Assignment 7

The Major Miners

5 November 2017

Question 1

Importing the dataset

```
optdigits<-read.csv('optdigits.csv',header=TRUE)
```

Part a

Setting seed to 10 and performing k means clustering

```
set.seed(10)
fit<-kmeans(optdigits[1:64],10,iter.max=200)
str(fit)
```

```
## List of 9
##  $ cluster     : int [1:3823] 6 6 4 1 7 2 1 5 6 10 ...
##  $ centers     : num [1:10, 1:64] 0 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:10] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:64] "feature1" "feature2" "feature3" "feature4" ...
##  $ totss       : num 4602967
##  $ withinss    : num [1:10] 208108 217715 183942 318048 210258 ...
##  $ tot.withinss: num 2478719
##  $ betweenss   : num 2124248
##  $ size        : int [1:10] 263 349 297 441 305 373 385 308 719 383
##  $ iter        : int 4
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
#Matrix that records the number of instances of digits in each cluster
#Rows denote the cluster number
#Columns denote the digits
#fit$cluster has the cluster that each row belongs to
k<-matrix(nrow=10,ncol=10,0)
for(i in 1:length(fit$cluster))
{
     k[fit$cluster[i],optdigits$digit[i]+1]<-k[fit$cluster[i],optdigits$digit[i]+1]+1
     #optdigits$digit[i]+1 as it is indexed from 1 and digits start from 0
}
#The digits are from 0-9
colnames(k)<-c(0:9) #c(c())?
d<-vector()
#Labelling each cluster with the digit which has the maximum number of instances in it
for(i in 1:nrow(k))
{
  d[i]<-which.max(k[i,])-1
```

```
}
rownames(k)<-d
print(rownames(k))
```

```
##  [1] "1" "2" "1" "7" "5" "0" "6" "4" "3" "8"
```

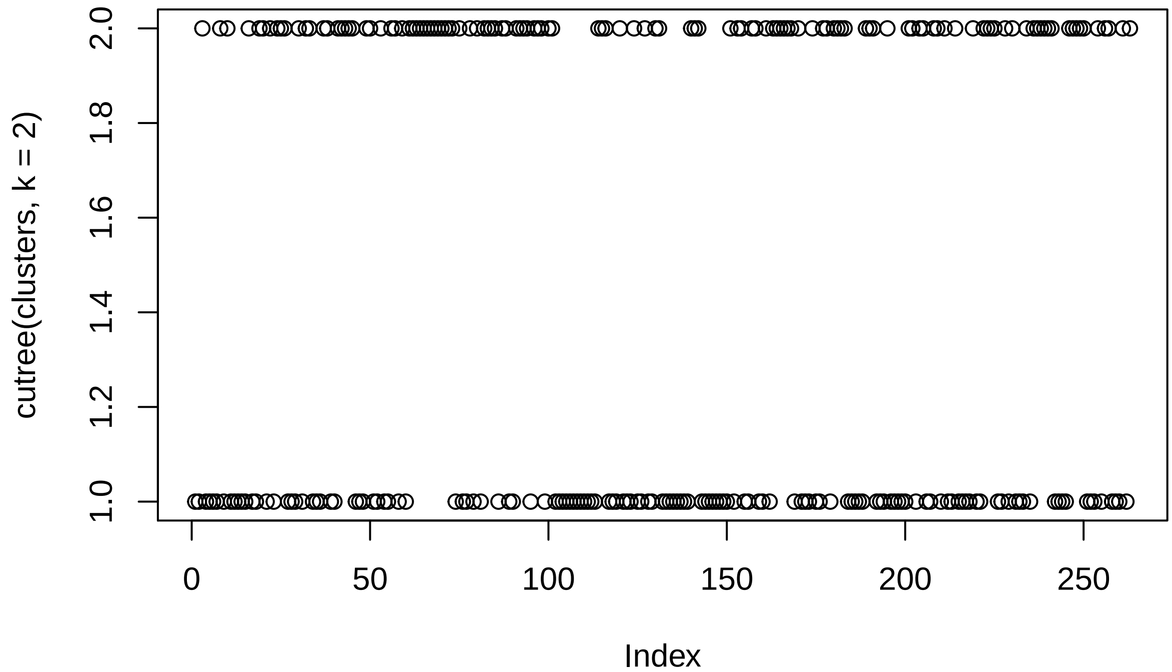
```
print(k)
```

```
##      0   1   2   3   4   5   6   7   8   9
## 1    1 113   0   5  30   6   0   6   5  97
## 2    0  15 329   5   0   0   0   0   0   0
## 1    0 250   0   2   6   0   3   5  29   2
## 7    0   0   4  10  29   0   0 373   1  24
## 5    0   1   0   4   7 289   0   0   3   1
## 0  373   0   0   0   0   0   0   0   0   0
## 6    1   1   1   0   4   1 373   0   4   0
## 4    1   0   0   0 306   0   1   0   0   0
## 3    0   9  19 346   0  80   0   0   9 256
## 8    0   0  27  17   5   0   0   3 329   2
```

