

5241

Yi Chen(yc3356)

April 15, 2018

Homework 05

```
H<-matrix(readBin("histograms.bin", "double", 640000), 40000, 16)
dim(H)
```

```
## [1] 40000    16
```

first write the function of Multinomial EM

```
MultinomialEM <- function(H, K, tau) {

  init = sample(1:nrow(H), size = K, replace = F)
  centroid = H[init, ]
  centroid = ifelse(centroid == 0, 0.01, centroid)

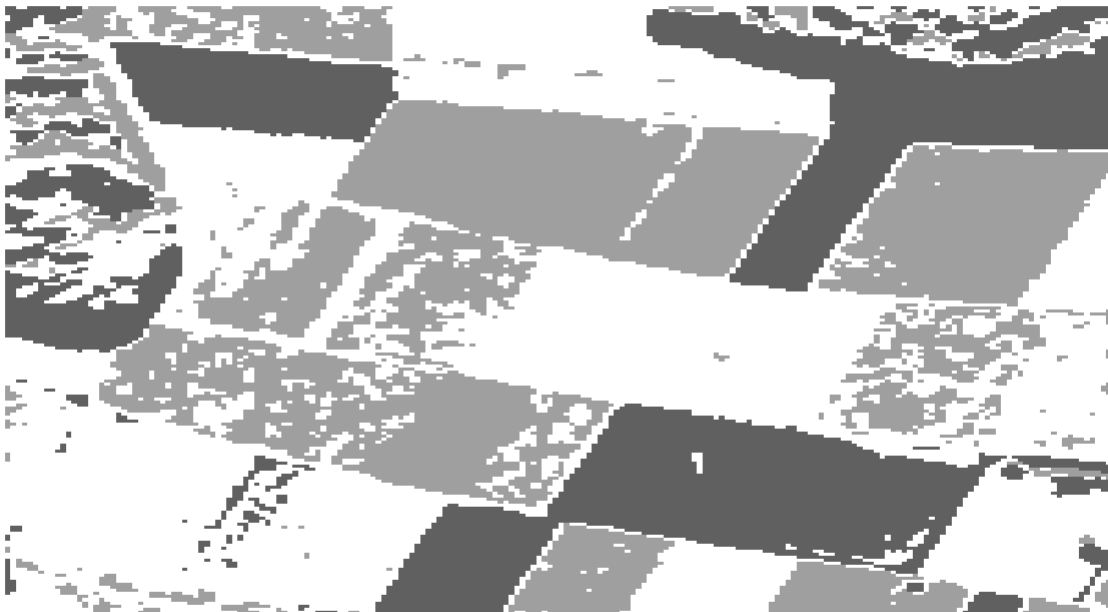
  delta = Inf
  A = matrix(0, nrow(H), K)
  phi = matrix(0, nrow(H), K)
  c = rep(1 / K, length.out = K)
  b = matrix(0, K, ncol(H))
  m = rep(0, length.out = nrow(H))
  while (delta >= tau) {
    A_old = A

    ## E-step
    phi = exp(H %*% t(log(centroid)))
    A = t(c * t(phi)) / rowSums(t(c * t(phi)))

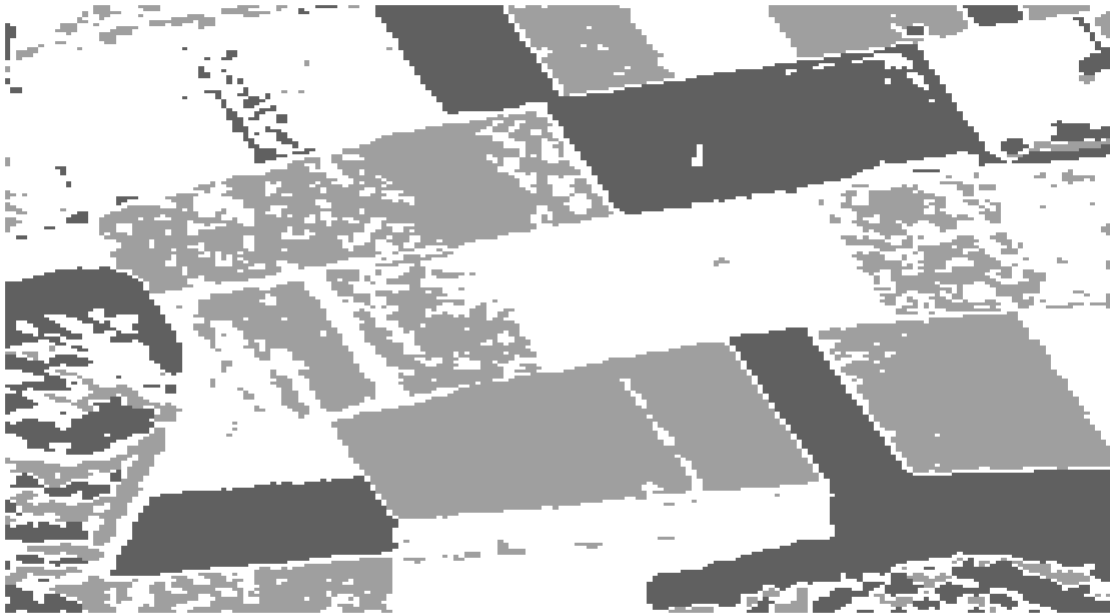
    ## M-step
    c = colSums(A)
    b = t(A) %*% H
    centroid = b / rowSums(b)
    ## Measurement of iteration
    delta = norm(A - A_old, "0")
  }
  ## Reparameterization
  m <- apply(A, 1, FUN = which.max)
  return(m)
}
```

