5241 HW5 Problem3

Mengjia Huang, UNI:mh3781

2018/4/23

0.Data Loading

```
H<-matrix(readBin("histograms.bin", "double", 640000), 40000, 16)
#The histograms were drawn at the nodes of a 4-by-4 pixel grid. Since the image has 8
00 × 800 pixels, there are 200 × 200 = 40000 histograms.
dim(H) #Each row is a histogram with 16 bins
```

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

```
sum(diff(rowSums(H))==0)==39999 #Each was drawn within a rectangle of edge length 11
 pixels, so check whether each histogram contains 11 × 11 = 121 values.
```

```
## [1] TRUE
```

1.Implement the EM algorithm

```
#inputs: the matrix of input histograms H
        #the number of clusters K
        #the threshold parameter τ
#output: a vector m of hard assignments

MultinomialEM<-function(H,K,tau){
  p <- ncol(H) # Number of bins per histogram
  n <- nrow(H) # Number of histograms
  H[H==0] <- 0.01 #may encounter numerical problems due to empty histogram bins. If s
o, add a small constant (such as 0.01) to the input histograms.

  #initialize: choose random centroids and normalize each
  centroids <- sample(c(1:n), size=K)
  t <- t(apply(H[centroids,], 1, function(row){row/sum(row)})) #These are Rd vectors
 (just like the input features).

  #E-steps
  a <- matrix(0, ncol=K, nrow=n) # a:assignment probabilities
  p <- matrix(0, ncol=K, nrow=n) # p:partial elements
  #M-steps
  ck <- matrix(1.0/K, ncol=1, nrow=K) # ck:mixture weights
  b <- matrix(0, ncol=K, nrow=1)

  m <- matrix(1.0/K, ncol=1, nrow=n) # m: hard assignments
  delta <- 1000 # inital delta(change of assignments)

  while(delta > tau){ #Terminate the iteration when δ < τ.
    apre <- a
    for(k in 1:K){
      # E-step:
      for(i in 1:n) {
        p[i, k] = exp(sum(H[i,] * log(t[k,])))
        a[i, k] = ck[k] * p[i, k] / sum(ck * p[i,])
      }
      a[is.nan(a)] = 0

      # M-step:
      ck[k] = sum(a[,k]) / n
      b = a[,k] %*% H
      t[k,] = b / sum(b)
    }
    #a measure of the change of assignments during the current iteration
    delta = norm(a - apre, "O")
  }
  #Turn the soft assignments into a vector m of hard assignments
  for( i in 1:n){
    m[i,1] <- which.max(a[i,])
  }
  return(m)
}
```

2.Train the algorithm

```
tau1 <- 0.1
K <- c(3, 4, 5)
M1 <- data.frame(matrix(NA, nrow = 40000, ncol = 0))

for(k in K){
    result <- MultinomialEM(H, k, tau1)
    result <- as.matrix(result)
    M1 <- cbind(M1, result)
  }

colnames(M1) <- c("K=3,τ=0.1", "K=4,τ=0.1", "K=5,τ=0.1")
dim(M1)
```

```
## [1] 40000     3
```

```
head(M1)
```

```
##   K=3,τ=0.1 K=4,τ=0.1 K=5,τ=0.1
## 1         1         4         4
## 2         1         4         4
## 3         1         4         4
## 4         2         4         4
## 5         1         4         4
## 6         1         4         4
```

```
tau2<-0.005
M2 <- data.frame(matrix(NA, nrow = 40000, ncol = 0))

for(k in K){
    result <- MultinomialEM(H, k, tau2)
    result <- as.matrix(result)
    M2 <- cbind(M2, result)
  }

colnames(M2) <- c("K=3,τ=0.005", "K=4,τ=0.005", "K=5,τ=0.005")
dim(M2)
```

