HW5

Cheng Huang 2658312

22 October 2018

verify the validity of the provided E- and M-steps

$$Q(\Psi|\Psi^{(k)}) = \sum_z p(z|x,y,\Psi^{(k)}) lnp(x,y,z|\Psi)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} p(z_{ij}|(x,y,\Psi^{(k)}))z_{ij}\{log\pi_i + log\varphi(y_i - x_i^T\beta_j; 0, \sigma^2)\}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} E(z_{ij}|x,y,\Psi^{(k)})\{log\pi_i + log\varphi(y_i - x_i^T\beta_j; 0, \sigma^2)\}$$

Since z_{ij} is 1 if the ith observatiojn is from the jth group, 0 otherwise. So

$$E(z_{ij}|x,y,\Psi^{(k)}) = \sum_z z_{ij} p(z_{ij}|(x,y,\Psi^{(k)})) = p(z_{ij}=1|x,y,\Psi^{(k)})$$

$$= \frac{\pi_j^{(k)}\varphi(y_i - x_i^T\beta_j^{(k)}; 0, \sigma^{2(k)})}{\sum_{j=1}^{m}\pi_j^{(k)}\varphi(y_i - x_i^T\beta_j^{(k)}; 0, \sigma^{2(k)})}$$

Maximize $Q(\Psi|\Psi^{(k)})$ respect to $\pi_j^{(k+1)}, \beta_j^{(k+1)}, \sigma^{2(k+1)}$:
$\pi_j^{(k+1)}$:

$$\frac{\partial Q(\Psi|\Psi^{(k)})}{\partial \pi_j} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}}{\pi_j} = 0$$

So $\pi_j^{(k+1)} = \frac{\sum_{i=1}^{n} p_{ij}^{(k+1)}}{n}$.
$\beta_j^{(k+1)}$:

$$log\varphi(y_i - x_i^T\beta_j; 0|\sigma^2) = -\frac{1}{2}log(2\pi\sigma^2) - \frac{(y_i - x_i^T\beta_j)^2}{2\sigma^2}$$

substitute above equation to $Q(\Psi|\Psi^{(k)})$.

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

$$\frac{\partial Q(\Psi|\Psi^{(k)})}{\partial \beta_j} = 0$$

gives that $\beta_j^{(k+1)} = (\sum_{i=1}^{n} x_i x_i^T p_{ij}^{(k+1)})^{-1}(\sum_{i=1}^{n} x_i p_{ij}^{(k+1)} y_i)$.

$$\frac{\partial Q(\Psi|\Psi^{(k)})}{\partial \sigma^2} = 0$$

gives that $\sigma^{2^{(k+1)}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}(y_i - \mathbf{x_i}^T\beta_j^{(k+1)})^2}{n}$.

Implement algorithm

```
knitr::opts_chunk$set(echo = TRUE)
require(data.table)
require(knitr)
require(pracma) #NR
require(ggplot2)

regmix_em <- function(y, xmat, pi.init, beta.init, sigma.init, control = list(maxiter = 100, to
  tol <- control$tol
  maxiter <- control$maxiter

  xmat <- as.matrix(xmat)
  n <- nrow(xmat)
  m <- length(pi.init)
  p <- ncol(xmat)
  pi <- pi.init
  beta <- beta.init
  sigma <- sigma.init
  pij <- matrix(0, nrow = n, ncol = m)
  pi.new <- rep(0, m)
  beta.new <- matrix(0, nrow = p, ncol = m)
  converge <- 1

  iter <- 1

  while ((converge > tol) && (iter < maxiter)){

    for (i in 1:n){
      ##print(length(xmat[i, ]))
      ##print(length(beta))
     pij[i, ] <- (pi * dnorm (y[i] - xmat[i, ] %*% beta, mean = 0, sd = sigma))/(sum(pi * dnorm
    }
    pi.new <- colMeans(pij)
    for (j in 1:m){
      beta.new[, j] <- solve(t(xmat) %*% diag (pij[, j])%*% xmat) %*% t(xmat) %*% diag (pij[ ,j
    }

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

    converge <- sum(abs(pi.new - pi)) + sum(abs(beta.new - beta))+ abs (sigma.new -sigma)

    iter <- iter + 1
    #if(iter >= maxiter) break
    pi <- pi.new
    beta <- beta.new
```

```
    sigma <- sigma.new
  }
  return(list(pi = pi.new, beta = beta.new, sigma = sigma.new, convergence = converge))
}
```

generate data

```
knitr::opts_chunk$set(echo = TRUE)
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)

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

```
## $pi
## [1] 0.3858259 0.2687730 0.3454011
##
## $beta
##           [,1]      [,2]       [,3]
## [1,] 0.8796634  0.991205 -0.9136811
## [2,] 0.9341892 -1.242468 -1.1990374
##
## $sigma
## [1] 1.023598
##
## $convergence
```

```
## [1] 8.74185e-06
```

Reference

https://stackoverflow.com/questions/19079152/contour-plot-of-a-custom-function-in-r [jun-yan/stat-5361]https://github.com/jun-yan/stat-5361