E-step)

- Assume we know an initial value for $\boldsymbol{\theta}^0$
- We can infer the contribution of every data x_i to every density (parameterized by θ_i^0):

$$\gamma_i(\boldsymbol{\theta}^0) = E[\Delta_i | \boldsymbol{\theta}^0, \Omega]$$
$$= \frac{\pi\phi(x_i, \boldsymbol{\theta}_2^0)}{(1-\pi)\phi(x_i \boldsymbol{\theta}_1^0) + \pi\phi(x_i, \boldsymbol{\theta}_2^0)}$$

- γ_i is the responsability.
- It is the expectation of Δ_i over all components

Responsability and soft-assignment

γ_i allows to determine $\Delta_i \Rightarrow x_i$ can be assigned to either ϕ_1 or ϕ_2

$\Rightarrow$ K-means-type hard-assignement. Each data is assigned to one and only one cluster

EM is "softer". A data may contribute (via γ_i) to several density modes (clusters). EM computes a soft-assignment (with $\sum_i \gamma_i = 1$)

Maximisation (M-step)

* Given every data responsability (γ_i), we can estimate the parameters by (weighted) maximum likelihood:

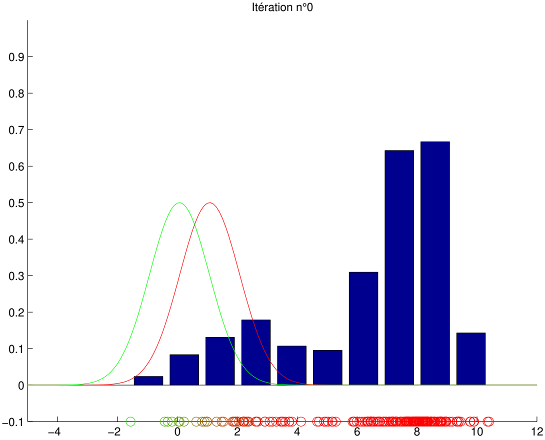
$$\hat{\mu}_1 = \frac{\sum_{i=1}^{N}(1-\gamma_i)x_i}{\sum_{i=1}^{N} 1-\gamma_i} \qquad \hat{\Sigma}_1 = \frac{\sum_{i=1}^{N}(1-\gamma_i)(x_i-\mu_1)(x_i-\mu_1)^{\mathsf{T}}}{\sum_{i=1}^{N} 1-\gamma_i}$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{N}\gamma_i x_i}{\sum_{i=1}^{N}\gamma_i} \qquad \hat{\Sigma}_2 = \frac{\sum_{i=1}^{N}\gamma_i(x_i-\mu_1)(x_i-\mu_1)^{\mathsf{T}}}{\sum_{i=1}^{N}\gamma_i}$$