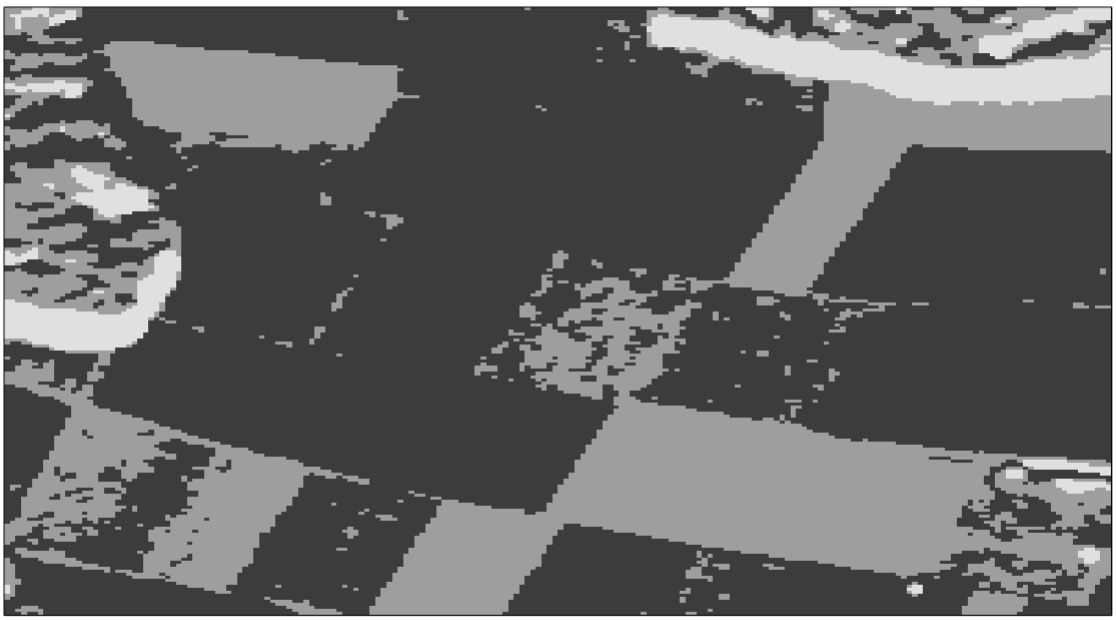
```
## [1] 40000     3
```

```
head(M2)
```

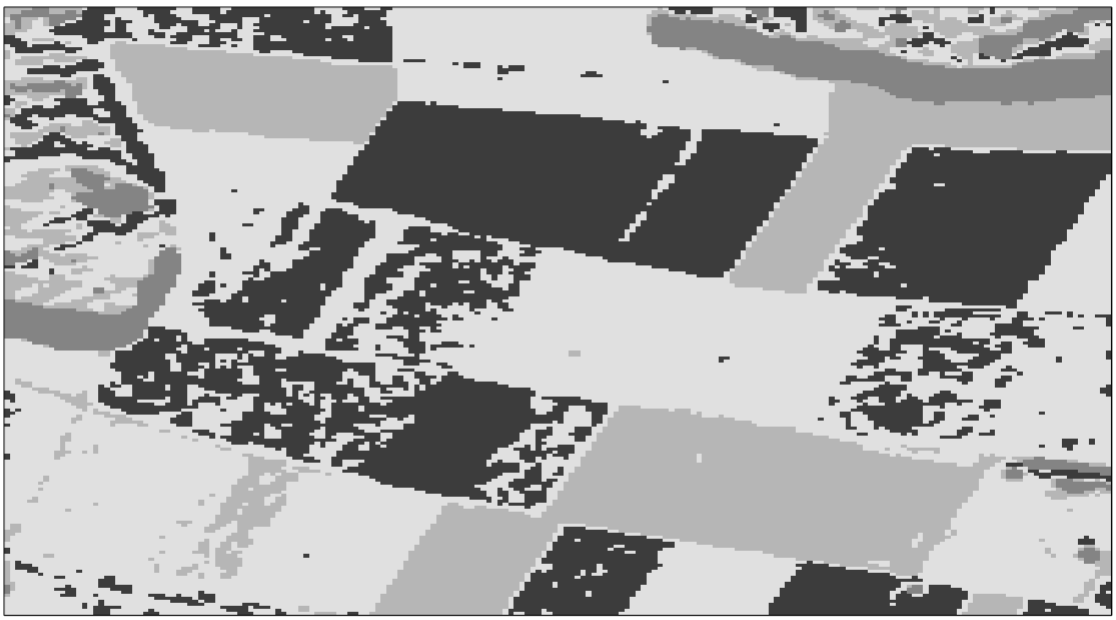
```
##   K=3,τ=0.005 K=4,τ=0.005 K=5,τ=0.005
## 1           1           2           3
## 2           1           2           3
## 3           1           2           3
## 4           1           2           3
## 5           1           2           3
## 6           1           2           3
```

