HW 3 - EM

Linqing(Waverly) Wei

October 23, 2018

```r
set.seed(20193847)

# simulate missing variables
i1 <- rbinom(5000,1, 0.6)
# simulate from P0
v1 <- rpois(5000,4) [i1]
# simulate from P1
v2 <- rpois(5000,5) [(1-i1)]
# get obseruvsed data
v <- c(v1,v2)
# tabulate data
t <- table(v)
# For each X=k, get the total number of X=k out of X1..Xn
df <- data.frame(rbind(t))
names(df) <- unique(sort(v))

# set starting values
mu0 <- 3
mu1 <- 4
p <- 0.65
# set the number of observations
n <- 5000
# set the stopping criteria
diff <- 1

# EM algorithm
# I use p for "pi" just for convenience
while(diff > 0.00001){
  # get k (for Xi = k )
  k <- as.numeric(names(df))
  # set empty vectors to collect parameters
  p_vec <- c()
  n_vec <- c()

  # E step
  # iterate through all k's from possion process
  for (i in 1: length(k)){
    # compute p when Xi=k using initial value p
    p_k <- (p * ((mu0)^k[i])/factorial(k[i])*exp(-mu0)) /
      (p * ((mu0)^k[i] / factorial(k[i])) *exp(-mu0) +
         (1-p) * (mu1^k[i] /factorial(k[i])) * exp(-mu1))
    # get the total counts of k's
    nk <- df[1,i]
    mk <- nk * p_k
    # collect p's
    p_vec <- c(p_vec,p_k)
    n_vec <- c(n_vec,nk)
  }


  # M step
  # get new p's
  p_new <- sum(n_vec*p_vec)/n
  # get new mu0
  mu0_new <- sum(k*n_vec*p_vec)/sum(n_vec * p_vec)
  # get new mu1
  mu1_new <- sum(k*n_vec*(1-p_vec)) / sum(n_vec * (1-p_vec))

  # compute the difference between old and new mu
  diff <- abs(mu0-mu0_new)

  # update p
  p <- max(p,p_new)
  # update mu0
  mu0 <- max(mu0,mu0_new)
  # update mu1
  mu1 <- max(mu1,mu1_new)
}

cat("true mu0 is 4", '\n')

## true mu0 is 4

cat("true mu1is 5", '\n')

## true mu1is 5

cat("mu0 is", mu0, '\n')

## mu0 is 3.601484

cat("mu1 is", mu1)

## mu1 is 4
```