EM Algorithme

Léos Coutrot

2023-12-17

Contents

Algorithme EM

Nous allons implémenter l'algorithme EM dans le cadre d'un mélange de deux gaussiennes (voir Exercice 1 du TD sur les mixtures pour avoir le détail des paramètres).

Initialisation

Pour cette étape, nous allons choisir aléatoirement des paramètres

```
#Initialisation
init <- function(X){
  pi <- runif(1,min = 0, max = 1)
  mu1 <- sample(X, 1) # Choisir aléatoirement un point pour mu1
  mu2 <- sample(X, 1) # Choisir aléatoirement un autre point pour mu2
  sigma1 <- 1
  sigma2 <- 1
  return(list(pi = pi, mu1 = mu1, mu2 = mu2, sigma1 = sigma1, sigma2 = sigma2))
}
```

E-step

Dans l'étape E de l'algorithme EM, pour chaque itération q, on calcul l'espérance de la log vraisemblance

$$\mathbb{E}_{Z|X;\theta^q}[\log P(X, Z; \theta)], \qquad (\theta = (\pi, \mu, \sigma))$$

On pose la fonction Q tel que:

$$Q(\theta, \theta^q) = \mathbb{E}_{Z|X;\theta^q}[\log P(X, Z; \theta)]$$

On a alors

$$\begin{aligned}\log \mathcal{P}(X, Z; \theta) &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}[1_{\{Z_i=k\}}] \log(P(x_i, Z_i = k)) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}[1_{\{Z_i=k\}}] \log(P(Z_i = k; \theta) P(x_i | Z_i = k; \theta)) \\ \log \mathcal{P}(X, Z; \theta) &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}[1_{\{Z_i=k\}}] \log(\pi_k) f_k(x; \theta)\end{aligned}$$

Et on pose pour chaque itération q et $\forall (i,k) \in [1,n] \times [1,K]$,

$$\begin{aligned}t_{ik}^{(q)} &= \mathbb{E}[1_{\{Z_i=k\}}] \\ &= P(Z_i = k | x_i; \theta^q) \\ t_{ik}^{(q)} &= \frac{\pi_k^{(q)} f(x_i, \theta_k^{(q)})}{\sum_{l=1}^{K} \pi_l^{(q)} f(x_i, \theta_l^{(q)})}\end{aligned}$$

On calcule tout les t_{ik} lors de cette étape. Voici une implémentation fonctionnelle :

```
# Etape E
E_step <- function(X, pi, mu1, mu2, sigma1, sigma2) {
  a <- pi * dnorm(X, mean = mu1, sd = sigma1)
  b <- (1 - pi) * dnorm(X, mean = mu2, sd = sigma2)
  a <- a / (a + b)
  b <- b / (a + b)
  return(list(a = a, b = b))
}
```

M-step

L'objectif de l'étape M est de trouver les paramètres qui vont maximiser la fonction Q de l'étape E. Pour ce faire, on regarde la dérivée de Q en 0.

Après un bref calcul on trouve :

$$\pi_k = \frac{\sum_{i=0}^{n} t_{ik}}{n}$$

$$\mu_k = \frac{\sum_{i=0}^{n} t_{ik} x_i}{\sum_{i=0}^{n} t_{ik}}$$

$$\sigma_k^2 = \frac{\sum_{i=0}^{n} t_{ik} (x_i - \mu_k)^2}{\sum_{i=0}^{n} t_{ik}}$$

```
# Etape M
M_step <- function(X, a, b) {
  pi <- mean(a)
  mu1 <- sum(a * X) / sum(a)
  mu2 <- sum(b * X) / sum(b)
  sigma1 <- sqrt(sum(a * (X - mu1)^2) / sum(a))
  sigma2 <- sqrt(sum(b * (X - mu2)^2) / sum(b))
return(list(pi = pi, mu1 = mu1, mu2 = mu2, sigma1 = sigma1, sigma2 = sigma2))
}
```