3.Visualize the results

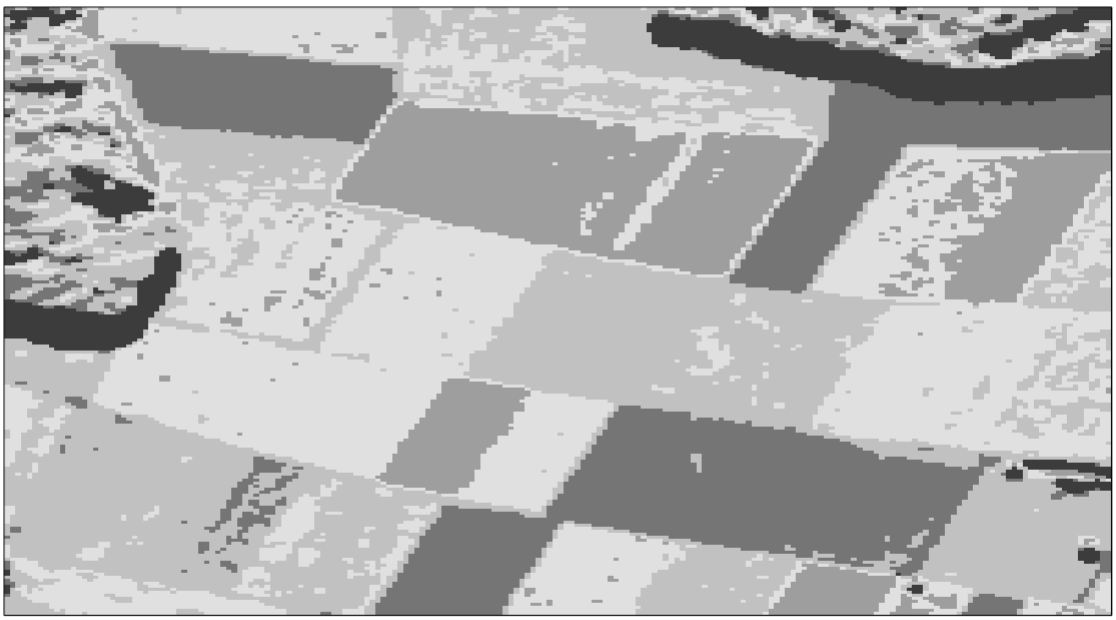
```
m_matrix1 <- matrix(M1[,1], nrow=200, ncol=200)
image(m_matrix1, col=gray.colors(3), xaxt="n", yaxt="n", main = "K=3 Clustering with
 τ=0.1")
```



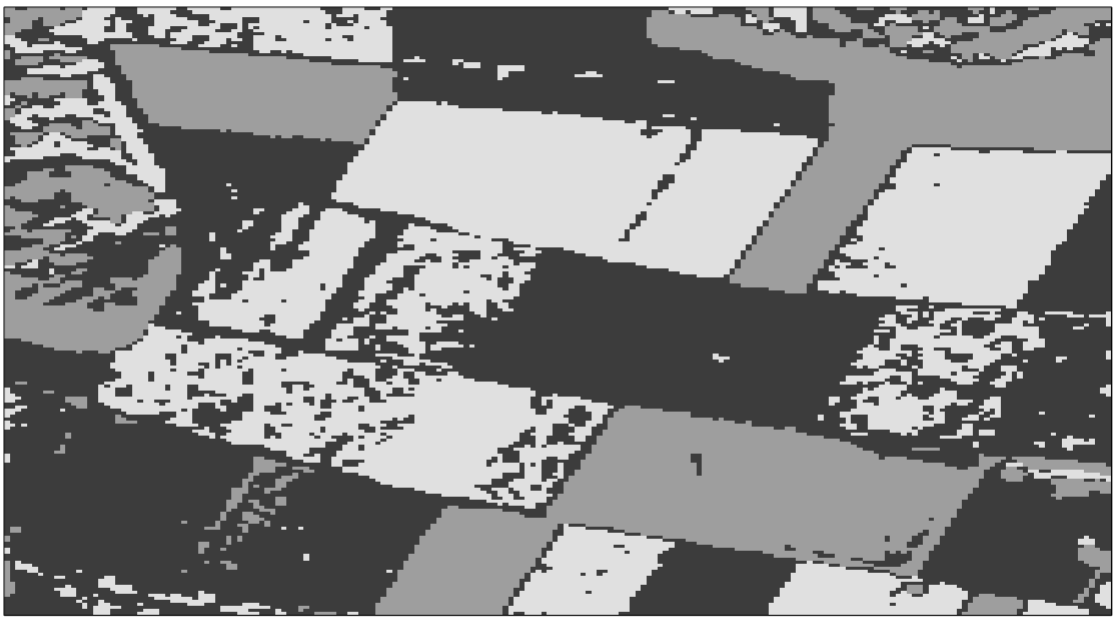

```
m_matrix2 <- matrix(M1[,2], nrow=200, ncol=200)
image(m_matrix2, col=gray.colors(4), xaxt="n", yaxt="n", main = "K=4 Clustering with
 τ=0.1")
```



