Assignment_3_8259_binfeng2

Bin Feng

```r
# include library
library(mclust)
```

```
## Package 'mclust' version 5.4.2
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
Estep <- function(data, G, para){
  # Your Code
  # return the n-by-G probability matrix
  pr = para$prob
  mu = para$mean
  inv = solve(para$Sigma)
  n = nrow(data)
  tmp = NULL
  for(k in 1:G){
    tmp = cbind(tmp,
                apply(data, 1,
                      function(x) t(x - mu[, k]) %*% inv %*% (x - mu[, k])))
  }
  tmp = -tmp/2 + matrix(log(pr), nrow=n, ncol=G, byrow=TRUE)
  tmp = exp(tmp)
  tmp = tmp / apply(tmp, 1, sum)
  return(tmp)
}

Mstep <- function(data, G, para, post.prob){
  # Your Code
  # Return the updated parameters
  # update prob
  prob.new <- apply(post.prob, 2, sum) / nrow(post.prob)
  # update mean
  mean.new <- sweep(t(data) %*% post.prob, 2, apply(post.prob, 2, sum), '/')
  data = data.matrix(data)
  sigma.new = 0
  # update sigma
  for(i in 1:G){
    tmp = t(sweep(data, 2, mean.new[, i], FUN = '-')) %*%
      ((sweep(data, 2, mean.new[, i], FUN = '-')) * post.prob[,i])
    tmp = tmp / sum(post.prob)
    sigma.new = sigma.new + tmp
  }

  return(list(prob = prob.new, mean = mean.new, Sigma = sigma.new))
}

myEM <- function(data, T, G, para){
  for(t in 1:T){
    post.prob <- Estep(data, G, para)
    para <- Mstep(data, G, para, post.prob)
```

```
  }
  return(para)
}
```

```
n <- nrow(faithful)
Z <- matrix(0, n, 2)
Z[sample(1:n, 120), 1] <- 1
Z[, 2] <- 1 - Z[, 1]

ini0 <- mstep(modelName="EEE", faithful , Z)$parameters
# Output from my EM alg
para0 <- list(prob = ini0$pro, mean = ini0$mean,
              Sigma = ini0$variance$Sigma)
myEM(data = faithful, T = 10, G = ncol(faithful), para = para0)
```

```
## $prob
## [1] 0.4412314 0.5587686
##
## $mean
##                [,1]      [,2]
## eruptions  3.513757  3.467273
## waiting   70.815041 70.961824
##
## $Sigma
##           eruptions  waiting
## eruptions  1.297406  13.9281
## waiting   13.928101 184.1385
```

```
# Output from mclust
Rout <- em(modelName = "EEE", data = faithful,
           control = emControl(eps=0, tol=0, itmax = 10),
           parameters = ini0)$parameters
list(Rout$pro, Rout$mean, Rout$variance$Sigma)
```

```
## [[1]]
## [1] 0.4412314 0.5587686
##
## [[2]]
##                [,1]      [,2]
## eruptions  3.513757  3.467273
## waiting   70.815041 70.961824
##
## [[3]]
##           eruptions  waiting
## eruptions  1.297406  13.9281
## waiting   13.928101 184.1385
```

The EM algorithm and its comparison with mclust package are shown above. Note that the results from my EM algorithm and the one from mclust are exactly the same.

Please find the derivation of the E and M steps in the pdf file.

Coding Assignment 3 (Derivation)

Bin Feng UIN. 656328259

1. E step.

Let θ_0 denote the current value of θ, we want to find $p(Z|x,\theta_0)$

Since we know π_k, μ_k, Σ, we know the distribution. Then, WOLG:

$$p(Z_n=k|x_i,\theta_0) = r_{nk} = \frac{\pi_k N(x_i|\mu_k,\Sigma)}{\sum_j \pi_j N(x_i|\mu_j,\Sigma)}$$

With $N(x_n|\mu_k,\Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(x_i-\mu_k)^T\Sigma^{-1}(x_i-\mu_k)\}$

The constant term in the front cancelled out, then we have:

$$p(Z_n=k|x_i,\theta_0) = r_{nk} = \frac{\pi_k \cdot \exp\{-\frac{1}{2}(x_i-\mu_k)^T\Sigma^{-1}(x_i-\mu_k)\}}{\sum_j \pi_j \cdot \exp\{-\frac{1}{2}(x_i-\mu_j)^T\Sigma^{-1}(x_i-\mu_j)\}} \quad ☆$$

