Homework_5_Part_I

Michael Piccolo

3/31/2021

Exercise J-4.1

Part (a)

```
x = c(1997,907,904,32)

EM <- function(theta, x, maxit, tolerr){
  theta_star = (-1657 + sqrt(3728689))/7680
  for (it in 1:maxit) {
    E = x[1]*(theta/(2+theta))
    theta_tilde <- (E + x[4])/(E + x[2] + x[3] + x[4])
    mod_rel_err <- max(abs((theta_tilde - theta) / max(1,theta_tilde)))
    convergence_ratio = abs(theta_tilde-theta_star)/abs(theta-theta_star)
    print(sprintf('it = %3.0f   theta = %12.12f     MRE=%2.1e     Convergence Ratio = %3.3e',
                  it, theta_tilde , mod_rel_err, convergence_ratio), quote = FALSE)
    if(mod_rel_err < tolerr) {
      break
    }
    theta = theta_tilde
  }
}


a <- EM(0.02, x, 200, 1e-6)
```

```
## [1] it =   1   theta = 0.027793132773     MRE=7.8e-03     Convergence Ratio = 5.028e-01
## [1] it =   2   theta = 0.031742941132     MRE=3.9e-03     Convergence Ratio = 4.989e-01
## [1] it =   3   theta = 0.033721123764     MRE=2.0e-03     Convergence Ratio = 4.969e-01
## [1] it =   4   theta = 0.034705946241     MRE=9.8e-04     Convergence Ratio = 4.959e-01
## [1] it =   5   theta = 0.035194771667     MRE=4.9e-04     Convergence Ratio = 4.954e-01
## [1] it =   6   theta = 0.035437045211     MRE=2.4e-04     Convergence Ratio = 4.951e-01
## [1] it =   7   theta = 0.035557033560     MRE=1.2e-04     Convergence Ratio = 4.950e-01
## [1] it =   8   theta = 0.035616437341     MRE=5.9e-05     Convergence Ratio = 4.950e-01
## [1] it =   9   theta = 0.035645841640     MRE=2.9e-05     Convergence Ratio = 4.949e-01
## [1] it =  10   theta = 0.035660395186     MRE=1.5e-05     Convergence Ratio = 4.949e-01
## [1] it =  11   theta = 0.035667598091     MRE=7.2e-06     Convergence Ratio = 4.949e-01
## [1] it =  12   theta = 0.035671162906     MRE=3.6e-06     Convergence Ratio = 4.949e-01
## [1] it =  13   theta = 0.035672927162     MRE=1.8e-06     Convergence Ratio = 4.949e-01
## [1] it =  14   theta = 0.035673800302     MRE=8.7e-07     Convergence Ratio = 4.949e-01
```

Part (b)

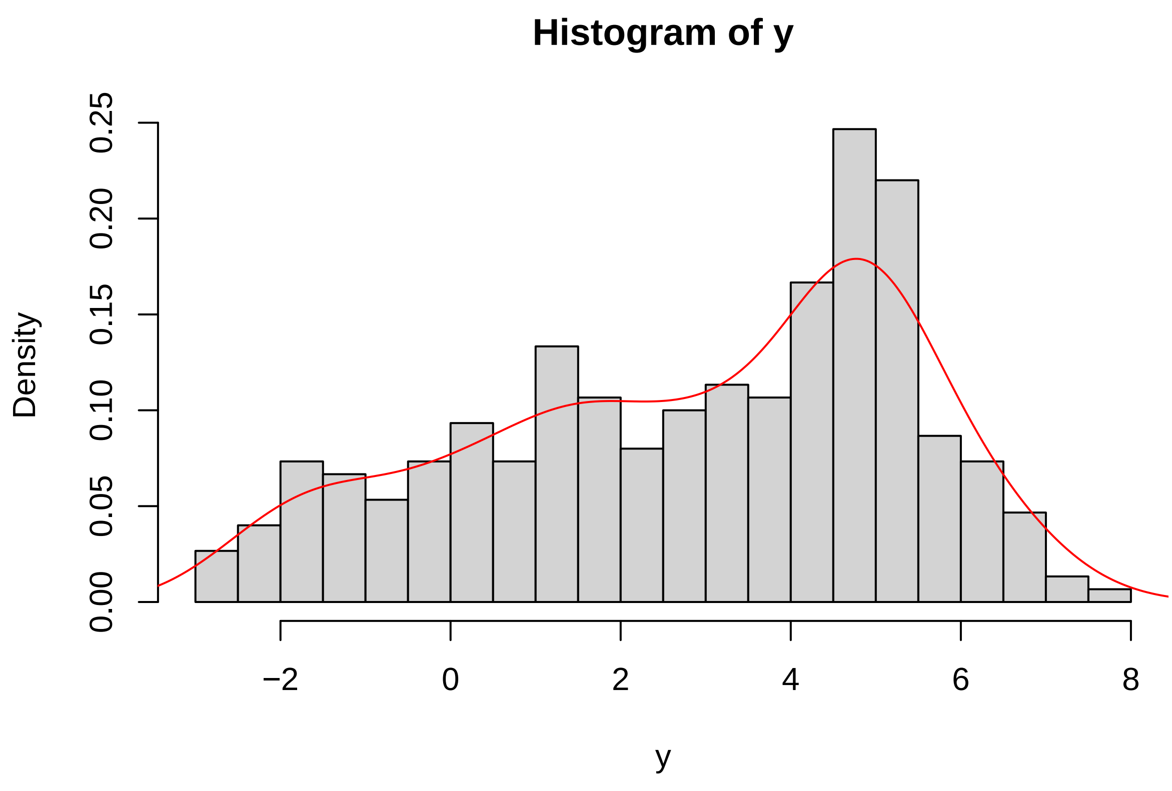
The convergence ratio from the output from Part (a) is $k = \frac{||\tilde{\theta}-\theta^*||}{||\theta-\theta^*||}$. From the output it appears k converges to the value 0.4949, which is less than 1, and therefore the EM Algorithm converges linearly for this problem.

