Clustering

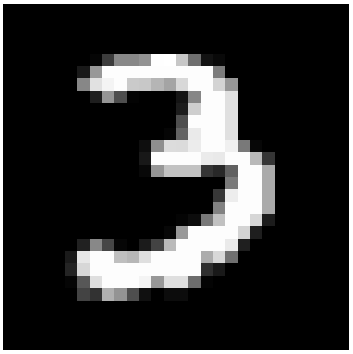
Load the MNIST digit recognition dataset into R from Kaggle https://www.kaggle.com/c/digit-recognizer/data

Each image is 28 by 28 pixels for a total of d = 784 pixels. Each pixel value is an integer between 0 and 255. We make use of the training dataset only, called 'train.csv', which has 785 columns, the first column being the label of the image: the true identity of the digit drawn by the user

```
data <- read.csv("train.csv")
head(names(data))
```

```
## [1] "label"  "pixel0" "pixel1" "pixel2" "pixel3" "pixel4"
```

```
m <- matrix(unlist(data[10,-1]), nrow=28, byrow=TRUE)
par(mar=c(.5,.5,.5,.5), pty="s")
image(t(m)[,nrow(m):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
```



For clustering purposes, we select images corresponding to labels 2,3 and 4, and we run the clustering algorithm on it with $K = 3$

```
xdata <- data[data$label==c(2,3,4),]
xdata <- xdata[,-1]
xsubset <- xdata[seq(1,500),] # Consider a subset of the data only
```

1. K-means clustering

Consider `nstart` random starts of the algorithm

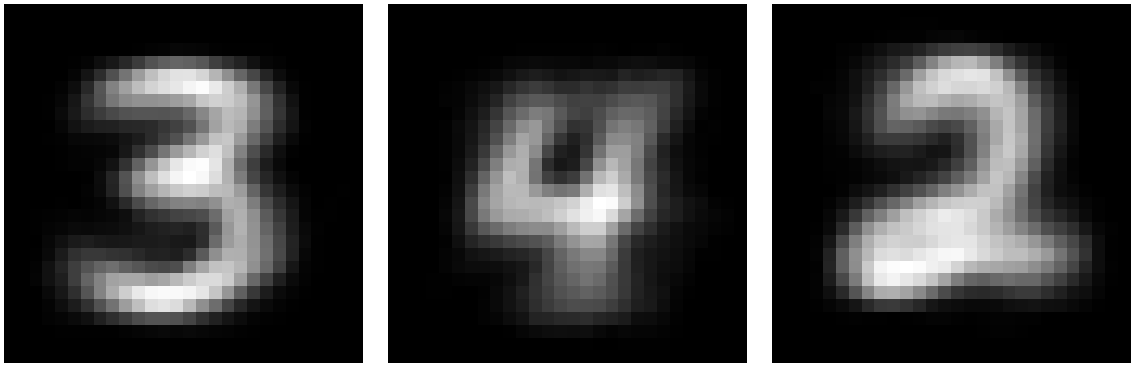
```
K=3
km.out <- kmeans(xsubset, centers=K, nstart=10)
summary(km.out)
```

```
##              Length Class  Mode
## cluster       500   -none- numeric
## centers      2352   -none- numeric
## totss           1   -none- numeric
## withinss        3   -none- numeric
```

```
## tot.withinss    1   -none- numeric
## betweenss       1   -none- numeric
## size            3   -none- numeric
## iter            1   -none- numeric
## ifault          1   -none- numeric
```

The centers are blurry (average) versions of the numbers 2,3 and 4

```
m1 <- km.out$centers[1,]; m1 <- matrix(m1, nrow=28, byrow=TRUE)
m2 <- km.out$centers[2,]; m2 <- matrix(m2, nrow=28, byrow=TRUE)
m3 <- km.out$centers[3,]; m3 <- matrix(m3, nrow=28, byrow=TRUE)
par(mfrow=c(1,3), mar=c(.5,.5,.5,.5), pty="s")
image(t(m1)[,nrow(m1):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
image(t(m2)[,nrow(m2):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
image(t(m3)[,nrow(m3):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
```

