# Introduction to Probabilistic Graphical Models and Deep Generative Models

P. Latouche

**PR UCA**
Ecole Polytechnique / ENS Paris Saclay
pierre.latouche@math.cnrs.fr
https://lmbp.uca.fr/~latouche/
https://lmbp.uca.fr/~latouche/mva/IntroductiontoProbabilisticGraphicalModelsMVA.html

ENS Paris Saclay, Master MVA

# Part I

## Lecture 2: K-means
## EM
## Gaussian mixtures

# Outline Part 1

Clustering

Gaussian mixture models

EM algorithm

Model selection

# Introduction to clustering

- we are provided with a data set $(x_i)_i = (x_1, \ldots, x_n)$ (sample of size $n$) with $x_i \in \mathbb{R}^d$
- goal : retrieve groups $=$ clusters $=$ classes of individuals where :
  - 2 individuals within a group must be as similar as possible
  - 2 individuals of different groups must be as different as possible
- unsupervised learning setting : no target variable
- we aim at uncovering (learning) what is hidden in the data set

Types of clusters :

- disjoint
- hierarchical
- overlapping

**Def**

A partition of a data set $(x_i)_i$ into $K$ clusters $\mathcal{P} = (C_k)_k = (C_1, \ldots, C_K)$ verifies :

1. $\cup_{k=1}^{K} C_k = (x_i)_i$
2. $C_k \cap C_l = \emptyset, \forall k \neq l$

So each observation is clustered into a unique cluster

## Def

The $n$th Bell number $B_n$ counts the number of different ways to partition a set that has exactly $n$ elements

## Theorem

The Bell numbers satisfy the following recurrence relation :

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

ex : $B_{18} = 682\,076\,806\,159$

## Dobinski's formula

$$B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!}$$

- so exact clustering is a combinatorial problem (NP-hard)
- two most famous heuristics : kmeans and hierarchical clustering
- the statistical point of view : mixture models and expectation maximisation

# Inertia

## Def

The total inertia of a data cloud of points in $\mathbb{R}^d$ (sample of observations) is :

$$S = \frac{1}{n} \sum_{i=1}^{n} ||x_i - \bar{x}||^2,$$

where $\bar{x} = (1/n) \sum_{i=1}^{n} x_i$ is the barycentre of the data cloud (= empirical mean of the sample)

## Remark

The total inertia can be written :

$$S = \sum_{j=1}^{d} \hat{\sigma}_j^2,$$

where $\hat{\sigma}_j^2 = (1/n) \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2$ is the empirical (biased) variance of variable $j$

## Def

The intra class inertia of a data cloud of points in $\mathbb{R}^d$ (sample of observations), for a partition $\mathcal{P} = (C_k)_k$ with $K$ clusters, is :

$$W = \frac{1}{n} \sum_{k=1}^{K} \sum_{x_i \in C_k} ||x_i - \mu_k||^2,$$

where $\mu_k = (1/n_k) \sum_{x_i \in C_k} x_i$ is the empirical mean of the observations in cluster $C_k$ and $n_k$ is the number of observations in $C_k$

## Def

The inter class inertia of a data cloud of points in $\mathbb{R}^d$ (sample of observations), for a partition $\mathcal{P} = (C_k)_k$ with $K$ clusters, is :

$$B = \frac{1}{n} \sum_{k=1}^{K} n_k ||\mu_k - \bar{x}||^2,$$

## Huygens theorem

$$S = W + B$$

## Remarks

- $S$ does not depend on the partition $\mathcal{P}$ contrary to $W$ and $B$
- so, when $W$ decreases, $B$ increases, and vice versa