* Proportion for mixture 1: $\pi = \sum_{i=1}^{N}\gamma_i/N$

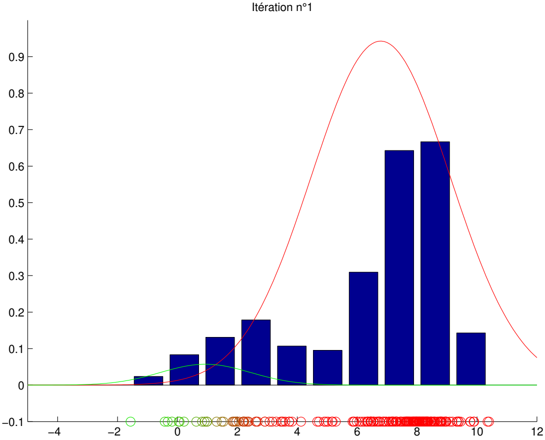
Illustration

Successive iterations of E- and M-steps



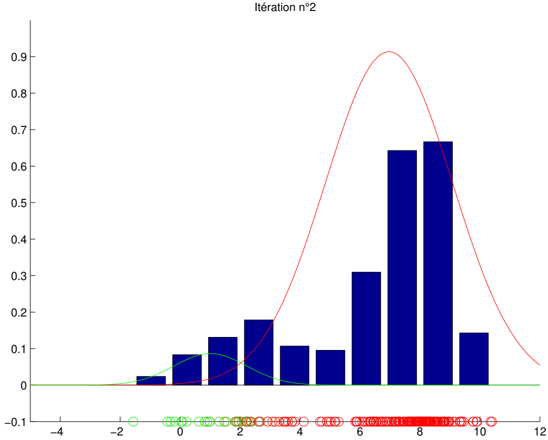
Illustration

Successive iterations of E- and M-steps



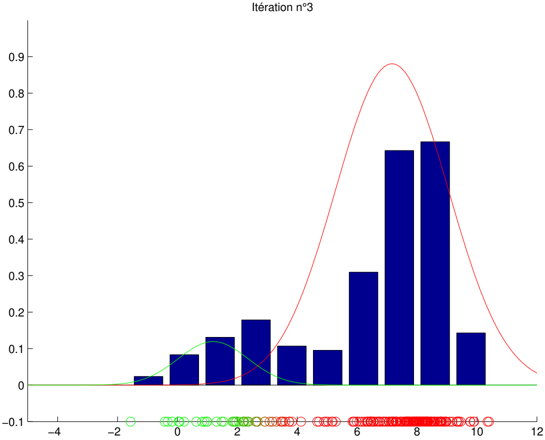
Illustration

Successive iterations of E- and M-steps



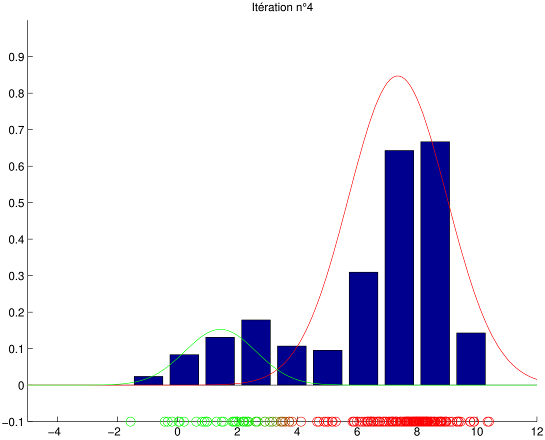
Illustration

Successive iterations of E- and M-steps



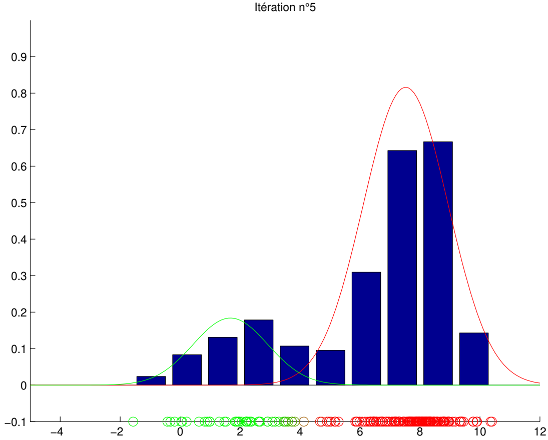
Illustration

Successive iterations of E- and M-steps



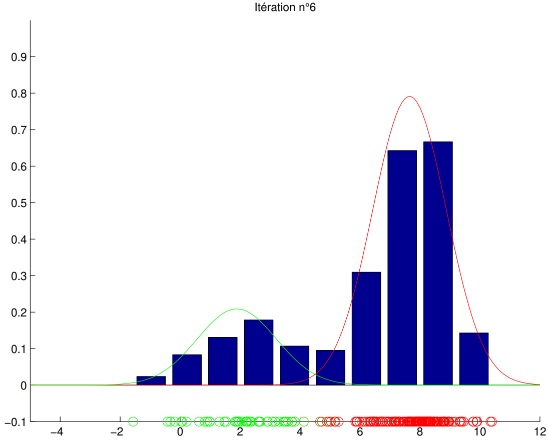
Illustration

Successive iterations of E- and M-steps



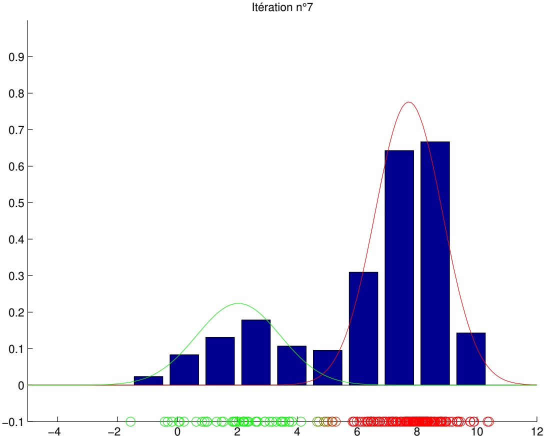
Illustration

Successive iterations of E- and M-steps



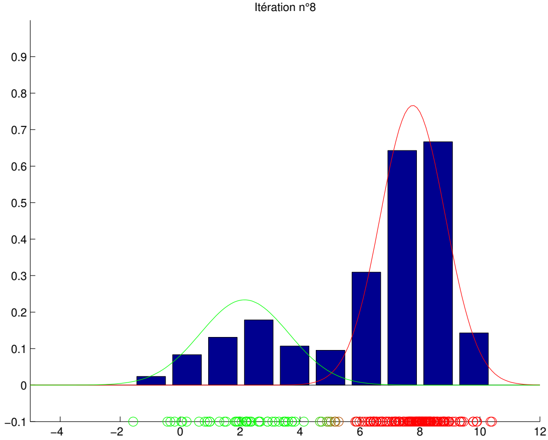
Illustration

Successive iterations of E- and M-steps



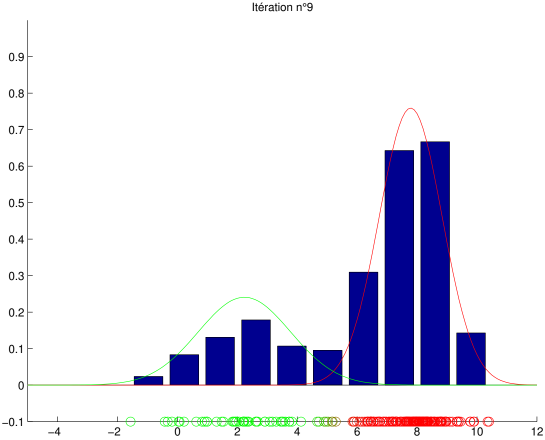
Illustration

Successive iterations of E- and M-steps



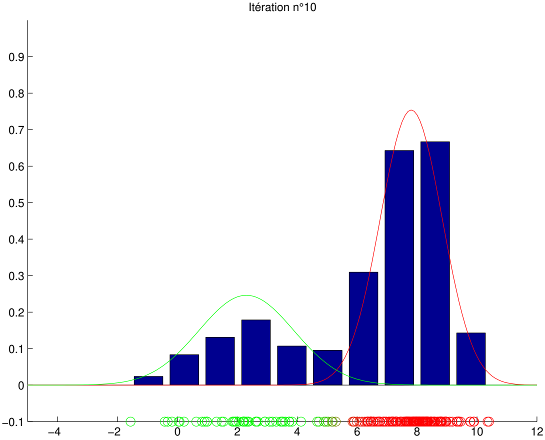