Exercise J-4.2

Part (a)

```r
y <- read.table('C:/Users/mikej/Desktop/CSUF/Math 534/Homework_5_Part_I/ExJ42.txt')
y <- y[-1,]
y <- as.numeric(y)
y <- as.matrix(y)

hist(y,breaks = 20,freq = FALSE)
lines(density(y),col='red')
```



Part (b)

Pseudo EM Algorithm

Start with initial guesses for $\alpha, \beta, \mu_1, \mu_2, \mu_3$, and σ^2.

E-Step: Compute $E^*(z_{ij}) = \frac{f_j(y_i|\mu_j,\sigma^2)\pi_j}{\sum_{k=1}^{3} f_k(y_i|\mu_k,\sigma^2)\pi_k}$,

where $\pi = (\alpha, \beta, 1-\alpha-\beta)$

M-Step: Obtain:

$\tilde{\alpha} = \frac{\sum_{i=1}^{n} E^*(z_{i1})}{n}$

$\tilde{\beta} = \frac{\sum_{i=2}^{n} E^*(z_{i1})}{n}$

$\tilde{\mu_j} = \frac{\sum_{i=1}^{n} E^*(z_{ij})y_i}{\sum_{i=1}^{n} E^*(z_{ij})}$

$\tilde{\sigma^2} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} E^*(z_{ij})(y_i-\mu_j)^2}{n}$

Replace θ with $\tilde{\theta}$ and go back the the E-step.

Repeat until convergence.

