EM for Univariate Gaussian Mixture

Kehinde Fagbamigbe

2022-08-17

Expectation-Maximization Algorithm (Mixture of Univariate Gaussian)

E-Step

```
x1 = c(-3.28,-1.4,-1.57,-2.02,0.95,2.24,3.02,2.00,2.91,4.43)
x1
```

```
##  [1] -3.28 -1.40 -1.57 -2.02  0.95  2.24  3.02  2.00  2.91  4.43
```

```
class(x1)
```

```
## [1] "numeric"
```

```
#initial guess

mu_init = c(-2,2,5) #the initial mean of the three normal/gaussian distributions
tau_init = c(0.2,0.5,0.3) # Mixing proportion/weight of the mixtures must always add up to 1, a restric
sigma_init = c(1.2,0.5,2) #the initial variance of the three normal/gaussian distributions
```

```
E.step <- function(x, tau, Mu, S2){  #tau is mixture proportion, Mu is mean S2 is standard deviation
 K <- length(tau)
 # cat("K", K, "\n")
 # cat("Tau", tau, "\n")
 n <- length(x)
 Pi <- matrix(NA, n, K)
 for (i in 1:n){
  for (k in 1:K){
   Pi[i,k] <- tau[k] * dnorm(x[i], Mu[k], sqrt(S2[k]))  #dnorm means normal distribution
   # cat("pi", i, k, Pi[i,k], "\n")
  }
  Pi[i,] <- Pi[i,] /sum(Pi[i,])
 }
return((Pi))
}
```

```
E.step(x1,tau_init,mu_init,sigma_init)
```

```
##               [,1]         [,2]         [,3]
##  [1,] 9.999999e-01 5.985338e-12 8.279641e-08
##  [2,] 9.999089e-01 4.292458e-05 4.820543e-05
##  [3,] 9.999620e-01 1.220248e-05 2.582496e-05
##  [4,] 9.999944e-01 3.713155e-07 5.184244e-06
```

```
##  [5,] 1.998825e-02 9.655629e-01 1.444883e-02
##  [6,] 1.457709e-04 9.546767e-01 4.517753e-02
##  [7,] 1.525790e-05 7.583418e-01 2.416430e-01
##  [8,] 3.184196e-04 9.690407e-01 3.064084e-02
##  [9,] 2.084629e-05 8.127207e-01 1.872584e-01
## [10,] 3.049581e-08 9.759543e-03 9.902404e-01
```

```
Pi = E.step(x1,tau_init,mu_init,sigma_init)
Pi
```

```
##               [,1]         [,2]         [,3]
##  [1,] 9.999999e-01 5.985338e-12 8.279641e-08
##  [2,] 9.999089e-01 4.292458e-05 4.820543e-05
##  [3,] 9.999620e-01 1.220248e-05 2.582496e-05
##  [4,] 9.999944e-01 3.713155e-07 5.184244e-06
##  [5,] 1.998825e-02 9.655629e-01 1.444883e-02
##  [6,] 1.457709e-04 9.546767e-01 4.517753e-02
##  [7,] 1.525790e-05 7.583418e-01 2.416430e-01
##  [8,] 3.184196e-04 9.690407e-01 3.064084e-02
##  [9,] 2.084629e-05 8.127207e-01 1.872584e-01
## [10,] 3.049581e-08 9.759543e-03 9.902404e-01
```

```
class(Pi)
```

```
## [1] "matrix" "array"
```

```
dim(Pi)
```

```
## [1] 10  3
```

```
dim(Pi)[2]
```

```
## [1] 3
```

Maximization Step

M-Step

```
M.step <- function(x, Pi){

 K <- dim(Pi)[2]
 n <- dim(Pi)[1]

 Sum.Pi <- apply(Pi, 2, sum) #2 means column summation 1 means sum by rows
 # cat("Sum.Pi", Sum.Pi, "\n")

 tau <- Sum.Pi / n

 Mu <- rep(0, K) #repeat 0 in K number of time
 S2 <- rep(0, K)

 for (k in 1:K){

  for (i in 1:n){
   Mu[k] <- Mu[k] + Pi[i,k] * x[i] #is the Mu needed here since it is zero?  #calculating the new mean
  }
```

```
  Mu[k] <- Mu[k] / Sum.Pi[k]

  for (i in 1:n){
   S2[k] <- S2[k] + Pi[i,k] * (x[i] - Mu[k])^2 #is the S2 needed here since it is zero?  #calculating t
  }
  S2[k] <- S2[k] / Sum.Pi[k]

 }

 return(list(tau = tau, Mu = Mu, S2 = S2))

}
```

```
M.step(x1,Pi)
```

```
## $tau
## [1] 0.4020354 0.4470158 0.1509488
##
## $Mu
## [1] -2.051994  2.168202  3.867228
##
## $S2
## [1] 0.5858546 0.5622469 0.6693368
```

```
class(M.step(x1,Pi))
```

```
## [1] "list"
```

```
new_element <- M.step(x1,Pi)
```

```
new_element$tau
```

```
## [1] 0.4020354 0.4470158 0.1509488
```