Illustration

Successive iterations of E- and M-steps



Illustration

Successive iterations of E- and M-steps



Results

⋆ True parameters

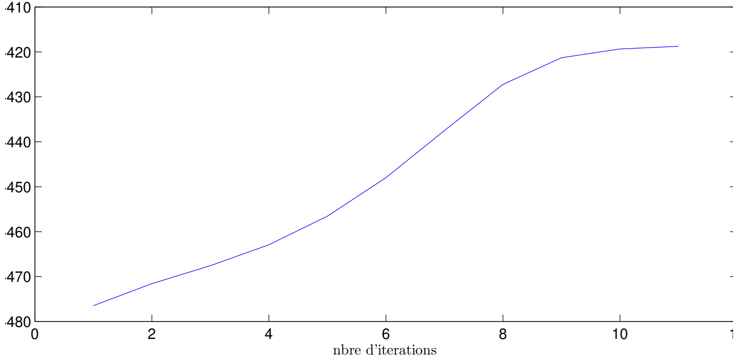
π	μ_1	σ_1	μ_2	σ_2
0.75	2	2	8	1

⋆ Estimated parameters

$\hat{\pi}$	$\hat{\mu_1}$	$\hat{\sigma_1}$	$\hat{\mu_2}$	$\hat{\sigma_2}$
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-Maximisation $\rightarrow$ increases the likelihood of the data w.r.t mixture model



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

Limitations

- ⋆ Hill-climbing $\Rightarrow$ depends on initial parameters
- ⋆ Potential slow convergence, depending on the distributions
- ⋆ Hill-climbing $\Rightarrow$ sensitive to local maxima

c-component mixtures

Generalisation with :

$$\Delta_{i1}, \Delta_{i2}, \dots, \Delta_{ik}, \dots, \Delta_{ic}$$

True (=1) if data x_i is generated by component ϕ_k

$\Rightarrow$ Responsability γ_{ik} : expectation of Δ_{ik} over all the components