Part(c)

```
EM_mixture <- function(y, theta, maxit, tolerr){
  n <- length(y)
  for (it in 1:maxit) {
    #E-Step
    alpha <- theta[1]
    beta <- theta[2]
    pi <- c(alpha, beta, 1-alpha-beta)
    mu1 <- theta[3]
    mu2 <- theta[4]
    mu3 <- theta[5]
    sigma2 <- theta[6]
    f1 <- dnorm(y, mean = mu1, sd=sqrt(sigma2))
    f2 <- dnorm(y, mean = mu2, sd=sqrt(sigma2))
    f3 <- dnorm(y, mean = mu3, sd=sqrt(sigma2))
    N1 <- f1*pi[1]
    N2 <- f2*pi[2]
    N3 <- f3*pi[3]
    D <- N1 + N2 + N3
    Post1 <- N1/D #Posterior probability of belonging to group 1
    Post2 <- N2/D #Posterior probability of belonging to group 2
    Post3 <- N3/D #Posterior probability of belonging to group 3
    Z <- cbind(Post1,Post2,Post3)

    # M-Step
    alpha_tilde <- sum(Z[,1])/n
    beta_tilde <-  sum(Z[,2])/n
    mu1_tilde <- sum(Z[,1]*y)/sum(Z[,1])
    mu2_tilde <- sum(Z[,2]*y)/sum(Z[,2])
    mu3_tilde <- sum(Z[,3]*y)/sum(Z[,3])
    sigma2_tilde <- (sum(Z[,1]*((y - matrix(rep(mu1, times=n), nrow=n))^2)) +
                     sum(Z[,2]*((y - matrix(rep(mu2, times=n), nrow=n)))^2) +
                     sum(Z[,3]*((y - matrix(rep(mu3, times=n), nrow=n))^2)))/n

    #Compute the log-likelihood
    log_likelihood <- sum(Z[,1]*(log(dnorm(y,mu1_tilde,sqrt(sigma2_tilde))) +log(alpha_tilde))) +
      sum(Z[,2]*(log(dnorm(y,mu2_tilde,sqrt(sigma2_tilde))) +log(beta_tilde))) +
      sum(Z[,3]*(log(dnorm(y,mu3_tilde,sqrt(sigma2_tilde))) +log(1-alpha_tilde-beta_tilde)))

    #Convergence criteria
```

```r
    theta_tilde <- c(alpha_tilde, beta_tilde, mu1_tilde, mu2_tilde, mu3_tilde, sigma2_tilde)
    mod_rel_err <- max(abs((theta_tilde - theta) / max(1,theta_tilde)))
    print(sprintf('it = %3.0f   log_likelihood = %12.12f     MRE=%2.1e',
                  it, log_likelihood , mod_rel_err), quote = FALSE)
    theta <- theta_tilde
    if(mod_rel_err < tolerr) {
      break
    }
  }
  return(list(alpha = theta[1], beta = theta[2], mu1 = theta[3],
      mu2 = theta[4], mu3 = theta[5], sigma2 = theta[6], Z=Z))
}


# Initial parameter guesses.
theta0 <- c(.1, .2, 0, 1, 2, 1)
# Run the EM algorithm
theta <- EM_mixture(y, theta0, maxit=200, tolerr = 1e-6)
```

```
## [1] it =   1   log_likelihood = -828.573361926917     MRE=8.2e-01
## [1] it =   2   log_likelihood = -847.376545247625     MRE=3.6e-01
## [1] it =   3   log_likelihood = -839.855640621881     MRE=1.2e-01
## [1] it =   4   log_likelihood = -829.055718039915     MRE=1.1e-01
## [1] it =   5   log_likelihood = -817.878180641273     MRE=9.8e-02
## [1] it =   6   log_likelihood = -808.398016839742     MRE=7.8e-02
## [1] it =   7   log_likelihood = -802.001251476859     MRE=5.4e-02
## [1] it =   8   log_likelihood = -798.526998770470     MRE=3.5e-02
## [1] it =   9   log_likelihood = -796.991170581882     MRE=2.2e-02
## [1] it =  10   log_likelihood = -796.481039119413     MRE=1.4e-02
## [1] it =  11   log_likelihood = -796.431953954026     MRE=9.4e-03
## [1] it =  12   log_likelihood = -796.551061329122     MRE=7.7e-03
## [1] it =  13   log_likelihood = -796.697897377332     MRE=7.3e-03
## [1] it =  14   log_likelihood = -796.807957114947     MRE=7.1e-03
## [1] it =  15   log_likelihood = -796.852822979920     MRE=7.2e-03
## [1] it =  16   log_likelihood = -796.820197001719     MRE=7.4e-03
## [1] it =  17   log_likelihood = -796.703751046794     MRE=7.9e-03
## [1] it =  18   log_likelihood = -796.497873780821     MRE=8.5e-03
## [1] it =  19   log_likelihood = -796.194945112223     MRE=9.3e-03
## [1] it =  20   log_likelihood = -795.783935428070     MRE=1.0e-02
## [1] it =  21   log_likelihood = -795.249694350829     MRE=1.1e-02
## [1] it =  22   log_likelihood = -794.572596176300     MRE=1.2e-02
## [1] it =  23   log_likelihood = -793.728373935990     MRE=1.3e-02
## [1] it =  24   log_likelihood = -792.688050374327     MRE=1.4e-02
## [1] it =  25   log_likelihood = -791.417883979494     MRE=1.5e-02
## [1] it =  26   log_likelihood = -789.879213085658     MRE=1.6e-02
## [1] it =  27   log_likelihood = -788.028052121566     MRE=1.7e-02
## [1] it =  28   log_likelihood = -785.814390554164     MRE=1.7e-02
## [1] it =  29   log_likelihood = -783.181578697131     MRE=1.7e-02
## [1] it =  30   log_likelihood = -780.067207587940     MRE=1.7e-02
## [1] it =  31   log_likelihood = -776.408570401906     MRE=1.6e-02
## [1] it =  32   log_likelihood = -772.157525670545     MRE=1.5e-02
## [1] it =  33   log_likelihood = -767.309369690879     MRE=1.4e-02
## [1] it =  34   log_likelihood = -761.944428050459     MRE=1.5e-02
```