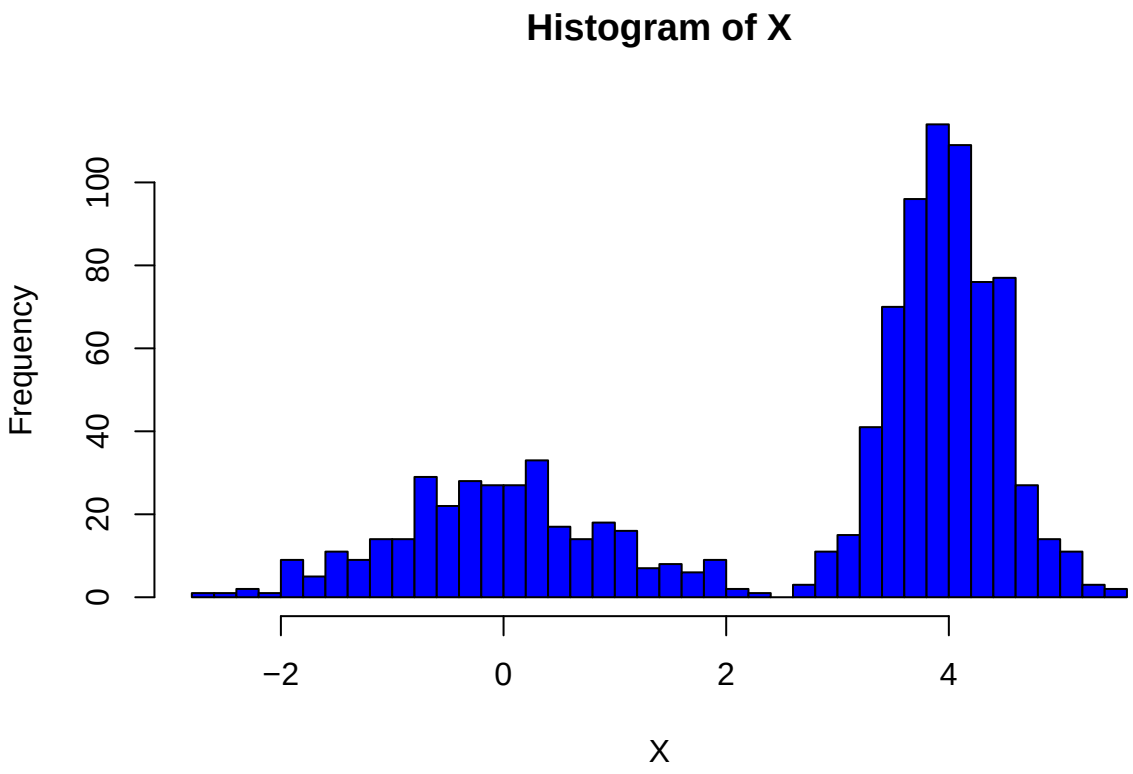
Algorithme EM

Nous allons donc implémenter notre algorithme EM.

```
# L'algorithme EM final
em_algorithme <- function(X, max_iterations = 1000, tolerance = 1e-10) {
  # Initialisation des paramètres
  params <- init(X)
  for (iteration in 1:max_iterations) {
    params_old <- params
    # E-étape
    t_values <- E_step(X, params$pi, params$mu1, params$mu2, params$sigma1, params$sigma2)
    # M-étape
    params <- M_step(X, t_values$a, t_values$b)
    # Vérifier la convergence
    if (abs(params$mu1 - params_old$mu1) < tolerance && abs(params$mu2 - params_old$mu2) < tolerance &&
    break
  }
  }
return(params)
}
```