$\{\pi_k, \mu_k, \Sigma_k\}_{k=1,\dots,c}$ unknown parameters (before: $\pi_1 = \pi$, $\pi_2 = 1 - \pi$)

EM algorithm, c components

1. Initial $\boldsymbol{\theta}^0 = \{\pi_k^0, \mu_k^0, \Sigma_k^0\}_{k=1,\dots,c}$
   - In general $\pi_k = 1/c$, μ_k is chosen at random and $\Sigma_k = \mathbf{Id}$
   - Alternative : use k-means as initialisation
2. E-step : compute responsabilities for every data $i = 1, \dots, N$ and every component $k = 1, \dots, c$

$$\gamma_{ik} = \frac{\pi_k \phi(x_i, \boldsymbol{\theta}_k)}{\sum_{j=1}^{c} \pi_j \phi(x_i, \boldsymbol{\theta}_j)}$$

3. M-step Estimations of mixture parameters

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} x_i}{\sum_i \gamma_{ik}}; \qquad \Sigma_k = \frac{X_k \Gamma_k X_k^{\mathsf{T}}}{\mathrm{Tr}(\Gamma_k)}; \qquad \pi_k = \frac{\sum_i \gamma_{ik}}{N}$$

with $\Gamma_k = \mathrm{diag}[\gamma_{1k}, \dots, \gamma_{Nk}]$, X_k centered on μ_k

4. Iterate 2. and 3. until convergence

Modeling

- ⋆ The *a priori* parametrisation of the mixture changes the convergence
- ⋆ Parameters
  1. c: number of components
  2. Σ_k: the shape of covariance matrices (diagonal, full, parameterised)
- ⋆ Too flexible or too rigid models mean wrong or no convergence...
- ⋆ Number of variables : $p \times p \times c + 2 \times c$: if N low, p large and c large $\rightarrow$ over-parameterised

Shape of the covariance matrix

Over-parameterised problem

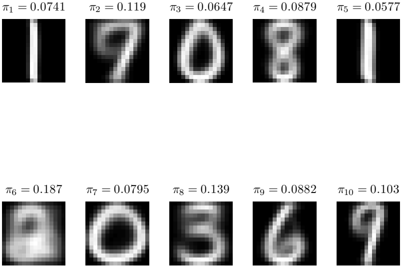
- $\star$ $\Sigma \in \mathbb{R}^{p \times p}$
- $\star$ Eg: character recognition $\mathbf{x}_i \in \mathbb{R}^{256}$

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

Matrix parameterisation

- $\star$ Spherical models $\Sigma = \sigma * \mathbf{Id}$, 1 parameter
- $\star$ Diagonal models $\Sigma = \text{diag}[\sigma_1, \ldots, \sigma_p]$, p parameters
- $\star$ Full models $\Sigma \in \mathbb{R}^{p \times p}$, p^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

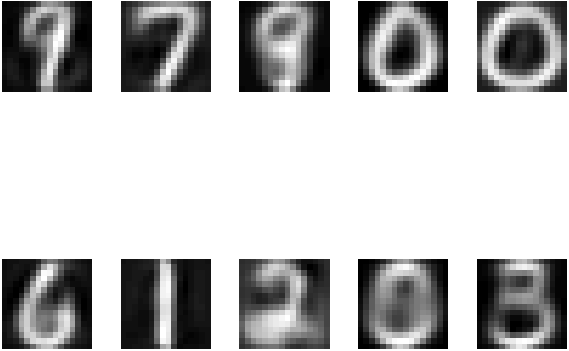
Recall : 50 principal components reconstruct 90% of the signal

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

Parcimony

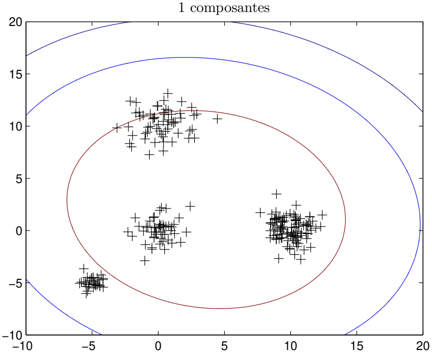
- ⋆ The larger c, the less points may be assigned to every component (in average)
- ⋆ Search for parcimonious models, ie small number of parameters to estimate