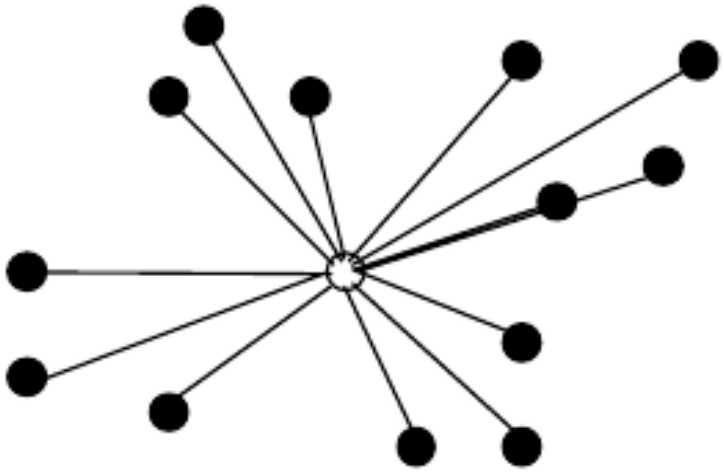
## A clustering task

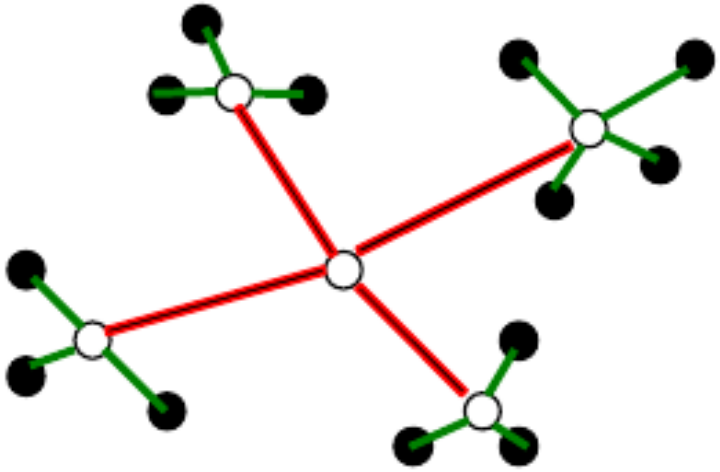
Find $\mathcal{P}$ which minimises $W$ (maximises $B$) with $K < n$

## Remark

When $K = n$ (each observation is in its own cluster), $W = 0 \rightarrow$ useless in practice

# Inertia and Huygens theorem





- again : combinatorial problem (NP hard) (if $K < n$)
- heuristics

# kmeans

The kmeans algorithm focuses directly on the optimisation of $W$, $K$ being fixed

- init : initialise all the (bary)centres $\mu_k$ (at random in $\mathbb{R}^d$ or on random observations)

1. each observation is clustered in the cluster with the closest centre
2. recompute the centres
3. if the $\mu_k$ have moved (no eps convergence) back to 1.

# Pros and cons

kmeans :

+ fast. Complexity : $\mathcal{O}(nK)$
+- dependent on the initialisation
+ easy to parallelise
- fixed $K$

