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

$$\mathbb{E}_{Z_i|x_i, \pi, \theta}\left[Z_i\right] = \tau_i$$

Model parameters: $\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_K \end{pmatrix}$, $\theta = (\mu_k, \Sigma_k)_k$

Property

τ_{ik} is given by :

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

Inference: $(\tau_i)_i$

- τ_{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 :

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

$$E_{Z_i | x_i, \pi, \theta}\left[Z_{ik}\right] = \tau_{ik}$$

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 π and θ 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

if kmeans find i in cluster k

$\Rightarrow \tau_{ik} = 1, \tau_{il} = 0, \forall l \neq k$

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

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

E compute the τ_i with respect to π and θ

- 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

Rk:

EM for fixed K $\to$ $\hat{\pi}$

$\to$ $\hat{\theta} = (\hat{\mu}_k, \hat{\Sigma}_k)_k$

$\to$ $(\hat{\tau}_{ik})_{ik}$

$$\boxed{\tau_i = \begin{pmatrix} 0,1 \\ 0 \\ 0,9 \end{pmatrix}} \updownarrow K=3$$

soft clustering

$$\tau_i = \begin{pmatrix} 0 \\ 0,45 \\ 0,35 \end{pmatrix}$$

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

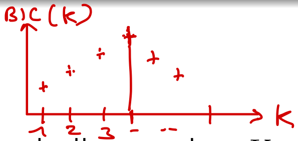
Clustering

Gaussian mixture models

EM algorithm

Model selection

Model selection



R mclust

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 : $\mathrm{AIC}(K) = L_{(x_i)_i}(\hat{\pi}, \hat{\theta}) - M_K$

$$M_K = K - 1 + Kd + K\frac{d(d+1)}{2}$$

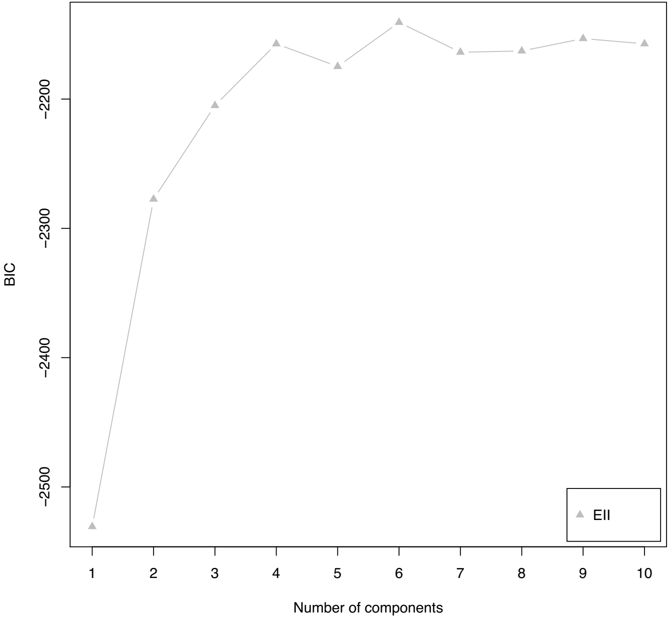


Figure: Analysis of the *swiss* data set with Mclust

Part II

Bayesian linear regression
Gaussian processes
EM revisited
Model selection

Outline Part 2

Bayesian linear regression

EM revisited

Gaussian processes

Bayesian linear regression

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad y_i \in \mathbb{R}, \ \forall i$$

Linear regression model

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

$$Y = X\beta + \epsilon, \quad \Rightarrow \quad Y | X, \beta, \sigma^2 \sim \mathcal{N}(X\beta, \sigma^2 I_n)$$

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 β:

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

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

Reminders

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

$\hat{\beta}_{MLE}$ vs $\hat{\beta}_{MAP}$

Property

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

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

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

Remark

In practice, in ridge regression, λ 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 β is given by:

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

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