Ex5

Guanting Wei

Oct.7.2018

1. Verifying E-step and M-step

$$
\begin{aligned}
f(y_i|x_i,\Psi) &= \sum_{j=1}^{m} \pi_j \phi(y_i; x_i^T\beta_j, \sigma^2) \\
&= \sum_{j=1}^{m} \pi_j \phi(y_i - x_i^T\beta_j; 0, \sigma^2)
\end{aligned}
$$

$$
\begin{aligned}
l_c^n(\Psi) &= \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij} \ln\{\pi_j \phi(y_i - x_i^T\beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij} \{\ln \pi_j + \ln \phi(y_i - x_i^T\beta_j; 0, \sigma^2)\}
\end{aligned}
$$

$$
\begin{aligned}
E[z_{ij}|x_i, y_i; \Psi^{(k)}] &= \frac{\pi_j^{(k)} \phi(y_i - x_i^T\beta_j; 0, \sigma^2)}{\sum_{j=1}^{m} \pi_j^{(k)} \phi(y_i - x_i^T\beta_j; 0, \sigma^2)} \\
&= p_{ij}^{(k+1)}
\end{aligned}
$$

$$
y = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_i \\ ... \\ y_n \end{pmatrix} \quad x = \begin{pmatrix} x_1 & x_2 & ... & x_i & ... & x_n \end{pmatrix}^T = \begin{pmatrix} x_{11} & x_{12} & ... & x_{1p} \\ x_{21} & x_{22} & ... & x_{2p} \\ ... & ... & ... & ... \\ x_{i1} & x_{i2} & ... & x_{ip} \\ ... & ... & ... & ... \\ x_{n1} & x_{n2} & ... & x_{np} \end{pmatrix} \beta_j = \begin{pmatrix} \beta_{j1} \\ \beta_{j2} \\ ... \\ \beta_{jp} \end{pmatrix}
$$

1.1 E-step

$$
\begin{aligned}
Q(\Psi|\Psi^{(k)}) &= E[l_n^c(\Psi)|x, y, \Psi^{(k)}] \\
&= \sum_z l_n^c(\Psi) P(z|x, y, \Psi^{(k)}) \\
&= \sum_z P(z|x, y, \Psi^{(k)}) \sum_{i=1}^n \sum_{j=1}^m z_{ij} \{\ln \pi_j + \ln \phi(y_i - x_i^T \beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^n \sum_{j=1}^m [\sum_z P(z|x, y, \Psi^{(k)}) z_{ij}] \{\ln \pi_j + \ln \phi(y_i - x_i^T \beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^n \sum_{j=1}^m E[z_{ij}|x_i, y_i; \Psi^{(k)}] \{\ln \pi_j + \ln \phi(y_i - x_i^T \beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \{\ln \pi_j + \ln \phi(y_i - x_i^T \beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{(k+1)} \{\ln \pi_j - \frac{1}{2} \ln 2\pi\sigma^2 - \frac{1}{2} \frac{(y_i - x_i^T \beta_j)^2}{\sigma^2}\}
\end{aligned}
$$

1.2 M-step

1.2.1 For π_j

$$
\begin{aligned}
\frac{\partial Q}{\partial \pi_j} &= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} \ln \pi_j \\
&= \sum_{j=1}^{m} (\ln \pi_j \sum_{i=1}^{n} p_{ij}^{(k+1)}) \\
&= \ln \pi_1 \sum_{i=1}^{n} p_{i1}^{(k+1)} + ... + \ln \pi_j \sum_{i=1}^{n} p_{ij}^{(k+1)} + ... \ln \pi_m \sum_{i=1}^{n} p_{im}^{(k+1)} \\
&= \ln \pi_1 \sum_{i=1}^{n} p_{i1}^{(k+1)} + ... + \ln \pi_j \sum_{i=1}^{n} p_{ij}^{(k+1)} + ... \ln(1 - \pi_1 - ... - \pi_j - ... - \pi_{m-1}) \sum_{i=1}^{n} p_{im}^{(k+1)} \\
&= \frac{\sum_{i=1}^{n} p_{ij}}{\pi_j} - \frac{\sum_{i=1}^{n} p_{im}}{\pi_m} = 0
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\sum_{i=1}^{n} p_{ij}}{\pi_j} = \frac{\sum_{i=1}^{n} p_{im}}{\pi_m} \\
\Rightarrow &\sum_{i=1}^{n} p_{ij} = \pi_j \frac{\sum_{i=1}^{n} p_{im}}{\pi_m} \\
\Rightarrow &\sum_{j=1}^{m}\sum_{i=1}^{n} p_{ij} = \sum_{j=1}^{m} \pi_j \frac{\sum_{i=1}^{n} p_{im}}{\pi_m} \\
\Rightarrow &\sum_{i=1}^{n} (\sum_{j=1}^{m} p_{ij}) = \frac{\sum_{i=1}^{n} p_{im}}{\pi_m} (\sum_{j=1}^{m} \pi_j) \\
\Rightarrow &\sum_{i=1}^{n} 1 = \frac{\sum_{i=1}^{n} p_{im}}{\pi_m} = n \\
\Rightarrow &\frac{\sum_{i=1}^{n} p_{ij}}{\pi_j} = n \\
\Rightarrow &\pi_j = \frac{\sum_{i=1}^{n} p_{ij}}{n}
\end{aligned}
$$