# Outline Part 1

Clustering

Gaussian mixture models

EM algorithm

Model selection

# Gaussian mixture models

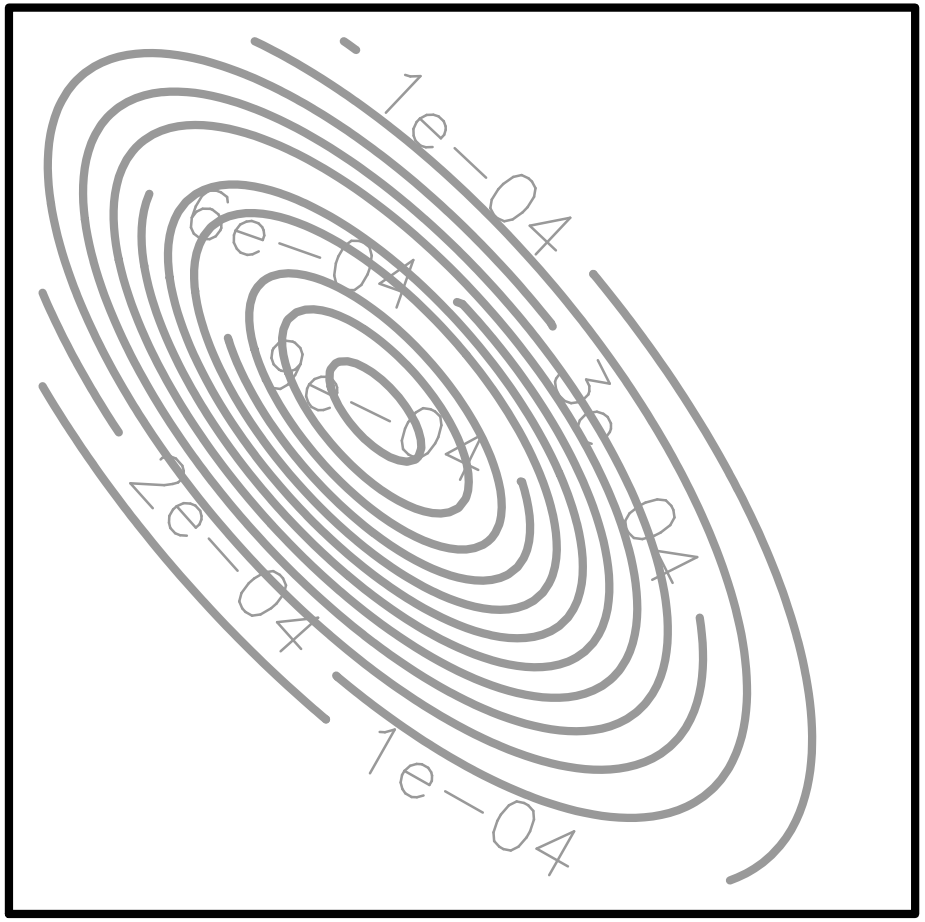
Let us first consider a random sample $(X_1, \ldots, X_n)$ where $X_i \sim \mathcal{N}(\mu, \Sigma)$ (assumed mulvariate Gaussian).

## Property

The MLE for $\mu$ and $\Sigma$ are :

- $\hat{\mu} = \bar{x}$ (empirical mean)
- $\hat{\Sigma} = (1/n) \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^\intercal$ (empirical variance-covariance matrix)

