University of Edinburgh, School of Mathematics
Incomplete Data Analysis, 2020/2021
EM algorithm for the two-component Gaussian mixture

Vanda Inácio

Here we implement the EM algorithm for the two-component Gaussian mixture model. Remember that for the E-step we have

$$\begin{aligned} E[Z_i \mid y, \theta^{(t)}] &= E[Z_i \mid y_i, \theta^{(t)}] \\ &= 1 \times \Pr(Z_i = 1 \mid y_i, \theta^{(t)}) + 0 \times \Pr(Z_i = 0 \mid y_i, \theta^{(t)}) \\ &= \frac{p^{(t)} \phi\left(y_i; \mu_1^{(t)}, (\sigma_1^{(t)})^2\right)}{p^{(t)} \phi\left(y_i; \mu_1^{(t)}, (\sigma_1^{(t)})^2\right) + (1 - p^{(t)}) \phi\left(y_i; \mu_2^{(t)}, (\sigma_2^{(t)})^2\right)} \\ &= \widetilde{p}_i^{(t)}, \end{aligned}$$

and thus

$$Q(\theta \mid \theta^{(t)}) = \sum_{i=1}^{n} \widetilde{p}_i^{(t)} \left\{\log p + \log \phi(y_i; \mu_1, \sigma_1^2)\right\} + \sum_{i=1}^{n} \left(1 - \widetilde{p}_i^{(t)}\right) \left\{\log(1-p) + \log \phi(y_i; \mu_2, \sigma_2^2)\right\}.$$

For the M-step we have

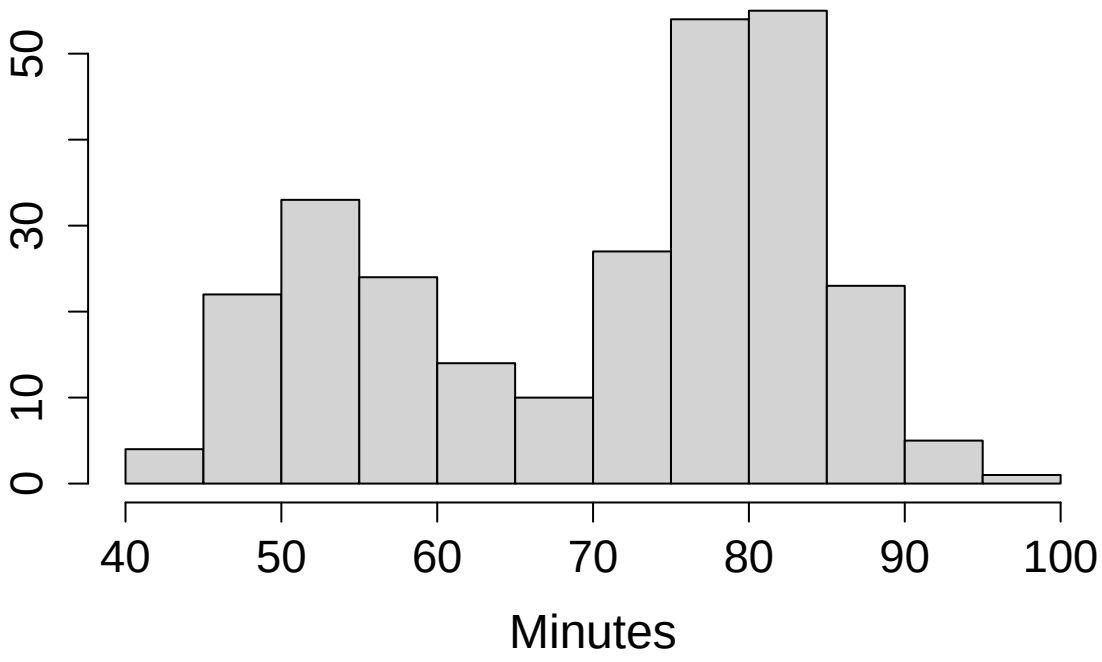
$$\begin{aligned} p^{(t+1)} &= \frac{\sum_{i=1}^{n} \widetilde{p}_i^{(t)}}{n}, \\ \mu_1^{(t+1)} &= \frac{\sum_{i=1}^{n} \widetilde{p}_i^{(t)} y_i}{\sum_{i=1}^{n} \widetilde{p}_i^{(t)}}, \\ (\sigma_1^{(t+1)})^2 &= \frac{\sum_{i=1}^{n} \widetilde{p}_i^{(t)} (y_i - \mu_1^{(t+1)})^2}{\sum_{i=1}^{n} \widetilde{p}_i^{(t)}}, \\ \mu_2^{(t+1)} &= \frac{\sum_{i=1}^{n} (1 - \widetilde{p}_i^{(t)}) y_i}{\sum_{i=1}^{n} (1 - \widetilde{p}_i^{(t)})}, \\ (\sigma_2^{(t+1)})^2 &= \frac{\sum_{i=1}^{n} (1 - \widetilde{p}_i^{(t)}) (y_i - \mu_2^{(t+1)})^2}{\sum_{i=1}^{n} (1 - \widetilde{p}_i^{(t)})}. \end{aligned}$$

```
em.mixture.two.normal <- function(y, theta0, eps){
n <- length(y)

theta <- theta0

p <- theta[1]
mu1 <- theta[2]; sigma1 <- theta[3]
```

```
mu2 <- theta[4]; sigma2 <- theta[5]

diff <- 1
while(diff > eps){

theta.old <- theta

#E-step
ptilde1 <- p*dnorm(y, mean = mu1, sd = sigma1)
ptilde2 <- (1 - p)*dnorm(y, mean = mu2, sd = sigma2)
ptilde <- ptilde1/(ptilde1 + ptilde2)

#M-step
p <- mean(ptilde)

mu1 <- sum(y*ptilde)/sum(ptilde)
sigma1 <- sqrt(sum(((y - mu1)^2)*ptilde)/sum(ptilde))

mu2 <- sum(y*(1 - ptilde))/sum(1 - ptilde)
sigma2 <- sqrt(sum(((y - mu2)^2)*(1 - ptilde))/sum(1 - ptilde))

theta <- c(p, mu1, sigma1, mu2, sigma2)
diff <- sum(abs(theta - theta.old))
}
return(theta)
}
```