*In the R code, the procedures are as follow:

① For each element, calculate $(x_i-\mu_k)^T\Sigma^{-1}(x_i-\mu_k)$

② Add up: $\log\pi_k - \frac{1}{2}(x_i-\mu_k)^T\Sigma^{-1}(x_i-\mu_k)$

③ Take exponential: $\pi_k \cdot \exp\{-\frac{1}{2}(x_i-\mu_k)^T\Sigma^{-1}(x_i-\mu_k)\}$

④ Calculate property as above $p(Z_n=k|x_i,\theta_0)$

⑤ Return $p(Z_n=k|x_i,\theta_0)$. a $n\times G$ matrix.

2. M step

The M step is to maximize the expected complete data log likelihood:

$$\begin{aligned}
g(\theta) &= \mathbb{E}_{Z|x,\theta_0} \log p(x,Z|\theta) \\
&= \mathbb{E}_{Z|x,\theta_0} \sum_n \log \prod_k \pi_k N(x_n|\mu_k,\Sigma)^{I(Z_n=k)} \\
&= \mathbb{E}_{Z|x,\theta_0} \sum_n\sum_k I_{(Z_n=k)} \log[\pi_k N(x_n|\mu_k,\Sigma)] \\
&= \sum_n\sum_k r_{nk} \log[\pi_k N(x_n|\mu_k,\Sigma)] \\
&= \sum_n\sum_k r_{nk}\log\pi_k + \sum_n\sum_k r_{nk} N(x_n|\mu_k,\Sigma)
\end{aligned}$$

① Update π_k, add a Lagrange multiplier to constrait $\sum_k \pi_k = 1$

$$\Rightarrow \left.\begin{aligned}
&\frac{\partial}{\partial\pi_i}\left[\sum_n\sum_k r_{nk}\log\pi_k + \lambda(1-\sum_k\pi_k)\right] = \sum_n r_{nk}/\pi_k - \lambda = 0 \\
&\frac{\partial}{\partial\lambda}\left[\sum_n\sum_k r_{nk}\log\pi_k + \lambda(1-\sum_k\pi_k)\right] = 1-\sum_k\pi_k = 0
\end{aligned}\right\}$$

$$\Rightarrow \lambda = \frac{\sum_n r_{nk}}{\pi_k} = \frac{\sum_n\sum_k r_{nk}}{1} = N$$

$$\Rightarrow \pi_k = \frac{1}{N}\sum_n r_{nk} \quad ☆$$

② Update μ_k

$$\frac{\partial}{\partial \mu_k} g(\theta) \propto \frac{\partial}{\partial \mu_k}\left\{-\frac{1}{2}\sum_n \sum_k r_{nk}\left[\log|\Sigma| + (x_n-\mu_k)^T \Sigma^{-1} (x_n-\mu_k)\right]\right\}$$

$$= + \sum_n r_{nk}(x_n-\mu_k)\Sigma^{-1} = 0$$

drop constant $\propto \sum_n r_{nk} x_n - \sum_n r_{nk}\mu_k = 0$

$$\Rightarrow \quad \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}} \quad ☆$$

③ Update Σ

$$\frac{\partial}{\partial \Sigma} g(\theta) \propto \frac{\partial}{\partial \Sigma}\left\{-\frac{1}{2}\sum_n \sum_k r_{nk}\left[\log|\Sigma| + (x_n-\mu_k)^T \Sigma^{-1} (x_n-\mu_k)\right]\right\}$$

$$= -\frac{1}{2}\sum_n \sum_k r_{nk}\left[\frac{1}{|\Sigma|} - \frac{(x_n-\mu_k)^T (x_n-\mu_k)}{\Sigma^{-2}}\right] = 0$$

$$\Rightarrow \Sigma = \frac{\sum_k \sum_n r_{nk}(x_n-\mu_k)^T(x_n-\mu_k)}{\sum_k \sum_n r_{nk}} \quad ☆$$

In the R code, procedures are as follow:

① Update π_k as the derived formular

② Update μ_k as the derived formular

③ Recursively calculate $\sum_k \left[\frac{\sum_n r_{nk}(x_n-\mu_k)^T(x_n-\mu_k)}{\sum_k \sum_n r_{nk}}\right]$