# Multivariate Gaussian density ($d = 2$)



# Mixture of densities

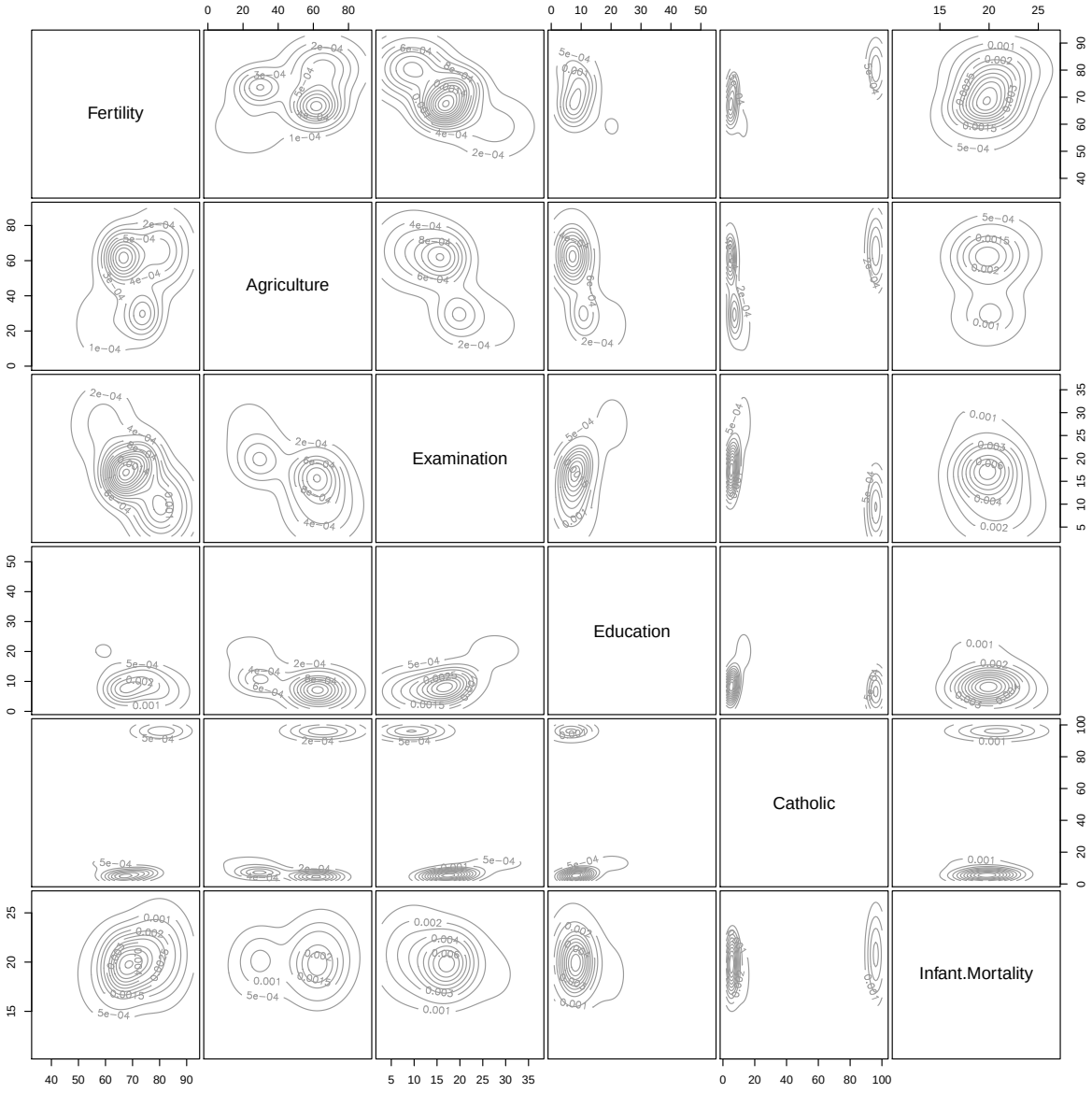



Figure: Analysis of the *swiss* data set with Mclust

## Def

A Gaussian mixture model with $K$ components is defined through the density :

$$p(x|\pi, \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k),$$

where $\theta = (\mu_k, \Sigma_k)_k$ and $\pi = (\pi_1, \ldots, \pi_K)^\intercal$ the vector of mixing weights lies in the standard $K$-simplex :

- $\pi_k \in ]0, 1[, \forall k \in \{1, \ldots, K\}$
- $\sum_{k=1}^{K} \pi_k = 1$

$\mathcal{N}(x; \mu_k, \Sigma_k)$ denotes here the multivariate Gaussian density with parameters $\mu_k$ and $\Sigma_k$ evaluated at $x \in \mathbb{R}^d$

# Starting point

- as usual, we are provided with a random sample $(X_1, \ldots, X_n)$ but now $X_i \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$
- we aim at estimating $\pi$ and the component parameters $\theta$

## Property

The log-likelihood of a Gaussian mixture model is given by :

$$\begin{aligned} L_{(x_1,\ldots,x_n)}(\pi, \theta) &= \sum_{i=1}^{n} \log p(x_i | \pi, \theta) \\ &= \sum_{i=1}^{n} \log \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k) \right) \end{aligned}$$

## Property

- the optimisation task is not trivial
- no analytical expression for the estimators of $\pi$ and $\theta$
- can rely on numerical algorithms for optimisation (conjugate gradient descent for instance) but ...

# Another point of view

The Gaussian mixture model can be rewritten by introducing auxiliary variables :

1. $Z_i \sim \mathcal{M}(1, \pi), \forall i \in \{1, \ldots, n\}$ iid
2. $X_i | Z_{ik} = 1 \sim \mathcal{N}(\mu_k, \Sigma_k)$

So

- $Z_i \in \{0, 1\}^K$ such that $\sum_{k=1}^K Z_{ik} = 1$
- $Z_{ik} = 1$ encodes the fact that observation $i$ is from component $k$
- by definition of the multinomial law : $\mathbb{P}(Z_{ik} = 1) = \pi_k$
- the observations are now sampled conditionally on their components

# Outline Part 1

Clustering

Gaussian mixture models

EM algorithm

Model selection

## Def

Considering the (complete) random sample of couples $((X_1, Z_1) \ldots, (X_n, Z_n))$, the complete data log-likelihood is given by :

$$L_{(x_i,z_i)_i}(\pi, \theta) = \log p((x_i, z_i)_i | \pi, \theta)$$
$$= \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \left(\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)\right)$$