Application

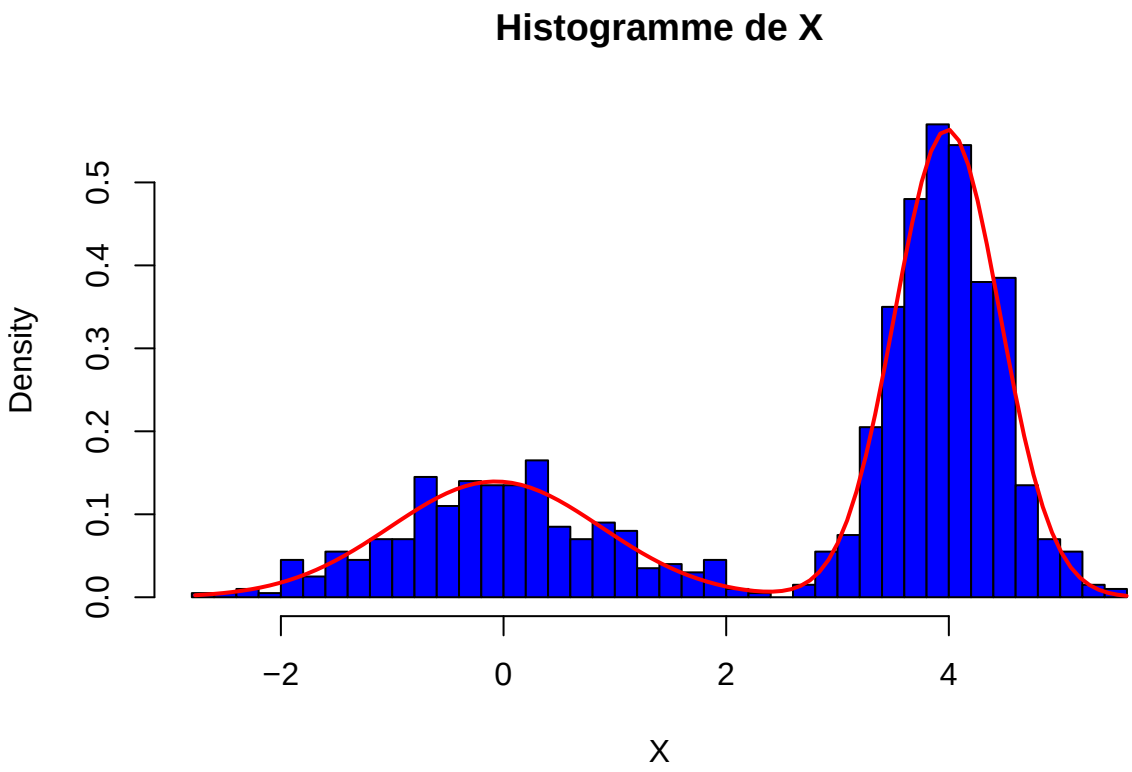
```
n <- 1000
pi_k=c(1/3,2/3)
Z=rep(NA,n)
X=rep(NA,n)
Z = sample(x=c(1,2),size=n,prob=pi_k,replace = TRUE)
X=rep(NA,n)
X[Z==1]=rnorm(sum(Z==1),0,1)
X[Z==2]=rnorm(sum(Z==2),4,1/2)
hist(X,breaks = 50,col = "blue")
```




```
res<-em_algorithme(X)
res
```

```
## $pi
## [1] 0.6682918
##
## $mu1
## [1] 3.986655
##
## $mu2
## [1] -0.07187505
##
## $sigma1
## [1] 0.4727907
##
## $sigma2
## [1] 0.9484916
```

```
mixture_density <- function(x) {
  res$pi * dnorm(x, mean = res$mu1, sd = res$sigma1) + (1-res$pi) * dnorm(x, mean = res$mu2, sd = res$si
}
hist(X, breaks = 50, freq = FALSE, col = "blue", xlab = "X", main = "Histogramme de X")
curve(mixture_density, from = min(X), to = max(X), col = "red", add = TRUE, lwd = 2)
```



On retrouve bien nos différents paramètres.