Homework 5

Louis Bensard

March 17, 2018

Problem J-4.1:

(a)

```
EM_mult <- function(x, theta, maxit, tolerr){

  beta = 1
  theta_star = (-1657 + sqrt(3728689))/7680

    cat("\nIteration number n", "    theta_n", "        MRE", "        CV ratio\n")
    cat(sprintf('%2.0f                     %6.6f       %1.1e          %1.1e\n',
                    0, theta, NA, NA))

    for (it in 1: maxit){

        y2_star = x[1]*theta/(2 + theta)
        theta1 = (y2_star + x[4])/(y2_star + x[2] + x[3] + x[4])

        mre = abs(theta1-theta)/max(1, abs(theta1))
        cv_ratio = abs(theta1 - theta_star)/(abs(theta - theta_star)^beta)

        cat(sprintf('%2.0f                     %6.6f       %1.1e     %1.1e\n',
                 it, theta1, mre, cv_ratio))

        if(mre < tolerr) break

        theta = theta1
    }
}

x = c(1997, 907, 904, 32)
theta_0 = 0.02
maxit = 200 ; tolerr = 1e-6

EM_mult(x, theta_0, maxit, tolerr)
```

```
##
## Iteration number n     theta_n        MRE        CV ratio
##  0                    0.020000       NA          NA
##  1                    0.027793       7.8e-03     5.0e-01
##  2                    0.031743       3.9e-03     5.0e-01
##  3                    0.033721       2.0e-03     5.0e-01
##  4                    0.034706       9.8e-04     5.0e-01
##  5                    0.035195       4.9e-04     5.0e-01
##  6                    0.035437       2.4e-04     5.0e-01
##  7                    0.035557       1.2e-04     5.0e-01
```

```
## 8                          0.035616       5.9e-05      4.9e-01
## 9                          0.035646       2.9e-05      4.9e-01
## 10                         0.035660       1.5e-05      4.9e-01
## 11                         0.035668       7.2e-06      4.9e-01
## 12                         0.035671       3.6e-06      4.9e-01
## 13                         0.035673       1.8e-06      4.9e-01
## 14                         0.035674       8.7e-07      4.9e-01
```

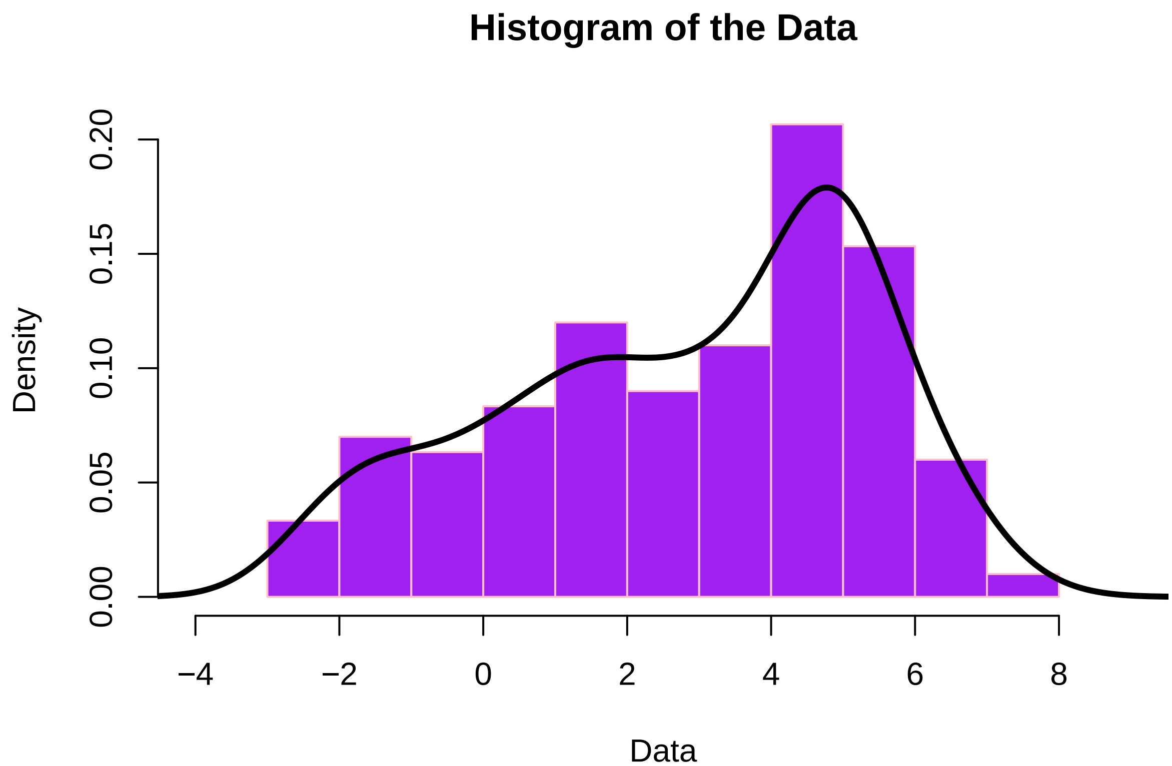
(b)

