EM Project

Yiyi Xu

10/9/2018

Verify the E- and M-steps.

E- stapes

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

$$because\ z_{ij}\ is\ binary\ r.v$$

$$
= \sum_{i=1}^n \sum_{j=1}^m [\log \pi_j + \log \varphi(y_i - x_i^T\beta_j; 0, \sigma^2)] p_{ij}^{k+1}
$$

M-steps

$$
\begin{aligned}
Q(\Psi|\Psi^{(k)}) &= \sum_{i=1}^n \sum_{j=1}^m [\log \pi_j + \log \varphi(y_i - x_i^T\beta_j; 0, \sigma^2)] p_{ij}^{k+1} \\
&= \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{k+1}[\log \pi_j + \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(y_i - x_i^T\beta_j)^2}{2\sigma^2}}] \\
&= [\sum_{i=1}^n \sum_{j=1}^m p_{ij}^{k+1} \log \pi_j] - [\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m p_{ij}^{k+1} \log(2\sigma^2\pi)] - [\sum_{i=1}^n \sum_{j=1}^m p_{ij}^{k+1} \frac{(y_i - x_i^T\beta_j)^2}{2\sigma^2}]
\end{aligned}
$$

Verify

$$\pi_j^{k+1}$$

.

$$
\begin{aligned}
\frac{\partial}{\partial \pi_j} Q &= \frac{\partial}{\partial \pi_j} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^{k+1} \log \pi_j \\
&= \frac{\sum_{i=1}^{n} p_{ij}^{k+1}}{\pi_j} - \frac{\sum_{i=1}^{n} p_{im}^{k+1}}{\pi_m} \\
&= 0 \\
\frac{\sum_{i=1}^{n} p_{ij}^{k+1}}{\pi_j} &= \frac{\sum_{i=1}^{n} p_{im}^{k+1}}{\pi_m} \\
&= c \\
\sum_{i=1}^{n} p_{ij}^{k+1} &= c \sum_{j=1}^{m} \pi_j \\
c &= n \\
\pi_j^{k+1} &= \frac{\sum_{i=1}^{n} p_{ij}^{k+1}}{n}
\end{aligned}
$$

Verify

$$\beta_j^{k+1}$$

.

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

berify

$$\sigma^{2(k+1)}$$

.

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

implement the algorithm

```r
regmix_em <- function(y, xmat, pi.init, beta.init, sigma.init,
                      control=list(maxit = 100, tol = .Machine$double.eps^0.5)) {
  n <- nrow(xmat)
  c <- ncol(xmat)
  m <- length(pi.init)

  maxit <- control$maxit
  tol <- control$tol

  pi <- pi.init
  beta <- beta.init
  sigma <- sigma.init

  p <- matrix(NA, nrow = n, ncol = m)
  beta.new <- matrix(NA, nrow = c, ncol = m)

  xmat <- as.matrix(xmat)

  for (i in 1:maxit) {
    for (j in 1:n) {
      p[j, ] <- pi * dnorm(y[j] - xmat[j, ] %*% beta, mean = 0, sd = sigma) /
        sum(pi * dnorm(y[j] - xmat[j, ] %*% beta, mean = 0, sd = sigma))
    }

    pi.new <- colMeans(p)

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

    sigma.new <- sqrt(sum(p * (y %*% t(rep(1, m)) - xmat %*% beta.new)^2)/n)

    if (sum(abs(pi-pi.new))+sum(abs(beta-beta.new))+abs(sigma-sigma.new) < tol) {
      break
    }
    pi <- pi.new
    beta <- beta.new
    sigma <- sigma.new
  }
  if (i == maxit) {
    print("Reach the Maximum Iteration")
  }
  return(list(pi = pi.new, beta = beta.new, sigma = sigma.new,
              conv = 1, iter = i))
}
```

Generate data & Estimate parameters

```
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)



fit <-  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))
fit
```

```
## $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
##
## $conv
## [1] 1
##
## $iter
## [1] 2
```

Because when beta.initial equals zero, the parameter didnt update after 2 iterations, So i randomly choose another value as the beta.initial. Then, we can see it is convergence to 1 in 55 iterations.

```
fit1 <-  regmix_em(y = dat[,1], xmat = dat[,-1],
             pi.init = pi / pi / length(pi),
             beta.init = matrix(c(1,2,3,1,2,3),2,3),
             sigma.init = sig / sig,
             control = list(maxit = 500, tol = 1e-5))
fit1
```

```
## $pi
## [1] 0.3453911 0.2687873 0.3858217
##
## $beta
##            [,1]       [,2]      [,3]
## [1,] -0.9136974  0.9911857 0.8796608
## [2,] -1.1990372 -1.2424565 0.9341965
##
## $sigma
## [1] 1.023598
##
## $conv
## [1] 1
##
## $iter
## [1] 55
```