```
## [1] it =  35   log_likelihood = -756.266383327765     MRE=1.4e-02
## [1] it =  36   log_likelihood = -750.603916729834     MRE=1.3e-02
## [1] it =  37   log_likelihood = -745.345597722240     MRE=1.1e-02
## [1] it =  38   log_likelihood = -740.822914213470     MRE=9.0e-03
## [1] it =  39   log_likelihood = -737.210537372520     MRE=6.9e-03
## [1] it =  40   log_likelihood = -734.505990041782     MRE=5.1e-03
## [1] it =  41   log_likelihood = -732.583365343959     MRE=3.6e-03
## [1] it =  42   log_likelihood = -731.268527671768     MRE=2.5e-03
## [1] it =  43   log_likelihood = -730.393930454436     MRE=2.0e-03
## [1] it =  44   log_likelihood = -729.823552695244     MRE=1.6e-03
## [1] it =  45   log_likelihood = -729.457077940072     MRE=1.3e-03
## [1] it =  46   log_likelihood = -729.224618796992     MRE=1.1e-03
## [1] it =  47   log_likelihood = -729.079111802824     MRE=8.7e-04
## [1] it =  48   log_likelihood = -728.989493628829     MRE=7.1e-04
## [1] it =  49   log_likelihood = -728.935504953188     MRE=5.8e-04
## [1] it =  50   log_likelihood = -728.904031529259     MRE=4.7e-04
## [1] it =  51   log_likelihood = -728.886630608670     MRE=3.9e-04
## [1] it =  52   log_likelihood = -728.877894341387     MRE=3.2e-04
## [1] it =  53   log_likelihood = -728.874378952614     MRE=2.6e-04
## [1] it =  54   log_likelihood = -728.873908007438     MRE=2.2e-04
## [1] it =  55   log_likelihood = -728.875120340745     MRE=1.8e-04
## [1] it =  56   log_likelihood = -728.877177210541     MRE=1.5e-04
## [1] it =  57   log_likelihood = -728.879572861849     MRE=1.2e-04
## [1] it =  58   log_likelihood = -728.882012204605     MRE=1.0e-04
## [1] it =  59   log_likelihood = -728.884332033721     MRE=8.3e-05
## [1] it =  60   log_likelihood = -728.886450483784     MRE=6.9e-05
## [1] it =  61   log_likelihood = -728.888334773714     MRE=5.7e-05
## [1] it =  62   log_likelihood = -728.889980778668     MRE=4.7e-05
## [1] it =  63   log_likelihood = -728.891400229350     MRE=3.9e-05
## [1] it =  64   log_likelihood = -728.892612810898     MRE=3.2e-05
## [1] it =  65   log_likelihood = -728.893641391651     MRE=2.7e-05
## [1] it =  66   log_likelihood = -728.894509235944     MRE=2.2e-05
## [1] it =  67   log_likelihood = -728.895238461098     MRE=1.8e-05
## [1] it =  68   log_likelihood = -728.895849262945     MRE=1.5e-05
## [1] it =  69   log_likelihood = -728.896359605739     MRE=1.3e-05
## [1] it =  70   log_likelihood = -728.896785183536     MRE=1.0e-05
## [1] it =  71   log_likelihood = -728.897139531946     MRE=8.6e-06
## [1] it =  72   log_likelihood = -728.897434215435     MRE=7.1e-06
## [1] it =  73   log_likelihood = -728.897679044929     MRE=5.9e-06
## [1] it =  74   log_likelihood = -728.897882299227     MRE=4.9e-06
## [1] it =  75   log_likelihood = -728.898050935582     MRE=4.0e-06
## [1] it =  76   log_likelihood = -728.898190782032     MRE=3.3e-06
## [1] it =  77   log_likelihood = -728.898306708538     MRE=2.8e-06
## [1] it =  78   log_likelihood = -728.898402776536     MRE=2.3e-06
## [1] it =  79   log_likelihood = -728.898482367950     MRE=1.9e-06
## [1] it =  80   log_likelihood = -728.898548295444     MRE=1.6e-06
## [1] it =  81   log_likelihood = -728.898602895978     MRE=1.3e-06
## [1] it =  82   log_likelihood = -728.898648109762     MRE=1.1e-06
## [1] it =  83   log_likelihood = -728.898685546626     MRE=8.9e-07
```