Part b

```
#Part 2
#the first cluster has 113 1s and 97 9s. It's close.
#The cluster is cluster 1
count<-1
for(i in 1:length(fit$cluster))
{
  if(fit$cluster[i]==1)
  {
     d[count]<-i
     count<-count+1
  }
}
#New matrix containing only the rows that got clustered into cluster 1
newopt<-optdigits[d,]
#Hierarchical Clustering
clusters<-hclust(dist(newopt[1:64]))

plot(cutree(clusters, k = 2)) #can choose number of branches or cut height
```




```r
memships = cutree(clusters, k = 2)
cluster1 = subset(newopt, memships==1)
cluster2 = subset(newopt, memships==2)
table(cluster1$digit)
```

```
##
##  0  1  3  4  5  7  8  9
##  1  2  5 25  6  2  2 96
```

```r
table(cluster2$digit)
```

```
##
##   1   4   7   8   9
## 111   5   4   3   1
```

Part c

```r
clusnum <- vector()
clusindex <- vector()
fit$centers[1,]
```

```
##     feature1     feature2     feature3     feature4     feature5
##  0.000000000  0.015209125  0.456273764  4.247148289 11.935361217
##     feature6     feature7     feature8     feature9    feature10
## 11.817490494  3.615969582  0.197718631  0.000000000  0.376425856
##    feature11    feature12    feature13    feature14    feature15
```

```
##   4.851711027 10.334600760 11.939163498 13.060836502  6.053231939
##     feature16    feature17    feature18    feature19    feature20
##   0.349809886  0.000000000  2.011406844  9.771863118 10.353612167
##     feature21    feature22    feature23    feature24    feature25
## 10.840304183 13.072243346  4.631178707  0.110266160  0.000000000
##     feature26    feature27    feature28    feature29    feature30
##   3.730038023 12.030418251 12.163498099 13.441064639 13.391634981
##     feature31    feature32    feature33    feature34    feature35
##   3.855513308  0.000000000  0.000000000  2.038022814  6.479087452
##     feature36    feature37    feature38    feature39    feature40
##   6.825095057 11.448669202 12.874524715  2.182509506  0.000000000
##     feature41    feature42    feature43    feature44    feature45
##   0.000000000  0.239543726  0.912547529  2.277566540 12.079847909
##     feature46    feature47    feature48    feature49    feature50
## 10.980988593  0.806083650  0.000000000  0.000000000  0.034220532
##     feature51    feature52    feature53    feature54    feature55
##   0.368821293  3.866920152 13.657794677  8.806083650  0.828897338
##     feature56    feature57    feature58    feature59    feature60
##   0.000000000  0.000000000  0.007604563  0.307984791  4.927756654
##     feature61    feature62    feature63    feature64
## 10.916349810  7.182509506  1.159695817  0.000000000
```

```r
#fit$centers[1,] is the set of centers for the first cluster. There are 10 clusters.
#Load test data

test<-read.csv('optdigits_test.csv',header=TRUE)
for(i in 1:nrow(test)){
  distance = .Machine$integer.max
  for(j in 1:10){ #there are 10 clusters
    if(dist(rbind(test[i,2:ncol(test)], fit$centers[j,])) < distance){
      distance = dist(rbind(test[i,2:ncol(test)], fit$centers[j,]))
      clusnum[i] = rownames(k)[j]
      clusindex[i] = j
    }
  }
}
#clusnum refers to the digit that matches the input
#imagenumber is the index of the image in the test data
print(clusnum)
```

```
##  [1] "3" "1" "0" "1" "2" "7" "4" "5" "6" "8" "3" "0" "1" "2" "3" "4" "5"
## [18] "6" "7" "8"
```

```r
imagenumber = c(1:20)
result = data.frame(imagenumber, clusnum, clusindex)
print(result)
```

```
##    imagenumber clusnum clusindex
## 1            1       3         9
## 2            2       1         3
## 3            3       0         6
## 4            4       1         1
## 5            5       2         2
## 6            6       7         4
## 7            7       4         8
```

```
## 8              8       5          5
## 9              9       6          7
## 10            10       8         10
## 11            11       3          9
## 12            12       0          6
## 13            13       1          1
## 14            14       2          2
## 15            15       3          9
## 16            16       4          8
## 17            17       5          5
## 18            18       6          7
## 19            19       7          4
## 20            20       8         10
```

