Homework 5 - STAT 5362 Statistical Computing

*Sen Yang**

14 October 2018

Abstract

This is homework 5 for STAT 5362 - Statistical Computing.

Contents

*sen.2.yang@uconn.edu; M.S. student at Department of Statistics, University of Connecticut.

1 Finite mixture regression

1.1 Validity of the provided E-step and M-step

E-Step:

$$
\begin{aligned}
&Q(\mathbf{\Psi}|\mathbf{\Psi}^{(k)}) \\
&= \mathbb{E}[l_n^c(\mathbf{\Psi})|y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)}] \\
&= \sum_z p(z|y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)}) l_n^c(\mathbf{\Psi}) \\
&= \sum_z p(z|y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)}) \sum_{i=1}^n \sum_{j=1}^m z_{ij} \log\{\pi_j \phi(y_i - \mathbf{x}_i^\top \beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^n \sum_{j=1}^m [\sum_z p(z|y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)}) z_{ij}] \log\{\pi_j \phi(y_i - \mathbf{x}_i^\top \beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^n \sum_{j=1}^m E(z_{ij}|y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)}) \log\{\pi_j \phi(y_i - \mathbf{x}_i^\top \beta_j; 0, \sigma^2)\}
\end{aligned}
$$

Here,

$$
\begin{aligned}
&E(z_{ij}|y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)}) \\
&= p(z_{ij} = 1|y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)}) \\
&= \frac{p(z_{ij} = 1, y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)})}{p(y_i, \mathbf{x}_i; \mathbf{\Psi}^{(k)})} \\
&= \frac{\pi_j^{(k)} \phi(y_i - \mathbf{x}_i^\top \beta_j^{(k)}; 0, \sigma^{2^{(k)}})}{\sum_{j=1}^m \pi_j^{(k)} \phi(y_i - \mathbf{x}_i^\top \beta_j^{(k)}; 0, \sigma^{2^{(k)}})} \\
&= p_{ij}^{(k+1)}
\end{aligned}
$$

M-Step:

Since $\phi(y_i - \mathbf{x}_i^\top \beta_j; 0, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp[-\frac{1}{2}\frac{(y_i - \mathbf{x}_i^\top \beta_j)^2}{\sigma^2}]$, then we have

$$
\begin{aligned}
&Q(\mathbf{\Psi}|\mathbf{\Psi}^{(k)}) \\
&= \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \{\log \pi_j + \log \phi(y_i - \mathbf{x}_i^\top \beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \{\log \pi_j - \frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2}(y_i - \mathbf{x}_i^\top \beta_j)^2\} \\
&= \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \log \pi_j - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \log 2\pi\sigma^2 - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \frac{(y_i - \mathbf{x}_i^\top \beta_j)^2}{\sigma^2} \\
&= I_1 - \frac{I_2}{2} - \frac{I_3}{2}
\end{aligned}
$$

Therefore, we have $I_1 = \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \log \pi_j$, $I_2 = \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \log 2\pi\sigma^2$ and $I_3 = \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \frac{(y_i - \mathbf{x}_i^\top \beta_j)^2}{\sigma^2}$, with $\sum_{j=1}^m p_{ij} = 1$ and $\sum_{j=1}^m \pi_j = 1$.

1. For π_j, only I_1 contains π_j. Given $\sum_{j=1}^m \pi_j = 1$, we have

$$
\begin{aligned}
&\frac{\partial I_1}{\partial \pi_j} \\
&= \frac{\sum_{i=1}^n p_{ij}^{(k+1)}}{\pi_j} - \frac{\sum_{i=1}^n p_{il}^{(k+1)}}{1 - \sum_{a \neq l} \pi_a}, where \;\; l \neq j. \\
&= \frac{\sum_{i=1}^n p_{ij}^{(k+1)}}{\pi_j} - \frac{\sum_{i=1}^n p_{il}^{(k+1)}}{\pi_l}, where \;\; l \neq j. \\
&= 0 \\
&\Longrightarrow \frac{\sum_{i=1}^n p_{ij}^{(k+1)}}{\pi_j} = \frac{\sum_{i=1}^n p_{il}^{(k+1)}}{\pi_l} \\
&\Longrightarrow \pi_l = \frac{\sum_{i=1}^n p_{il}^{(k+1)} \pi_j}{\sum_{i=1}^n p_{ij}^{(k+1)}} \\
&\Longrightarrow \sum_{l=1}^m \pi_j = \sum_{l=1}^m [\frac{\sum_{i=1}^n p_{il}^{(k+1)} \pi_l}{\sum_{i=1}^n p_{ij}^{(k+1)}}] \\
&\Longrightarrow 1 = \frac{\sum_{i=1}^n \sum_{l=1}^m p_{il}^{(k+1)} \pi_j}{\sum_{i=1}^n p_{ij}^{(k+1)}} = \frac{\sum_{i=1}^n 1 \cdot \pi_j}{\sum_{i=1}^n p_{ij}^{(k+1)}} = \frac{n \cdot \pi_j}{\sum_{i=1}^n p_{ij}^{(k+1)}} \\
&\Longrightarrow \pi_j^{(k+1)} = \frac{\sum_{i=1}^n p_{ij}^{(k+1)}}{n}
\end{aligned}
$$