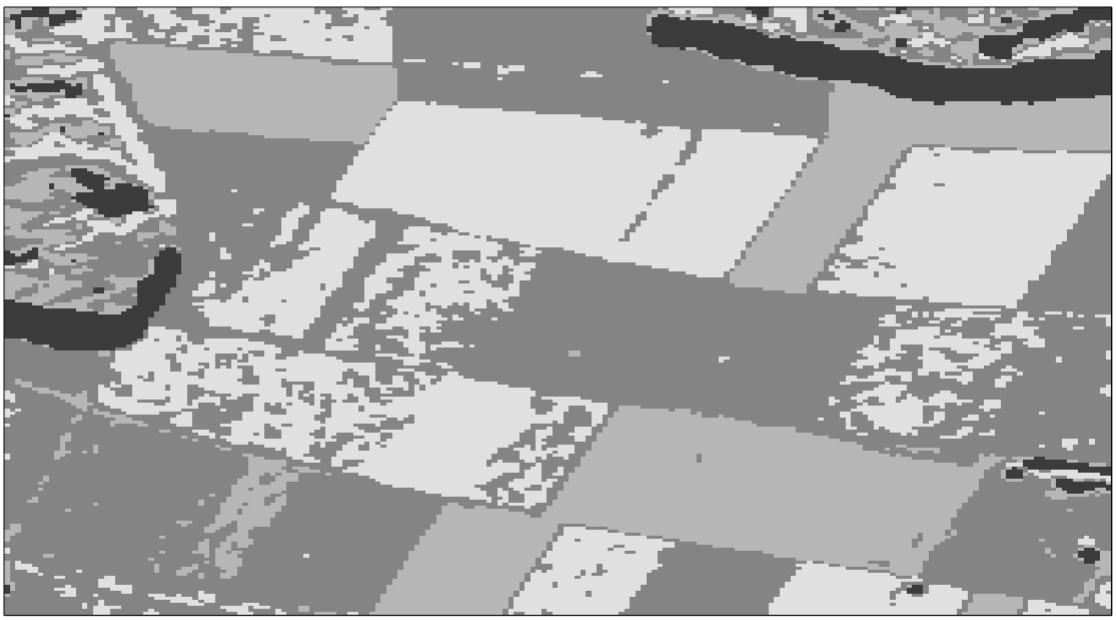

```
m_matrix3 <- matrix(M1[,3], nrow=200, ncol=200)
image(m_matrix3, col=gray.colors(5), xaxt="n", yaxt="n", main = "K=5 Clustering with
 τ=0.1")
```




```
m_matrix1 <- matrix(M2[,1], nrow=200, ncol=200)
image(m_matrix1, col=gray.colors(3), xaxt="n", yaxt="n", main = "K=3 Clustering with
 τ=0.005")
```




```
m_matrix2 <- matrix(M2[,2], nrow=200, ncol=200)
image(m_matrix2, col=gray.colors(4), xaxt="n", yaxt="n", main = "K=4 Clustering with
 τ=0.005")
```




```
m_matrix3 <- matrix(M2[,3], nrow=200, ncol=200)
image(m_matrix3, col=gray.colors(5), xaxt="n", yaxt="n", main = "K=5 Clustering with
 τ=0.005")
```

K=5 Clustering with τ=0.005