Statistical Computing Homework 5, EM algorithm

Ziqi Yang

12 October, 2018

Contents

Finite Mixture Regression:

Derive the update formula

$\Psi = (\pi_1, \beta_1, \pi_2, \beta_2, ..., \pi_m, \beta_m, \sigma)^T$, by the E-step in EM algorithm, we have:

$$
\begin{aligned}
Q(\Psi|\Psi^k) &= \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \mathbf{y}, \Psi^k) \log(p(\mathbf{x}, \mathbf{y}, \mathbf{z}|\Psi)) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} p(z_{ij} = 1|\mathbf{x}, \mathbf{y}, \Psi^k) \log(p(z_{ij} = 1, \mathbf{x}, \mathbf{y}|\Psi))
\end{aligned}
$$

In order to get the update formula for p_{ij}, which is $p(z_{ij} = 1|\mathbf{x}_i, \mathbf{y}_i, \Psi^k)$, by Bayes rule, it's equal to

$$
\begin{aligned}
&\frac{p(z_{ij} = 1, \mathbf{x}_i, \mathbf{y}_i|\Psi^k)}{p(\mathbf{x}_i, \mathbf{y}_i|\Psi^k)} \\
=&\frac{p(z_{ij} = 1|\Psi^k) p(\mathbf{x}_i, \mathbf{y}_i|z_{ij} = 1, \Psi^k)}{p(\mathbf{x}_i, \mathbf{y}_i|\Psi^k)}
\end{aligned}
$$

So if we only look at the numerator, it's $\pi_j^k \phi(y_i - x_i^T \beta_j^k; 0, \sigma^{2(k)})$. So we have the update form for p_{ij}^{k+1} as in the question if sum over j in the numerator to get the denominator. Since we also know in E-step, we have:

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

So in M-step, if we minimize the $Q(\Psi|\Psi^k)$ with respect to π_j, β_j^{k+1} and $\sigma^{2(k+1)}$, then we will get the update formula these three sets of parameters as in the questoion.

Implementation of EM algorithm

- **y**: Response vector
- **xmat**: Design matrix
- **pi.init**: Initial values of π_j's(m $\times$ 1)
- **beta.init**: Initial values of β_j's (p$\times$m, where p is ncol(xmat))

- **sigma.init**: Initial values of σ
- **control**: list for controlling max iteration number and convergence tolerance

```
regmix_em <- function(y, xmat, pi.init, beta.init, sigma.init, control = list(maxit = 1000, tol = 1e-5)
  p.iter <- matrix(0, nrow = dim(xmat)[1], ncol = length(pi.init) )
  pi.iter <- pi.init;  beta.iter <- beta.init;  sigma.iter <- sigma.init
  #temp_pi.iter <- c(0.9,0.05,0.05)
  xmat <- as.matrix(xmat)

  iter <- 1; maxit <- control$maxit; tol <- control$tol
  while ( (iter <= maxit)  ) {
    for (j in 1:dim(p.iter)[2]) {
      for (i in 1:dim(p.iter)[1]) {
        temp.sum  <- sum( pi.iter * dnorm(y[i] - xmat[i,]%*%beta.iter, mean=0, sd=sigma.iter^0.5) )
        p.iter[i, j] <- pi.iter[j] * dnorm(y[i] - xmat[i,]%*%beta.iter[ , j], mean=0, sd=sigma.iter^0.5

      }
    }
    #temp_pi.iter <- pi.iter
    for (j in 1:dim(p.iter)[2]) {
      pi.iter[j] <- mean(p.iter[,j])
      beta.iter[ ,j]<- solve( t(xmat) %*% ( xmat * matrix(rep(p.iter[,j],2), ncol = 2) ) ) %*% t(xmat) %
    }

    #temp_pi.iter <- pi.iter
    sigma.iter <- sum( p.iter * (matrix(rep(y, length(pi.iter)), ncol = length(pi.iter)) - xmat %*% beta

    iter <- iter + 1

  }

  return(list(pi.iter, beta.iter, sigma.iter))
}
```

Simulation Study

```
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
```

```r
set.seed(1205)
dat <- regmix_sim(n, pi, bet, sig)

result <- regmix_em(y = dat[,1], xmat = dat[,-1],
           #pi.init = pi / pi / length(pi),
           pi.init = c(0.8, 0.1, 0.1),
           beta.init = bet * 0 + rnorm(6),
           sigma.init = sig / sig,
           control = list(maxit = 1000, tol = 1e-5))
result
```

```
## [[1]]
## [1] 0.3453959 0.2687802 0.3858238
##
## [[2]]
##            [,1]       [,2]      [,3]
## [1,] -0.9136895  0.9911949 0.8796621
## [2,] -1.1990373 -1.2424624 0.9341928
##
## [[3]]
## [1] 1.047753
```

We need to choose initial values of π and β matrix wisely, in order to get relatively godd convergence.