## Property

The estimators of $\pi$ and $\theta$ maximising the complete data log-likelihood are :

- $\hat{\pi}_k = (1/n) \sum_{i=1}^{n} z_{ik}$
- $\hat{\mu}_k = (1/n_k) \sum_{i=1}^{n} z_{ik} x_i$
- $\hat{\Sigma}_k = (1/n_k) \sum_{i=1}^{n} z_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\intercal$

However :

- the $z_i$ are unknown in practice
- this is the clustering information we are looking for
- how estimating the parameters without knowing the clusters ?
- $\rightarrow$ the expectation maximisation (EM) algorithm (DLR77)

## Remark

Link with the kmeans algorithm : $\pi_k = 1/K$ and $\Sigma_k = I_d, \forall k \in \{1, \ldots, K\}$

The EM algorithm relies on two fondamental properties :

## Property

Given the observations (and the parameters), all the $Z_i$ are independent :

$$p((z_i)_i|(x_i)_i, \pi, \theta) = \prod_{i=1}^{n} p(z_i|x_i, \pi, \theta)$$

Recall that $Z_i$ is discrete so $p(z_i|x_i, \pi, \theta)$ translates into $\mathbb{P}(Z_i = z_i|x_i, \pi, \theta)$

## Property

The probabilities $p(z_i|x_i, \pi, \theta)$ have analytical forms :

$$p(z_i|x_i, \pi, \theta) = \mathcal{M}(z_i; 1, \tau_i)$$

where $\tau_i = (\tau_{i1}, \ldots, \tau_{iK})^\intercal$

## Property

$\tau_{ik}$ is given by :

$$\tau_{ik} = \frac{\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(x_i; \mu_l, \Sigma_l)}$$

- $\tau_{ik}$ is the probability for observation $i$ to be in cluster $k$, given $x_i$ and the (current) value of the parameters

## Remark

The $Z_i$ being unknown, they are treated as random vectors in the complete data log-likelihood :

$$L_{(x_i, Z_i)_i}(\pi, \theta)$$

## Remark

The $Z_i$ being unknown, the expectation of the complete data log-likelihood is computed

## Property

The expectation of the complete data log-likelihood is given by :

$$\mathbb{E}_{(Z_i)_i}[L_{(x_i,Z_i)_i}(\pi,\theta)] = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}\log\left(\pi_k\mathcal{N}(x_i;\mu_k,\Sigma_k)\right),$$

where the expectation is taken with respect to the random variables $Z_i \sim \mathcal{M}(1;\tau_i)$

## Property

The estimators of $\pi$ and $\theta$ maximising the expected complete data log-likelihood are :

- $\hat{\pi}_k = (1/n) \sum_{i=1}^{n} \tau_{ik}$
- $\hat{\mu}_k = (1/n_k) \sum_{i=1}^{n} \tau_{ik} x_i$
- $\hat{\Sigma}_k = (1/n_k) \sum_{i=1}^{n} \tau_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\intercal$

# EM algorithm for Gaussian mixture models

- init : initialise the clusters (with kmeans for instance)

M compute $\hat{\pi}$ and $\hat{\theta}$ with respect to the $\tau_i$

E compute the $\tau_i$ with respect to $\pi$ and $\theta$

- if the log-likelihood has changed (or the parameters) (no eps convergence) back to M.

## Remarks

The parameters can also be initialised (instead of the clusters) through a sampling. In that case, the algorithm starts with the E step

## Property

- the EM iteration does increase the log-likelihood $L_{(x_i)_i}(\pi, \theta)$
- in general, no guarantee to converge to the global maximum

# Outline Part 1

Clustering

Gaussian mixture models

EM algorithm

Model selection

# Model selection

In order to estimate the number $K$ of components from the data, the EM algorithm is run for various values of $K$ and the one maximising a criterion is chosen :

- $M_K$ denoting the total number of (free) parameters in the model with $K$ components
- Bayesian information criterion : $\mathrm{BIC}(K) = L_{(x_i)_i}(\hat{\pi}, \hat{\theta}) - (M_K/2)\log n$
- Akaike's information criterion : $\mathrm{AIC}(K) = L_{(x_i)_i}(\hat{\pi}, \hat{\theta}) - M_K$