Part d

```r
#Printing the number of data points present under each label
length(cluster1$digit) #139 numbers
```

```
## [1] 139
```

```r
length(cluster2$digit) #124 numbers
```

```
## [1] 124
```

```r
cluster1$clusternumber = seq(0,0,length = nrow(cluster1))
cluster2$clusternumber = seq(0,0,length = nrow(cluster2))
#cluster1 is mostly 9
#cluster2 is mostly 1
#add the cluster number and merge them
for(row in 1:nrow(cluster1)){
  cluster1[row,"clusternumber"] = 1;
}
for(row in 1:nrow(cluster2)){
  cluster2[row,"clusternumber"] = 2;
}

final = rbind(cluster1, cluster2)

#We observe that two images, the 4th and the 13th, were classified into cluster 1. They are the test da
testdata = test[c(4,13),] #the ones classified to clusindex 1
traindata = test[-c(4,13),] #the ones that weren't
test_labels = clusindex[c(4,13)]
train_labels = clusindex[-c(4,13)]
library(class)
knnpredicted<-knn(traindata,testdata,cl = train_labels,k=7,prob=TRUE)
table(knnpredicted)
```

```
## knnpredicted
##  2  3  4  5  6  7  8  9 10
##  0  0  0  0  0  0  0  2  0
```

Question 2

Importing the dataset and modifying it to make it suitable for computation

```
hwr<-read.csv('handwriting_recognition.csv',header=TRUE)
hwr<-hwr[rep(row.names(hwr),hwr$Freq),]
hwr<-hwr[,c(2:4)]
```

Association rules with default settings

```
default<-apriori(hwr,control=list(verbose=FALSE))
default_dt<-as.data.frame(data.table(lhs=labels(lhs(default)),rhs=labels(rhs(default)),quality(default)
default_dt<-default_dt[,c(1:5)]
print(default_dt)
```

```
##                                          lhs                           rhs
## 1                      {Profession=Engineer}                 {Gender=Male}
## 2                       {Profession=Teacher} {Recognition=Unrecognized}
## 3                        {Profession=Artist}                 {Gender=Male}
## 4 {Recognition=Recognized,Profession=Artist}                 {Gender=Male}
##     support confidence     lift
## 1 0.1237296  0.9572650 1.610026
## 2 0.1475917  0.9355742 1.528116
## 3 0.1822802  0.8842444 1.487213
## 4 0.1131242  0.8519135 1.432836
```

Association rules for the remaining parts

```
rules<-apriori(hwr,parameter = list(support=0.001, confidence=0.001),control=list(verbose=FALSE))
```

Subquestion 1

###{Artist,Female}=> Recognized

```
part1<-subset(rules, lhs %ain% c("Profession=Artist","Gender=Female") & rhs %ain% c("Recognition=Recogn
part1_dt<-as.data.frame(data.table(lhs=labels(lhs(part1)),rhs=labels(rhs(part1)),quality(part1)))
part1_dt<-part1_dt[,c(1:5)]
print(part1_dt)
```

```
##                                  lhs                      rhs    support
## 1 {Gender=Female,Profession=Artist} {Recognition=Recognized} 0.01966416
##   confidence     lift
## 1  0.8240741 2.125219
```

Subquestion 2

{Engineer}=>Male

```
part2<-subset(rules,lhs %ain% c("Profession=Engineer") & rhs %ain% c("Gender=Male"))
part2<-part2[1]
part2_dt<-as.data.frame(data.table(lhs=labels(lhs(part2)),rhs=labels(rhs(part2)),quality(part2)))
part2_dt<-part2_dt[,c(1:5)]
print(part2_dt)
```

```
##                     lhs           rhs   support confidence     lift
## 1 {Profession=Engineer} {Gender=Male} 0.1237296   0.957265 1.610026
```

Subquestion 3

{Actor,Recognized} => Female

```
part3<-subset(rules,lhs %ain% c("Profession=Actor","Recognition=Recognized") & rhs %ain% c("Gender=Femal
part3_dt<-as.data.frame(data.table(lhs=labels(lhs(part3)),rhs=labels(rhs(part3)),quality(part3)))
part3_dt<-part3_dt[,c(1:5)]
print(part3_dt)
```

```
##                                         lhs             rhs    support
## 1 {Recognition=Recognized,Profession=Actor} {Gender=Female} 0.04463102
##   confidence     lift
## 1  0.6273292 1.547298
```

Subquestion 4

{Doctor,Male} => Unrecognized

```
part4<-subset(rules,lhs %ain% c("Profession=Doctor","Gender=Male") & rhs %ain% c("Recognition=Unrecogni
part4_dt<-as.data.frame(data.table(lhs=labels(lhs(part4)),rhs=labels(rhs(part4)),quality(part4)))
part4_dt<-part4_dt[,c(1:5)]
print(part4_dt)
```

```
##                               lhs                        rhs   support
## 1 {Gender=Male,Profession=Doctor} {Recognition=Unrecognized} 0.0304905
##   confidence     lift
## 1  0.7225131 1.180113
```