With trials and error, the only value of beta that "stabilizes" the convergence ratio is $\beta = 1$. Indeed, when $\beta = 1$, the convergence ratio is $0.49 < 1$. Therefore, the EM is linerarly convergent for this problem.

Problem J-4.2:

(a)

```
data = read.table('http://mathfaculty.fullerton.edu/mori/Math534/Homework/Chapter%204/ExJ42.txt',
                  header=TRUE)
y = data[,1]

hist(y, xlab="Data", ylab='Density', main="Histogram of the Data", freq=FALSE,
     xlim=c(-4,9), col='purple', border='pink')

ksmooth <- density(data[,1], na.rm=T, from=-5, to=10)
lines(ksmooth, col="black", lwd=3)
```



(b)

Define $f(\boldsymbol{y}|\boldsymbol{\theta}) = \alpha f_1(\boldsymbol{y}|\boldsymbol{\mu}, \sigma^2) + \beta f_2(\boldsymbol{y}|\boldsymbol{\mu}, \sigma^2) + (1-\alpha-\beta) f_3(\boldsymbol{y}|\boldsymbol{\mu}, \sigma^2)$. We wish to estimate the parameter $\boldsymbol{\theta} = (\alpha, \beta, \mu_1, \mu_2, \mu_3, \sigma^2)^T$. For simplicity, let $\pi_1 = \alpha$, $\pi_2 = \beta$, $\pi_3 = 1-\alpha-\beta$.

Consider the complete data of the form: $(y_1, \boldsymbol{z_1})..., (y_n, \boldsymbol{z_n})$ where $\boldsymbol{z_i} = (z_{i1}, z_{i2}, z_{i3})$ with:

$$z_{ik} = \begin{cases} 1, & \text{if the } i^{th} \text{ case belongs to the } j^{th} \text{ group} \\ 0, & \text{otherwise} \end{cases}$$

It is now clear that the maximizing value $\widetilde{\pi_k}$ of π_k is:

$$\widetilde{\pi_k} = \sum_{i=1}^{n} \frac{z_{ik}}{n}$$

To write the likelihood of the complete data $l_c(\boldsymbol{\theta})$, we need the joint density of $(y_i, \boldsymbol{z_i})$: $f(y_i, \boldsymbol{z_i}) = f(y_i|\boldsymbol{z_i}) f(\boldsymbol{z_i})$. Suppose $z_{ik} = 1$, $(k = 1, 2, 3)$, then:

$$\begin{aligned} & f(y_i|z_{ik} = 1, \boldsymbol{\mu}, \sigma^2) = f_k(y_i|\mu_k, \sigma^2) \\ & f(z_{ik} = 1) = P(z_{ik} = 1) = \pi_k \\ \Rightarrow & f(y_i, z_{ik} = 1|\boldsymbol{\mu}, \sigma^2) = f_k(y_i|\mu_k, \sigma^2) \cdot \pi_k \end{aligned}$$