We will now load the old faithful data set.

```
data(faithful)
attach(faithful)
hist(waiting, main = "Time between Old Faithful eruptions",
     xlab = "Minutes",
     ylab="", cex.main = 1.5, cex.lab = 1.5, cex.axis = 1.4)
```



Lastly, we apply the developed function to the old faithful data. For the starting values we have considered

$$p^{(0)} = 0.4, \quad \mu_1^{(0)} = 40, \quad \sigma_1^{(0)} = 4, \quad \mu_2^{(0)} = 90, \quad \sigma_2^{(0)} = 4,$$

while for the stopping criterion we have used

$$|p^{(t+1)} - p^{(t)}| + |\mu_1^{(t+1)} - \mu_1^{(t)}| + |\sigma_1^{(t+1)} - \sigma_1^{(t)}| + |\mu_2^{(t+1)} - \mu_2^{(t)}| + |\sigma_2^{(t+1)} - \sigma_2^{(t)}| < \varepsilon,$$

with $\varepsilon = 0.00001$

```
res <- em.mixture.two.normal(y = waiting, c(0.4, 45, 4, 90, 4), 0.00001)
p <- res[1]
mu1 <- res[2]
sigma1 <- res[3]
mu2 <- res[4]
sigma2 <- res[5]

p; mu1; mu2; sigma1; sigma2
```

```
## [1] 0.3608862
```

```
## [1] 54.61486
```

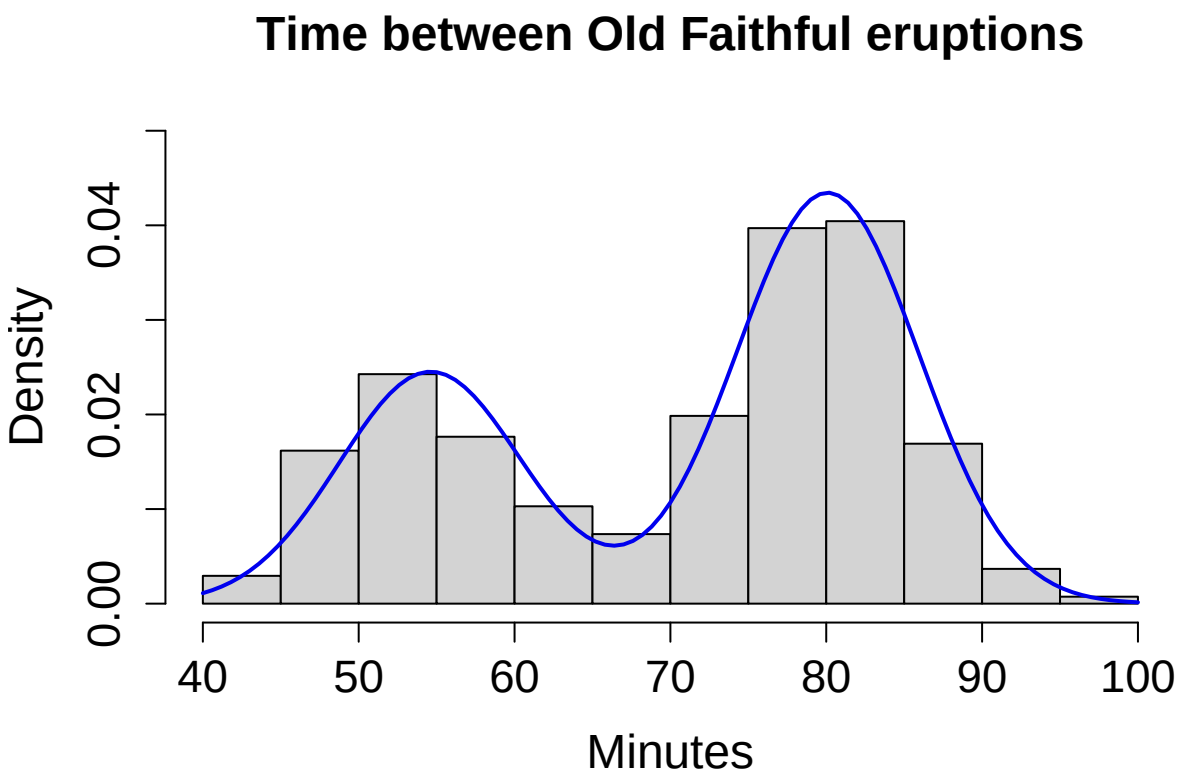
```
## [1] 80.09107
```

```
## [1] 5.871224
```

```
## [1] 5.867731
```

```
hist(waiting, main = "Time between Old Faithful eruptions",
     xlab = "Minutes",
     ylab = "Density",
     cex.main = 1.5, cex.lab = 1.5, cex.axis = 1.4,
```

```
     freq = F, ylim = c(0,0.05))
curve(p*dnorm(x, mu1, sigma1) + (1 - p)*dnorm(x, mu2, sigma2),
      add = TRUE, lwd = 2, col = "blue2")
```