1.2.2 For β_j

$$
\begin{aligned}
\frac{\partial Q}{\partial \beta_j} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n} p_{ij}^{(k+1)} \frac{\partial (y_i - x_i^T\beta_j^{(k+1)})^2}{\partial \beta_j} \\
&= -\frac{1}{2\sigma^2}\sum_{i=1}^{n} p_{ij}^{(k+1)} 2(y_i - x_i^T\beta_j^{(k+1)}) \frac{\partial (y_i - x_i^T\beta_j^{(k+1)})}{\partial \beta_j} \\
&= -\frac{1}{2\sigma^2}\sum_{i=1}^{n} p_{ij}^{(k+1)} (-2)(y_i - x_i^T\beta_j^{(k+1)}) \frac{\partial (x_i^T\beta_j^{(k+1)})}{\partial \beta_j} \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} p_{ij}^{(k+1)} x_i(y_i - x_i^T\beta_j^{(k+1)}) = 0 \ (\beta_j \ is\ a\ column\ vector; we\ will\ get\ x_i^T\ if\ \beta_j\ is\ row\ vector) \\
\Rightarrow \beta_j^{(k+1)} &= (\sum_{i=1}^{n} x_i x_i^T p_{ij}^{(k+1)})^{-1} (\sum_{i=1}^{n} x_i p_{ij}^{(k+1)} y_i)
\end{aligned}
$$

1.2.3 For σ^2

$$
\begin{aligned}
\frac{\partial Q}{\partial \sigma^2} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} (y_i - x_i^T\beta_j^{(k+1)})^2 = 0 \\
\Rightarrow \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} (y_i - x_i^T\beta_j^{(k+1)})^2}{\sigma^2} &= n \\
\Rightarrow \sigma^{2(k+1)} &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} (y_i - x_i^T\beta_j^{(k+1)})^2}{n}
\end{aligned}
$$

2. EM function

```
regmix_em=function(y,xmat,pi.init,beta.init,sigma.init,control=list(maxit=500,tol=1e-5)){
  n=nrow(xmat)
  p=ncol(xmat)
  m=length(pi.init)

  x=as.matrix(xmat)
  pi=pi.init
  beta=beta.init
  sigma=sigma.init

  maxit=control$maxit
  tol=control$tol

  P=matrix(0,n,m)
  beta.1=matrix(0,p,m)
  count=0

  for(iter in 1:maxit){
    for(i in 1:n){
      P[i, ]=pi*dnorm(y[i]-x[i, ]%*%beta,0,sigma)/sum(pi*dnorm(y[i]-x[i, ]%*%beta,0,sigma))
    }
```

```
    pi.1=colMeans(P)

    for(j in 1:m){
      beta.1[ ,j]<-solve(t(x)%*%diag(P[,j])%*%x)%*%t(x)%*%diag(P[,j])%*%y
    }

    sigma.1=sqrt(sum((P*(y%*%t(rep(1,m))-x%*%beta.1)^2))/n)

    if(sum(abs(pi.1-pi))+sum(abs(beta.1-beta))+sum(abs(sigma.1-sigma))<tol)break
      pi=pi.1
    beta=beta.1
    sigma=sigma.1
  }
  if(iter==maxit)
  print("reach maximun loop")
  list(pi=pi.1,beta=beta.1,sigma=sigma.1,iteration=iter)
}
```

3. Generate data

```
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)
}
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)
output1=regmix_em(y = dat[,1], xmat = dat[,-1],
           pi.init = pi / pi / length(pi),
           beta.init = bet*0,
           sigma.init = sig / sig,
           control = list(maxit = 500, tol = 1e-5))
output1
```

```
## $pi
## [1] 0.3333333 0.3333333 0.3333333
##
## $beta
##             [,1]       [,2]       [,3]
## [1,]  0.3335660  0.3335660  0.3335660
## [2,] -0.4754645 -0.4754645 -0.4754645
```

```
##
## $sigma
## [1] 1.732492
##
## $iteration
## [1] 2
```

Then choosing beta as beta.init instead of 0.

```
output2=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))
output2
```

```
## $pi
## [1] 0.3858218 0.2687873 0.3453909
##
## $beta
##           [,1]       [,2]       [,3]
## [1,] 0.8796608  0.9911852 -0.9136977
## [2,] 0.9341964 -1.2424569 -1.1990372
##
## $sigma
## [1] 1.023598
##
## $iteration
## [1] 49
```