2. For β_j, only I_3 contains β_j.

$$
\begin{aligned}
&\frac{\partial I_3}{\partial \beta_j} \\
&= -\sum_{i=1}^n p_{ij}^{(k+1)} \cdot 2\mathbf{x}_i \frac{(y_i - \mathbf{x}_i^\top \beta_j)}{\sigma^2} = 0 \\
&\Longrightarrow \sum_{i=1}^n p_{ij}^{(k+1)} \mathbf{x}_i y_i = \sum_{i=1}^n p_{ij}^{(k+1)} \mathbf{x}_i \mathbf{x}_i^\top \beta_j \\
&\Longrightarrow \beta_j^{(k+1)} = (\sum_{i=1}^n p_{ij}^{(k+1)} \mathbf{x}_i \mathbf{x}_i^\top)^{-1} (\sum_{i=1}^n p_{ij}^{(k+1)} \mathbf{x}_i y_i)
\end{aligned}
$$

3. For σ^2, I_2 and I_3 contain σ^2.

$$
\begin{aligned}
&\frac{\partial I_2}{\partial \sigma^2} + \frac{\partial I_3}{\partial \sigma^2} \\
&= \frac{n}{\sigma^2} - (\sigma^2)^{-2} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^{(k+1)} (y_i - \mathbf{x}_i^\top \beta_j^{(k+1)})^2 = 0 \\
&\implies \sigma^{2^{(k+1)}} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^{(k+1)} (y_i - \mathbf{x}_i^\top \beta_j^{(k+1)})^2}{n}
\end{aligned}
$$

1.2 Apply EM algorithm in R with function regmix_em

```
## regmix_em
regmix_em <- function(y, xmat, pi.init, beta.init , sigma.init,
                      control = list(maxit = 500, tol = 1e-5)) {
  n <- nrow(xmat)
  k <- ncol(beta.init)
  p <- matrix(0, nrow = nrow(xmat), ncol = ncol(beta.init))
  p_nume<-p
  beta1 <- beta.init
  pi1<-pi.init
for (r in 1:control$maxit) {
  for (j in 1:k) {
    p_nume[,j] <- as.matrix((pi.init[j]*(2*3.14159265*sigma.init^2)^(-0.5)*
               exp(-(y-as.matrix(xmat)%*%as.matrix(beta.init[,j]))^2/2/sigma.init^2)))
  }
  p_deno <- rowSums(p_nume)
  p <- p_nume/p_deno
  pi1 <- colSums(p)/n
  for (j in 1:k) {
    beta1_2nd <- matrix(0, nrow = ncol(xmat), ncol=k)
    for (i in 1:n) {
      beta1_2nd[,j] <- beta1_2nd[,j] + t(xmat[i,]*p[i,j]*y[i])
    }
    beta1[,j] <- (sum(diag(as.matrix(xmat)%*%t(as.matrix(xmat)))
                      *p[,j]))^(-1)*beta1_2nd[,j]
  }
  sigma_2 <- sum( (y-as.matrix(xmat)%*%as.matrix(beta1))^2 * p)
  sigma_1 <- sqrt(sigma_2)
  if ((max(abs(pi1-pi.init)) <= control$tol) &(max(abs(beta1-beta.init)) <= control$tol) &
      (max(sigma_1-sigma.init) <= control$tol )) break
  pi.init<-pi1
  beta.init<-beta1
  sigma.init<-sigma_1
}
```

```r
  return(list(pi=pi1, beta=beta1, sigma=sigma_1, iteration=r))
}
```

1.3 Parameters estimation for generated data

```r
## regmix_sim
regmix_sim <- function(n, pi, beta, sigma) {
  K <- ncol(beta)
  p <- NROW(beta)
  xmat <- matrix(rnorm(n * p), n, p) # normal covaraites
  error <- matrix(rnorm(n * K, sd = sigma), n, K)
  ymat <- xmat %*% beta + error # n by K matrix
  ind <- t(rmultinom(n, size = 1, prob = pi))
  y <- rowSums(ymat * ind)
  data.frame(y, xmat)
}

## simulation
n <- 400
pi <- c(.3, .4, .3)
bet <- matrix(c( 1,  1,  1,
                 -1, -1, -1), 2, 3)
sig <- 1
set.seed(1205)
dat <- regmix_sim(n, pi, bet, sig)
regmix_em(y = dat[,1], xmat = dat[,-1],
           pi.init = pi/pi/length(pi),
           beta.init = bet*1,
           sigma.init = sig / sig,
           control = list(maxit = 500, tol = 1e-5))
```

```
## $pi
## [1] 0.3357209 0.3262823 0.3379968
##
## $beta
##             [,1]       [,2]       [,3]
## [1,]  0.1687649  0.1687649  0.1687649
## [2,] -0.2339078 -0.2339078 -0.2339078
##
## $sigma
## [1] 35.178
##
## $iteration
## [1] 4
```