Thus, the contribution of $(y_i, z_{i1}, z_{i2}, z_{i3})$ to the complete data log-likelihood is:$[f_k(y_i|\mu_k, \sigma^2) \cdot \pi_k]^{z_{ik}}$. The full expression of the log-likelihood is then:

$$l_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{3} z_{ik} \cdot [log(f_k(y_i|\mu_k, \sigma^2)) + log(\pi_k)]$$

Define $Q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = E[l_c(\boldsymbol{\theta}')|\boldsymbol{y}, \boldsymbol{\theta}] = E^*[l_c(\boldsymbol{\theta}')]$. Our goal at each iteration is going to find the value $\boldsymbol{\theta}'$ that maximizes Q. After expanding all the terms, we obtain:

$$Q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{3} E^*[z_{ik}] \cdot [\frac{-1}{2} log(2\pi) - \frac{1}{2} log((\sigma^2)') - \frac{1}{2(\sigma^2)'}(y_i - \mu_k')^2 + log(\pi_k')]$$

If we take the partial derivative of Q with respect to each μ_k and σ^2, and set it equal to zero, we get the following maximizing values of the parameters:

$$\widetilde{\mu_k} = \frac{\sum_{i=1}^{n} z_{ik}^* y_i}{\sum_{i=1}^{n} z_{ik}^*} = \frac{\sum_{i=1}^{n} z_{ik}^* y_i}{n\widetilde{\pi_k}}$$
$$\widetilde{\sigma^2} = \frac{1}{n} \sum_{i=1}^{n} z_{i1}^*(y_i - \widetilde{\mu_1})^2 + z_{i2}^*(y_i - \widetilde{\mu_2})^2 + z_{i3}^*(y_i - \widetilde{\mu_3})^2$$

Therefore, at each iteration, $Q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ is maximized at $\boldsymbol{\theta}' = \widetilde{\boldsymbol{\theta}} = (\widetilde{\pi_1}, \widetilde{\pi_2}, \widetilde{\mu_1}, \widetilde{\mu_2}, \widetilde{\mu_3}, \widetilde{\sigma^2})$.

Here is the EM-Algorithm that we will use in part (c):

- Start with an initial $\boldsymbol{\theta_0}$
- E-step: Compute $z_{ik}^* = \frac{f_k(y_i)\pi_k}{\sum_{j=1}^{3} f_j(y_i)\pi_j}$
- M-step: Compute $\widetilde{\boldsymbol{\theta}}$ using z_{ik}^*
- replace $\boldsymbol{\theta}$ by $\widetilde{\boldsymbol{\theta}}$ and go to E-step

(c)

```
theta_0 = c(1/3,1/3,-1,2,6,1)
maxit = 200 ; tolerr = 1e-6

theta_ME = EM_mixture(y, theta_0, maxit, tolerr)
```

```
##
## It n    alpha_n    beta_n    mu1_n    mu2_n    mu3_n   sigma^2_n      MRE
##  0     0.333333   0.333333 -1.000000 2.000000 6.000000  1.000000      NA
##  1     0.212329   0.369427 -0.888351 2.334978 5.173024  0.999559     2.7e-01
##  2     0.216325   0.333440 -0.870753 2.246392 5.057131  1.012352     6.2e-02
## ...
## ...
## ...
## 59     0.180811   0.295449 -1.099083 1.680811 4.849163  1.005054     2.1e-06
## 60     0.180810   0.295449 -1.099085 1.680809 4.849162  1.005054     1.7e-06
```

```
## 61     0.180810   0.295449 -1.099086 1.680808 4.849162  1.005054     1.4e-06
## 62     0.180810   0.295449 -1.099087 1.680807 4.849161  1.005054     1.2e-06
## 63     0.180810   0.295449 -1.099087 1.680806 4.849161  1.005054     9.6e-07
## ----------------------------------------------------------
```

```r
cat("\ntheta_final = ", theta_ME)
```