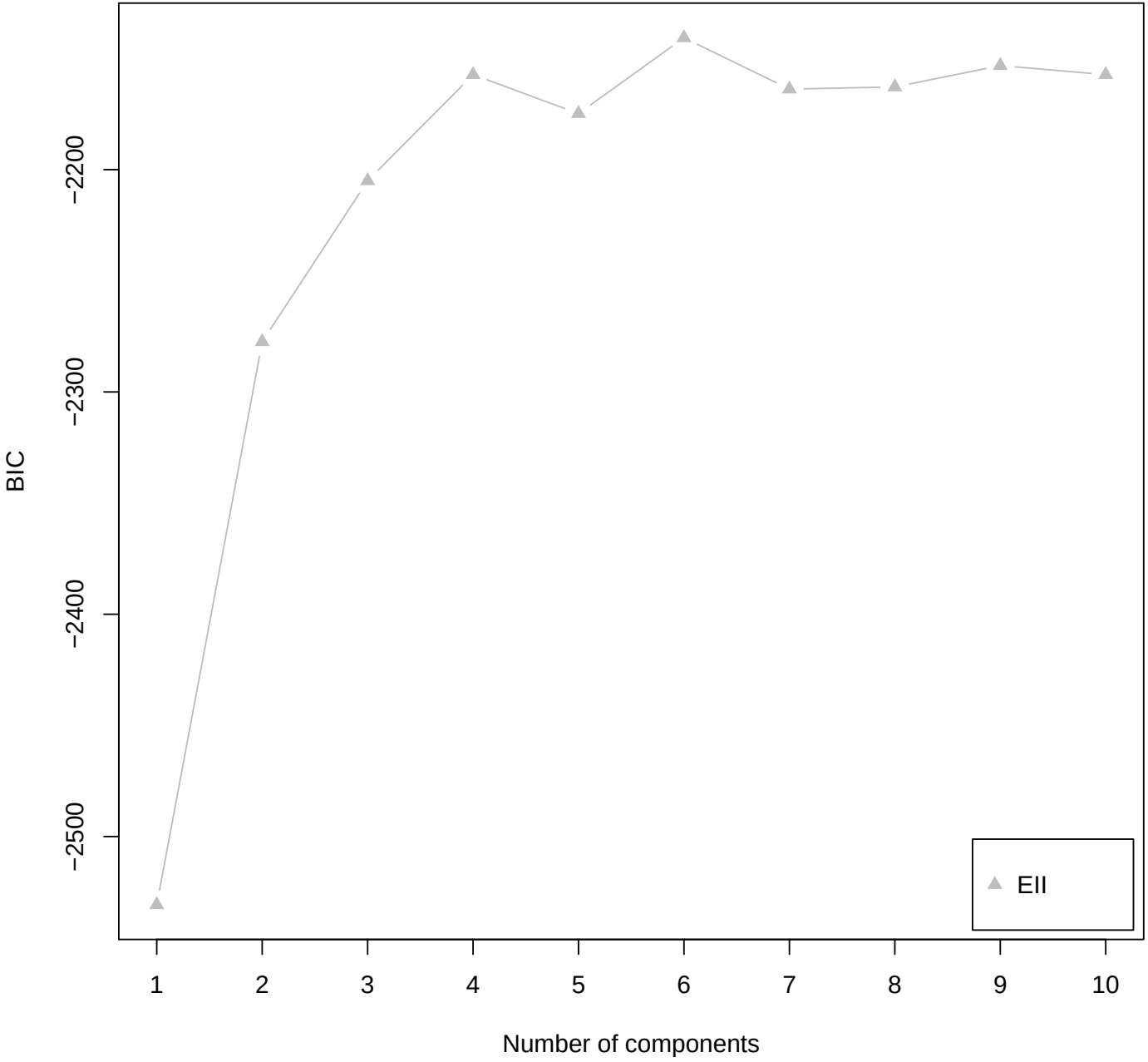



Figure: Analysis of the *swiss* data set with Mclust

# Part II

Bayesian linear regression
Gaussian processes
EM revisited
Model selection

# Outline Part 2

Bayesian linear regression

EM revisited

Gaussian processes

## Linear regression model

Using matrix notations, the linear regression model is given by:

$$Y = X\beta + \epsilon,$$

where $Y \in \mathbb{R}^n$ is a vector made out of the elements $y_i$, $X \in \mathcal{M}_{n\times p}(\mathbb{R})$ is a matrix where row $i$ is $x_i^\intercal$, and $\epsilon \in \mathbb{R}^n$ is a Gaussian random vector such that $\epsilon \sim \mathcal{N}(0_n, \sigma^2 I_n)$