Log Likelihood

```
logL <- function(x, tau, Mu, S2){

 n <- length(x)
 K <- length(tau)

 ll <- 0

 for (i in 1:n){

  ll2 <- 0

  for (k in 1:K){
   ll2 <- ll2 + tau[k] * dnorm(x[i], Mu[k], sqrt(S2[k]))
  }

  ll <- ll + log(ll2)

 }
```

```
 return(ll)

}
```

```
logL(x1, new_element$tau, new_element$Mu, new_element$S2)
```

```
## [1] -19.97625
```

```
EM <- function(x, tau, Mu, S2, eps){

 n <- length(x)
 K <- length(tau)

 b <- 0

 ll.old <- -Inf
 cat("ll.old", ll.old, "\n")
 ll <- logL(x, tau, Mu, S2)

 # cat("Iteration", b, "logL =", ll, "\n")

 repeat{

  b <- b + 1

  if ((ll - ll.old) / abs(ll) < eps) break

  ll.old <- ll

  Pi <- E.step(x, tau, Mu, S2)

  M <- M.step(x, Pi)
  tau <- M$tau
  Mu <- M$Mu
  S2 <- M$S2

  ll <- logL(x, tau, Mu, S2)

  # cat("Iteration", b, "logL =", ll, "\n")

 }

 id <- apply(Pi, 1, which.max) #choose the maximmum row value Q. Is there a reason we want the maximum

 M <- 3 * K - 1
 BIC <- -2 * ll + M * log(n) #calculation of Bayesian Information Criterion
 AIC <- -2 * ll + M * 2 #Calculation for Akaike Information Criterion

 return(list(tau = tau, Mu = Mu, S2 = S2, Pi = Pi, id = id,
  logL = ll, BIC = BIC, AIC = AIC))

}
```