try different parameter

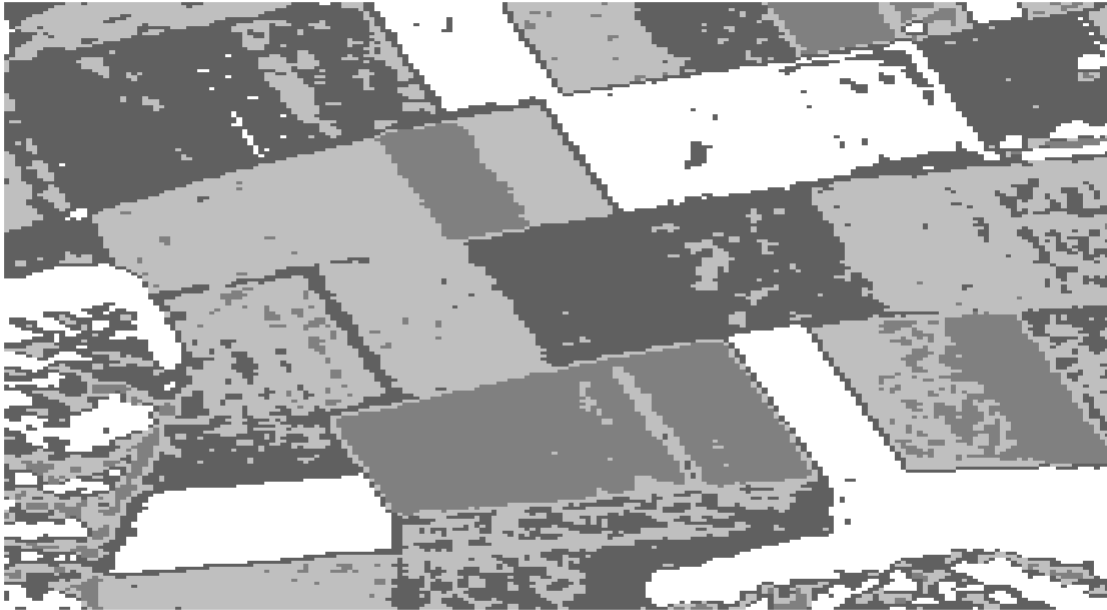
```
set.seed(1)
em_1 <- MultinomialEM(H, 3, 0.1)
em_2 <- MultinomialEM(H, 4, 0.1)
em_3 <- MultinomialEM(H, 5, 0.1)
# Reset the histograms into their original "position" in the image, then visualize the result matrix in grayscale to represent the clustering
hist_1 <- matrix(em_1, 200, 200)
hist_2 <- matrix(em_2, 200, 200)
hist_3 <- matrix(em_3, 200, 200)
image(hist_1, col = gray((3:8) / 8), axes = FALSE)
```



```
image(hist_1[, ncol(hist_1):1], col = gray((3:8) / 8), axes = FALSE)
```



```
image(hist_2[, ncol(hist_2):1], col = gray((3:8) / 8), axes = FALSE)
```



```
image(hist_3[, ncol(hist_3):1], col = gray((3:8) / 8), axes = FALSE)
```