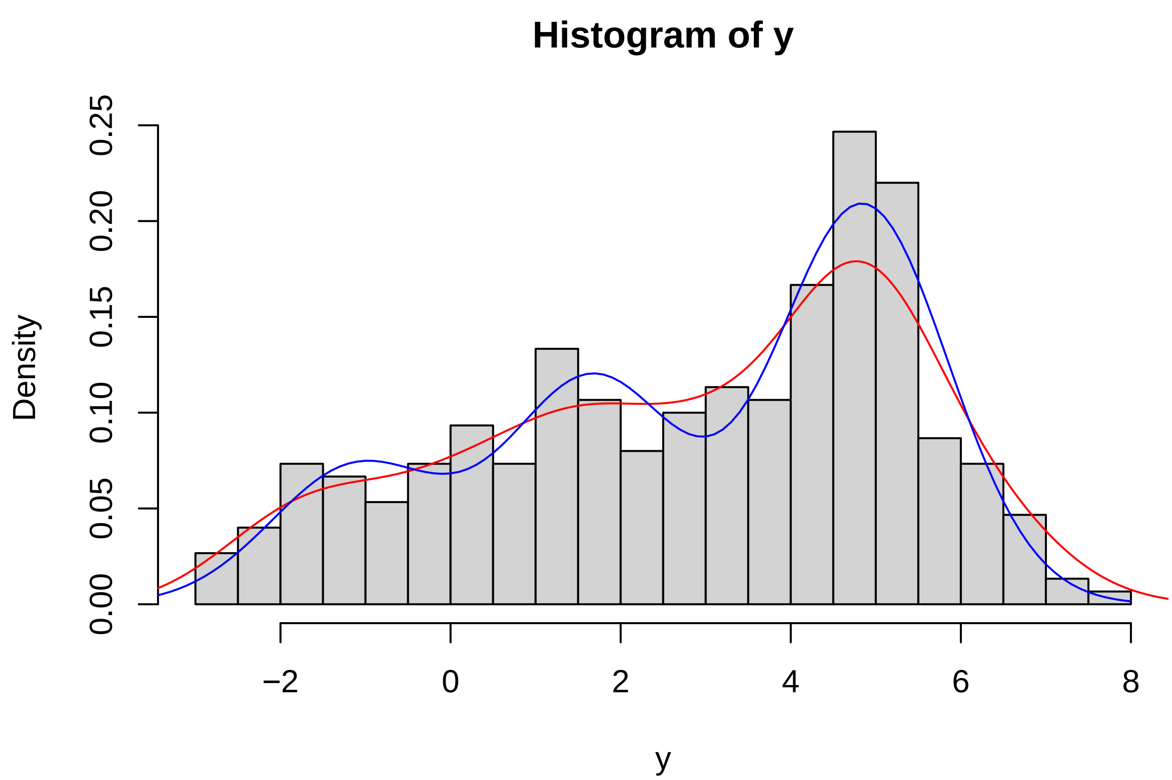
```r
list(alpha=theta$alpha, beta=theta$beta, mu1=theta$mu1, mu2=theta$mu2, mu3=theta$mu3, sigma2=theta$sigma
```

```
## $alpha
## [1] 0.1808072
```

```
##
## $beta
## [1] 0.2954499
##
## $mu1
## [1] -1.099106
##
## $mu2
## [1] 1.68078
##
## $mu3
## [1] 4.849154
##
## $sigma2
## [1] 1.005058
```

Part (d)

```
hist(y,breaks = 20,freq = FALSE)
lines(density(y),col='red')
x1 = seq(-4,8,0.1)
y1 <- theta$alpha*dnorm(x1, theta$mu1, sqrt(theta$sigma2)) +
  theta$beta*dnorm(x1, theta$mu2, sqrt(theta$sigma2)) +
  (1-theta$alpha-theta$beta)*dnorm(x1, theta$mu3, sqrt(theta$sigma2))
lines(x1,y1, col='blue')
```



Part (e)

```r
n=length(y)
class = numeric(n)
for (i in 1:n) {
  class[i] = which(theta$Z[i,] == max(theta$Z[i,]))
}
cases <- data.frame(case = 1:n, class = class)
library(ggplot2)
ggplot(data = cases) +
  geom_point(mapping = aes(x = case, y = class, color = factor(class))) +
  scale_color_manual("class", values = c("red", "green", "blue"))
```