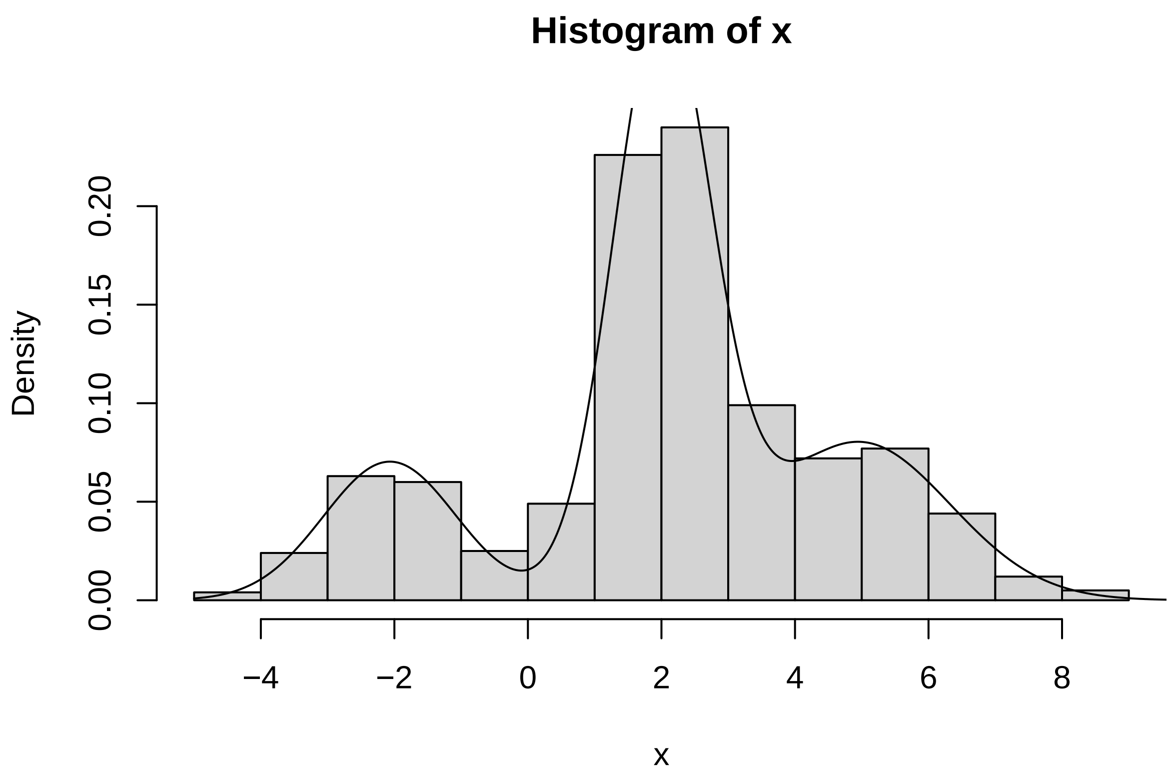
Test

```r
tau <- c(0.2, 0.5, 0.3)
Mu <- c(-2, 2, 5)
S2 <- c(1, 0.5, 2)

K <- length(tau)
n <- 1000

nk <- rmultinom(1, n, tau)  #rmultinom means multinomial distribution

x <- NULL
for (k in 1:K){
 x <- c(x, rnorm(nk[k], Mu[k], sqrt(S2[k]))) #rnorm means Normal Distribution
}
```

```r
hist(x, freq = FALSE)
tau.0 <- rep(1/3, 3)
Mu.0 <- c(-1, 0, 1)
S2.0 <- c(1, 1, 1)

A <- EM(x, tau = tau.0, Mu = Mu.0, S2 = S2.0, eps = 1e-8)
```

```
## ll.old -Inf
```

```r
t <- seq(-5, 10, by = 0.01)

d <- rep(0, length(t))
for (k in 1:K){
 d <- d + A$tau[k] * dnorm(t, A$Mu[k], sqrt(A$S2[k]))
}

points(t, d, type = "l")
```




```r
# K = 2 When mixture component is 2

tau.0 <- rep(1/2, 2)
Mu.0 <- c(-2, 1)
S2.0 <- c(1, 1)

A2 <- EM(x, tau = tau.0, Mu = Mu.0, S2 = S2.0, eps = 1e-8)
```

```
## ll.old -Inf
```

```r
A2$logL
```

```
## [1] -2274.917
```

```r
A2$BIC
```

```
## [1] 4584.374
```

```r
# K = 3 When mixture component is 3

tau.0 <- rep(1/3, 3)
Mu.0 <- c(-1, 0, 1)
S2.0 <- c(1, 1, 1)

A3 <- EM(x, tau = tau.0, Mu = Mu.0, S2 = S2.0, eps = 1e-8)
```

```
## ll.old -Inf
```

```r
A3$logL
```

```
## [1] -2167.946
```

```r
A3$BIC
```

```
## [1] 4391.155
```

```r
# K = 4 When mixture component is 4

tau.0 <- rep(1/4, 4)
Mu.0 <- c(-2, -1, 0, 1)
S2.0 <- c(1, 1, 1, 1)

A4 <- EM(x, tau = tau.0, Mu = Mu.0, S2 = S2.0, eps = 1e-8)
```

```
## ll.old -Inf
```

```r
A4$logL
```

```
## [1] -2165.282
```

```r
A4$BIC
```

```
## [1] 4406.549
```

```r
# K = 5 When mixture component is 5

tau.0 <- rep(1/5, 5)
Mu.0 <- c(-2, -1, 0, 1, 2)
S2.0 <- c(1, 1, 1, 1, 1)

A5 <- EM(x, tau = tau.0, Mu = Mu.0, S2 = S2.0, eps = 1e-8)
```

```
## ll.old -Inf
```

```r
A5$logL
```

```
## [1] -2165.257
```

```r
A5$BIC
```

```
## [1] 4427.222
```

```r
x1 = c(-3.28,-1.4,-1.57,-2.02,0.95,2.24,3.02,2.00,2.91,4.43)

tau_init = c(0.2,0.5,0.3) #must always add up to 1
mu_init = c(-2,2,5) #the mean of the two normal/gaussian distributions
sigma_init = c(1.2,0.5,2)


Univariate_Gaussian_mixture <- EM(x, tau = tau_init, Mu = mu_init, S2 = sigma_init, eps = 1e-8)
```

```
## ll.old -Inf
```

```r
Univariate_Gaussian_mixture$logL
```

```
## [1] -2167.946
```

```r
Univariate_Gaussian_mixture$BIC
```

```
## [1] 4391.155
```