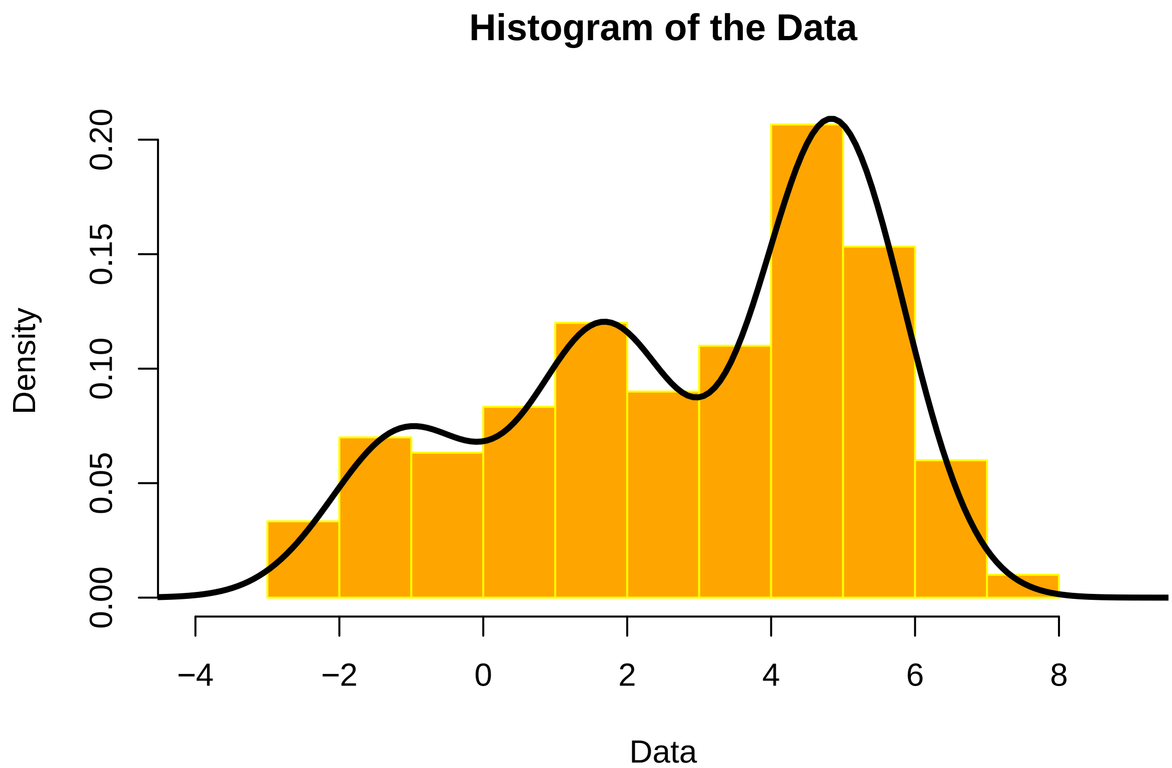
```
##
## theta_final =  0.1808101 0.2954494 -1.099087 1.680807 4.849161 1.005054
```

Thus, after 63 iterations, the converged solutions are: $\widetilde{\pi_1} = 0.1808$, $\widetilde{\pi_2} = 0.2954$, $\widetilde{\mu_1} = -1.0991$, $\widetilde{\mu_2} = 1.6808$, $\widetilde{\mu_3} = 4.8492$ and $\widetilde{\sigma^2} = 1.0051$.

We noted that using further off starting values such as $\theta_0 = (0.1, 0.7, 1, 4, 7, 0.7)^T$ or $\theta_0 = (0.1, 0.7, 1, -12, -14, 0.2)^T$, we obtain two different converged solutions. Those solutions are due to the fact that the algorithm gets "trapped" into a local maximum that is not the global maximum. To avoid it, we should compare the log-likehoods and keep the solution with the highest log-likelihood (the solution displayed above).