Bayesian Information Criterion (BIC)

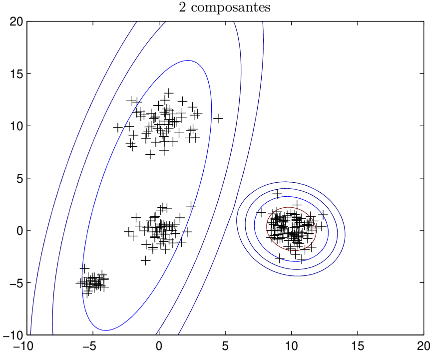
- ⋆ The larger c is, the better the estimate of l
- ⋆ Trade likelihood against complexity

$$BIC = 2 * l(\theta) - |\theta|.\log(N)$$

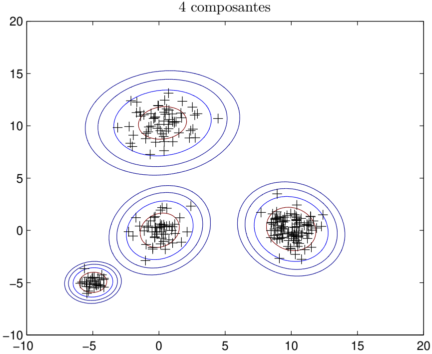
Example: 4-component mixture



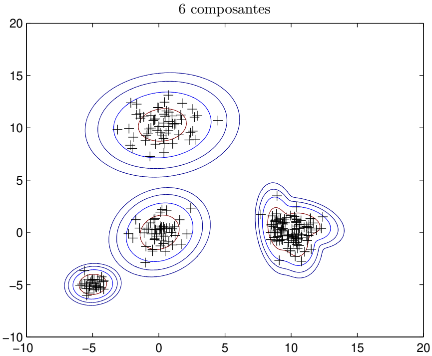
Example: 4-component mixture



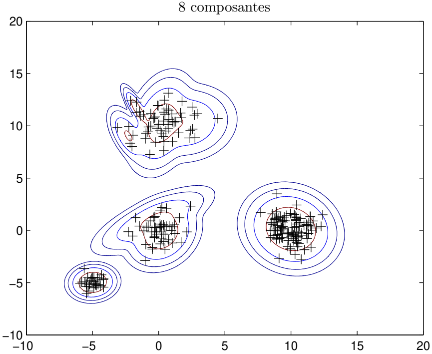
Example: 4-component mixture



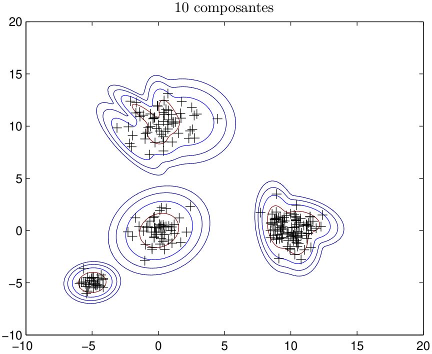
Example: 4-component mixture



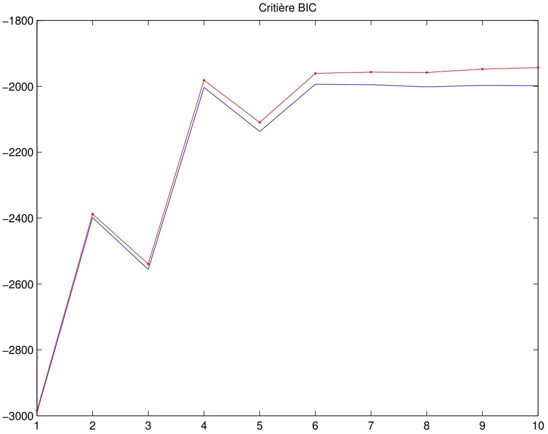
Example: 4-component mixture



Example: 4-component mixture



Example (cont'd)



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

Conclusions

Gaussian mixtures

- ⋆ The Gaussian mixture model generalises the underlying data hypothesis made by PCA and LDA
- ⋆ Explicit density modeling and estimation
- ⋆ Also classification (unsupervised): if components are classes, data point i is associated to class k for which $p(k|x_i) \approx \pi_k \phi_k(x_i)$ is maximised amongst all classes

Conclusions

EM algorithm

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