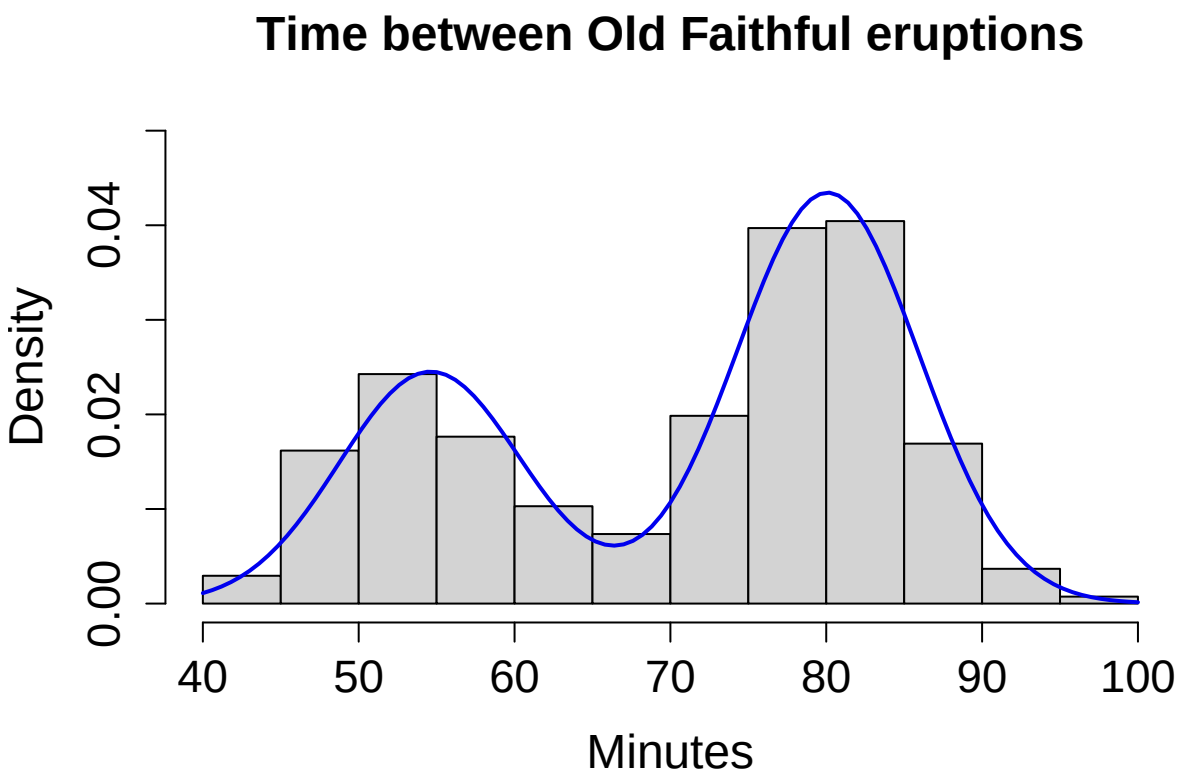


The package `mixtools` is capable of handling mixtures of Gaussians through the function `normalmixEM`. The code follows below.

```
require(mixtools)
res1 <- normalmixEM(waiting, lambda = 0.4, mu = c(45, 90), sigma = c(4, 4))
```

```
## number of iterations= 16
```

```
p <- res1$lambda[1]
mu1 <- res1$mu[1]
mu2 <- res1$mu[2]
sigma1 <- res1$sigma[1]
sigma2 <- res1$sigma[2]

hist(waiting, main = "Time between Old Faithful eruptions",
     xlab = "Minutes",
     ylab = "Density",
     cex.main = 1.5, cex.lab = 1.5, cex.axis = 1.4,
     freq = F, ylim = c(0,0.05))
curve(p*dnorm(x, mu1, sigma1)+(1 - p)*dnorm(x, mu2, sigma2),
      add = TRUE, lwd = 2, col = "blue2")
```


Time between Old Faithful eruptions
Density
0.00
0.02
0.04
40
50
60
70
80
90
100
Minutes