- we now introduce a prior distribution over the regression vector $\beta$:

$$p(\beta) = \mathcal{N}(\beta; 0_p, \frac{I_p}{\alpha}),$$

  with $\alpha > 0$ fixed (for now)

## Reminders

The maximum likelihood estimator of the weight vector in the linear regression model is given by:

$$\hat{\beta} = (X^\intercal X)^{-1} X^\intercal Y$$

- cannot be computed if $X^\intercal X$ is not full rank
- if $p > n$, if $p >> n$: the so called high-dimensional setting

# Bayesian framework: first step

## Property

In the Bayesian framework, and considering the prior distribution $p(\beta)$ introduced before, looking for the maximum a posteriori estimate $\hat{\beta}_{\mathrm{MAP}}$ is equivalent to compute the ridge estimator:

$$\begin{aligned}\hat{\beta}_{\mathrm{MAP}} &= \mathrm{argmax}_{\beta} \log p(\beta|X, Y, \sigma^2) \\ &= \mathrm{argmin}_{\beta} \left\{ ||Y - X\beta||^2 + \lambda ||\beta||^2 \right\},\end{aligned}$$

with $\lambda = \alpha\sigma^2$

## Remark

In practice, in ridge regression, $\lambda$ is estimated using cross validation

## Property

In the Bayesian framework, and considering the prior distribution $p(\beta)$ introduced before, the maximum a posteriori estimate of $\beta$ is given by:

$$\hat{\beta}_{\mathrm{MAP}} = (X^\intercal X + \alpha\sigma^2 I_p)^{-1} X^\intercal Y$$

- provided that $\lambda = \alpha\sigma^2$ is large enough, $(X^\intercal X + \lambda I_p)$ is full rank and so $\hat{\beta}_{\mathrm{MAP}}$ can be computed
- simple solution for the high dimensional setting

# Bayesian framework: step 2

## Property

In the Bayesian framework, and considering the prior distribution $p(\beta)$ introduced before, the posterior distribution of the regression vector given the data has an analytical form:

$$p(\beta|X, Y, \sigma^2) = \mathcal{N}(\beta; m_n, S_n),$$

with

$$S_n = (\frac{X^\intercal X}{\sigma^2} + \alpha I_p)^{-1},$$

and

$$m_n = (X^\intercal X + \alpha\sigma^2 I_p)^{-1} X^\intercal Y$$

## Remark

Since $p(\beta|X, Y, \sigma^2)$ is Gaussian, its mode is its expectation:

$$\hat{\beta}_{\mathrm{MAP}} = m_n$$

# Outline Part 2

Bayesian linear regression

EM revisited

Gaussian processes

we now want to see $\alpha$ as an (hyper)parameter to be estimated from the training data set (link with ridge regression). So, $p(\beta)$ is replaced by:

$$p(\beta|\alpha) = \mathcal{N}(\beta; 0_p, \frac{I_p}{\alpha}),$$

with $\alpha > 0$ to be estimated.

# Bayesian framework: step 3: EM revisited

Seing $\beta$ as a latent (unknown) random vector, an EM algorithm can be derived to estimate the pair $(\alpha, \sigma^2)$ on the *full* training data set:

- init: initialise the values of $(\alpha, \sigma^2)$

E compute

- $S_n = (\frac{X^\intercal X}{\hat{\sigma^2}} + \hat{\alpha} I_p)^{-1}$
- $m_n = (X^\intercal X + \hat{\alpha}\hat{\sigma^2} I_p)^{-1} X^\intercal Y$

M compute

- $\hat{\alpha} = p/(\mathrm{Tr}(S_n) + m_n m_n^\intercal)$
- $\hat{\sigma^2} = (1/n)\left\{||Y - X m_n||^2 + \mathrm{Tr}(X^\intercal X S_n)\right\}$

- if the log-likelihood has changed (or the parameters) (no eps convergence) back to E.