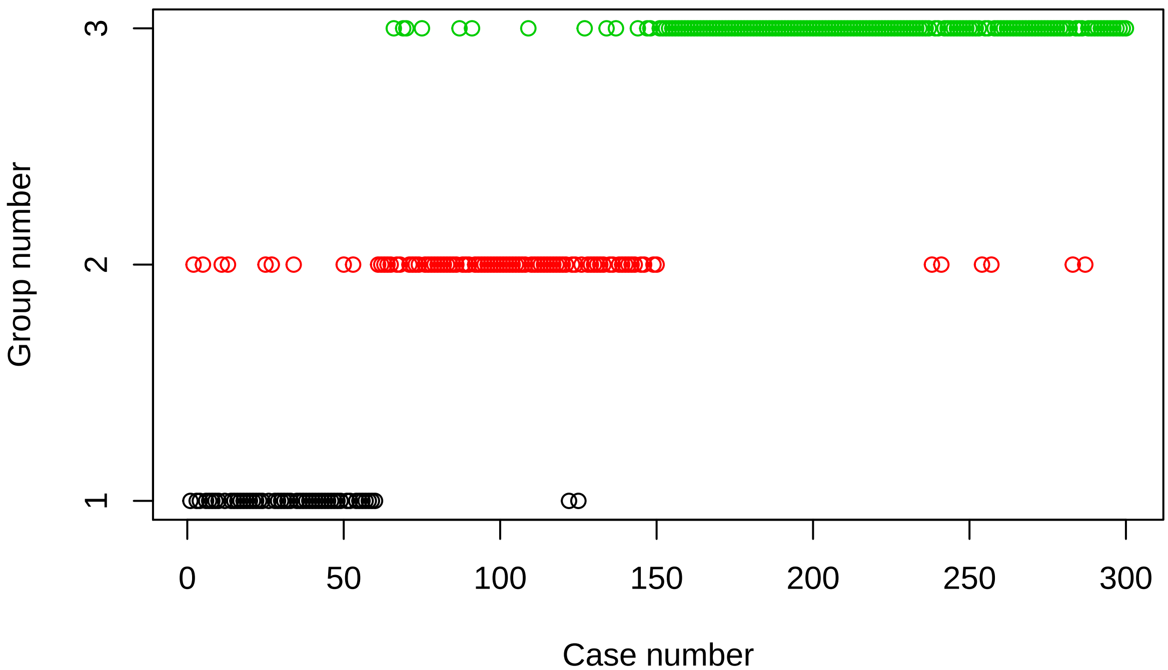
(d)

```r
x = seq(-5,10, length=200)

f1_ME = dnorm(x, mean = theta_ME[3], sd = sqrt(theta_ME[6]))
f2_ME = dnorm(x, mean = theta_ME[4], sd = sqrt(theta_ME[6]))
f3_ME = dnorm(x, mean = theta_ME[5], sd = sqrt(theta_ME[6]))
f_ME = theta_ME[1]*f1_ME +  theta_ME[2]*f2_ME + (1-theta_ME[1] - theta_ME[2])*f3_ME


hist(y, xlab="Data", ylab='Density', main="Histogram of the Data", freq=FALSE,
     xlim=c(-4,9), col='orange',border='yellow')
lines(x, f_ME, col='black', lwd=3)
```



(e)

```
n = length(y)
h_prob = rep(0,n)

for(i in 1:n){

    y_i = y[i]
    z_star_i1 = z_star_ik(y_i, 1, theta_ME)
    z_star_i2 = z_star_ik(y_i, 2, theta_ME)
    z_star_i3 = z_star_ik(y_i, 3, theta_ME)

    h_prob[i] = which.max(c(z_star_i1, z_star_i2, z_star_i3))

}

plot(h_prob, col=h_prob, xlab="Case number", ylab="Group number", yaxt="n")
axis(2, at = 1:3)
```



We can clearly see a pattern. The data is partitionned into three groups, with few cases that are missorted. This indicates that the data can be broken down into three distinct groups, where each group follows a normal distribution. The cases corresponding to group 1 (black, except for the ones missorted) are normally distributed with mean $\mu_1 = -1.0991$ and variance $\sigma^2 = 1.0051$. The cases corresponding to group 2 (red, except for the ones missorted) are normally distributed with mean $\mu_2 = 1.6808$ and variance $\sigma^2 = 1.0051$. The cases corresponding to group 3 (green, except for the ones missorted) are normally distributed with mean $\mu_3 = 4.8492$ and variance $\sigma^2 = 1.0051$.