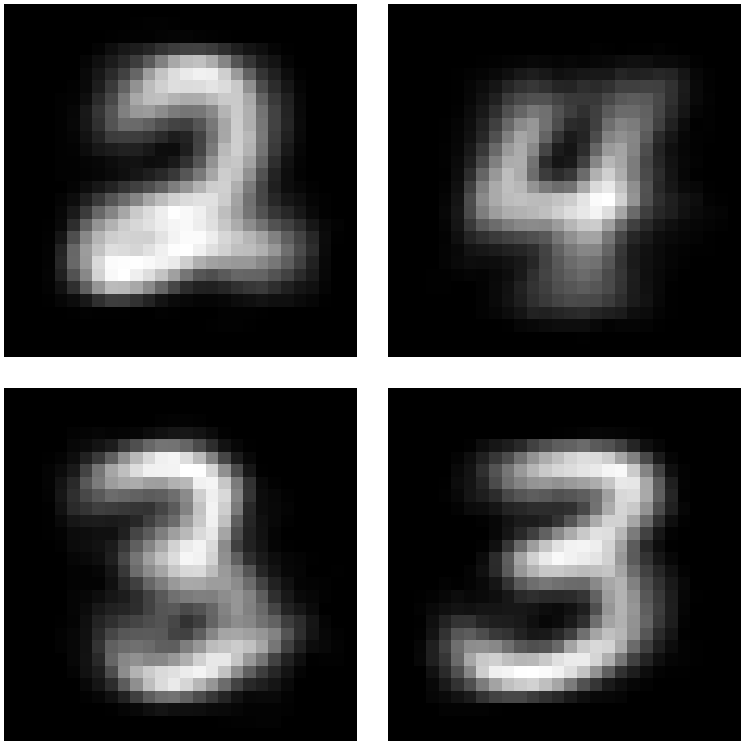


See what happens if you consider 4 clusters

```
K=4
km.out <- kmeans(xsubset, centers=K, nstart=10)
summary(km.out)
```

```
##              Length Class  Mode
## cluster       500   -none- numeric
## centers      3136   -none- numeric
## totss           1   -none- numeric
## withinss        4   -none- numeric
## tot.withinss    1   -none- numeric
## betweenss       1   -none- numeric
## size            4   -none- numeric
## iter            1   -none- numeric
## ifault          1   -none- numeric
```

```
m1 <- km.out$centers[1,]; m1 <- matrix(m1, nrow=28, byrow=TRUE)
m2 <- km.out$centers[2,]; m2 <- matrix(m2, nrow=28, byrow=TRUE)
m3 <- km.out$centers[3,]; m3 <- matrix(m3, nrow=28, byrow=TRUE)
m4 <- km.out$centers[4,]; m4 <- matrix(m4, nrow=28, byrow=TRUE)
par(mfrow=c(2,2), mar=c(.5,.5,.5,.5), pty="s")
image(t(m1)[,nrow(m1):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
image(t(m2)[,nrow(m2):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
image(t(m3)[,nrow(m3):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
image(t(m4)[,nrow(m4):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
```



The cluster corresponding to the number 3 is split into two clusters

2. Gaussian Mixture Model (GMM)

```
library("mclust")
```

```
## Warning: package 'mclust' was built under R version 3.1.3
```

```
## Package 'mclust' version 5.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
K=3
# EII : Equal Diagonal Covariance Matrices (Sigma_k = Lambda * Identity)
# VII : Unequal Diagonal Covariance Matrices (Sigma_k = Lambda_k * Identity)
gmm.out <- Mclust(xsubset, G=K, modelNames = "VII")
summary(gmm.out)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VII (spherical, varying volume) model with 3 components:
##
##  log.likelihood   n   df      BIC      ICL
##       -2166654 500 2357 -4347956 -4347956
##
## Clustering table:
```

```
##   1   2   3
## 165 156 179
```

```
gmm.out$parameters$pro # Mixing Proportions
```

```
## [1] 0.3301378 0.3119039 0.3579583
```

```
m1 <- gmm.out$parameters$mean[,1]; m1 <- matrix(m1, nrow=28, byrow=TRUE) # Mean vectors
m2 <- gmm.out$parameters$mean[,2]; m2 <- matrix(m2, nrow=28, byrow=TRUE)
m3 <- gmm.out$parameters$mean[,3]; m3 <- matrix(m3, nrow=28, byrow=TRUE)
C1 <- gmm.out$parameters$variance$sigma[,,1];  # Cov Matrices
C2 <- gmm.out$parameters$variance$sigma[,,2];
C3 <- gmm.out$parameters$variance$sigma[,,3];
par(mfrow=c(1,3), mar=c(.5,.5,.5,.5), pty="s")
image(t(m1)[,nrow(m1):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
image(t(m2)[,nrow(m2):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
image(t(m3)[,nrow(m3):1], axes = FALSE, col = grey(seq(0, 1, length = 256)))
```