## The evidence procedure

This algorithm is referred to as the evidence procedure (Mac92)

- the full training set is used to estimate $\alpha$ and $\sigma^2$ !
- no splits of the training data set are used as in cross validation !

# Outline Part 2

Bayesian linear regression

EM revisited

Gaussian processes

# Gaussian processes

As of now, we have:

- $Y|X, \beta, \sigma^2 \sim \mathcal{N}(X\beta, \sigma^2 I_n)$
- $\beta|\alpha \sim \mathcal{N}(0_p, \frac{I_p}{\alpha})$

## Reminder

The regression vector $\beta$ is seen as a latent (unknown) random vector. The hyperparameters are $\alpha$ and $\sigma^2$

# The Gaussian property

## Property

From the Gaussian property, we have:

$$Y|X, \sigma^2, \alpha \sim \mathcal{N}(O_n, \frac{XX^\intercal}{\alpha} + \sigma^2 I_n)$$

## Remark

- the associated likelihood $\mathcal{N}(Y; O_n, \frac{XX^\intercal}{\alpha} + \sigma^2 I_n)$ is sometimes referred to as the *type 2* maximum likelihood
- it can be optimised directly using optimisation algorithms
- warning: complexity: $O(n^3)$ !

# Gaussian processes

**Def**

More generally, Gaussian processes can be built directly as:

$$Y|X, \sigma^2, \theta \sim \mathcal{N}(0_n, C_n),$$

where $C_n = K_n + \sigma^2 I_n$ and

$$(K_n)_{ij} = k(x_i, x_j)$$

The function $k(\cdot, \cdot)$ is a kernel function.

# Example of a kernel function for Gaussian processes

**Def**

The exponential quadratic kernel is given by:

$$k(x_i, x_j) = \theta_0 \exp\left\{-\frac{\theta_1}{2}||x_i - x_j||^2\right\} + \theta_2 + \theta_3 x_i^\intercal x_j,$$

with

$$\beta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \in \mathbb{R}^4$$

# Optimisation

In Gaussian processes (GP), the optimisation problem is given by:

$$(\hat{\theta}, \hat{\sigma^2}) = \mathrm{argmax}_{\theta,\sigma^2} \log \mathcal{N}(Y; 0_n, C_n)$$

## Remarks

- again, the complexity is $O(n^3)$
- the parameters $\theta$ and $\sigma^2$ "only" play a role in the covariance matrix of the model

# Predictions in GP

- $(X, Y)$ is the training data set with $n$ elements
- let us consider a new observation $x_{n+1}$ for which we aim at predicting $y_{n+1}$
- we build

$$X_{n+1} = \begin{pmatrix} X \\ x_{n+1}^{\intercal} \end{pmatrix},$$

and

$$Y_{n+1} = \begin{pmatrix} Y \\ y_{n+1} \end{pmatrix},$$

- the model becomes:

$$Y_{n+1}|X_{n+1}, \theta, \sigma^2 \sim \mathcal{N}(0_{n+1}, C_{n+1})$$

with

$$C_{n+1} = \begin{pmatrix} C_n & k \\ k^\intercal & c \end{pmatrix},$$

and $k_i = k(x_i, x_{n+1}) = k(x_{n+1}, x_i), \forall i \in \{1, \ldots, n\}$, and $c = k(x_{n+1}, x_{n+1}) + \sigma^2$

## Property

From Gaussian property, it follows that:

$$y_{n+1}|X_{n+1}, Y, \theta, \sigma^2 \sim \mathcal{N}(\tilde{m}, \tilde{\sigma^2})$$

# References I


- A.P. Dempster, N.M. Laird, and D.B. Rubin, *Maximum likelihood for incomplete data via the em algorithm*, Journal of the Royal Statistical Society **B39** (1977), 1–38.
- D. MacKay, *A practical bayesian framework for backpropagation networks*, Neural Computation **4** (1992), 448–472.