Problem 8.1

Chen Bo Calvin Zhang

23/11/2020

This problem consist in classifying wine data.

```r
# load data set
library("whitening")
```

```
## Loading required package: corpcor
```

```r
data(forina1986)
wine.attrib = forina1986$attrib
wine.type = forina1986$type
print(dim(wine.attrib))
```

```
## [1] 178  27
```

```r
print(levels(wine.type))
```

```
## [1] "Barolo"     "Grignolino" "Barbera"
```

```r
print(table(wine.type))
```

```
## wine.type
##     Barolo Grignolino    Barbera
##         59         71         48
```

Now split the data in training and test sets of size {60, 90, 120, 150} and {118, 88, 58, 28} respectively.

```r
split_data = function(X, Y, idx)
{
  X.train = X[idx, ]
  Y.train = Y[idx]
  X.test = X[-idx, ]
  Y.test = Y[-idx]

  return (list(X.train=X.train, Y.train=Y.train, X.test=X.test, Y.test=Y.test))
}

split60 = split_data(wine.attrib, wine.type, sample(178, 60))
split90 = split_data(wine.attrib, wine.type, sample(178, 90))
split120 = split_data(wine.attrib, wine.type, sample(178, 120))
split150 = split_data(wine.attrib, wine.type, sample(178, 150))
```

Now, fit an LDA model and a QDA model using the four tranind data set and compute the number of missclassifed samples.

```r
library("MASS")

fit_missclass = function(data, method = "lda")
{
  if (method == "lda") {
    train.out = lda(data$X.train, data$Y.train)
  } else if (method == "qda") {
    train.out = qda(data$X.train, data$Y.train)
  }

  test.pred = predict(train.out, data$X.test)$class
  return (sum(test.pred != data$Y.test))
}

print(fit_missclass(split60))
```

```
## [1] 7
```

```r
print(fit_missclass(split90))
```

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

```r
print(fit_missclass(split120))
```

```
## [1] 0
```

```r
print(fit_missclass(split150))
```

```
## [1] 0
```

```r
# print(fit_missclass(split60, method = "qda"))  # throws and error because sample size is too small
# print(fit_missclass(split90, method = "qda"))  # throws and error because sample size is too small
print(fit_missclass(split120, method = "qda"))
```

```
## [1] 19
```

```r
print(fit_missclass(split150, method = "qda"))
```

```
## [1] 2
```

Let us write a predictor function for LDA and one for QDA.

```r
predfun.lda = function(train.x, train.y, test.x, test.y)
{
  lda.fit = lda(train.x, train.y)
  ynew = predict(lda.fit, test.x)$class
  # compute accuracy
  acc = mean(ynew == test.y)
  return(acc)
}

predfun.qda = function(train.x, train.y, test.x, test.y)
{
  qda.fit = qda(train.x, train.y)
  ynew = predict(qda.fit, test.x)$class
  # compute accuracy
  acc = mean(ynew == test.y)
  return(acc)
}
```

Now, perform cross validation with 5 folds and 50 repeats.

```r
library("crossval")

cv.out = crossval(predfun.lda, wine.attrib, wine.type, K=5, B=50, verbose=FALSE)
print(c(cv.out$stat, cv.out$stat.se))
```

```
## [1] 0.990702314 0.001010088
```

```r
cv.out = crossval(predfun.qda, wine.attrib, wine.type, K=5, B=50, verbose=FALSE)
print(c(cv.out$stat, cv.out$stat.se))
```

```
## [1] 0.929401146 0.002736926
```

Let us compare the results with cluster analysis using GMMs.

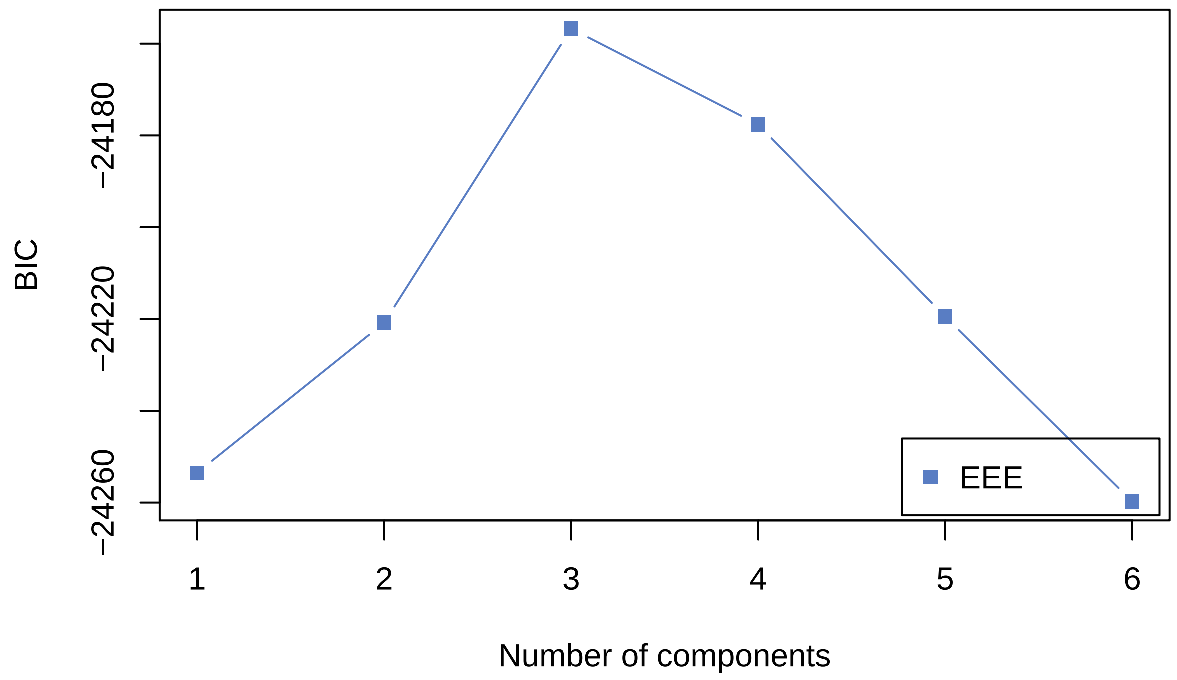
```r
library("mclust")
```

```
## Package 'mclust' version 5.4.6
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
gmm.out = Mclust(wine.attrib, G=1:6, verbose=FALSE)
print(gmm.out$G)  # the optimal number of groups
```

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

```r
plot(gmm.out, what="BIC", modelName="EEE")
```



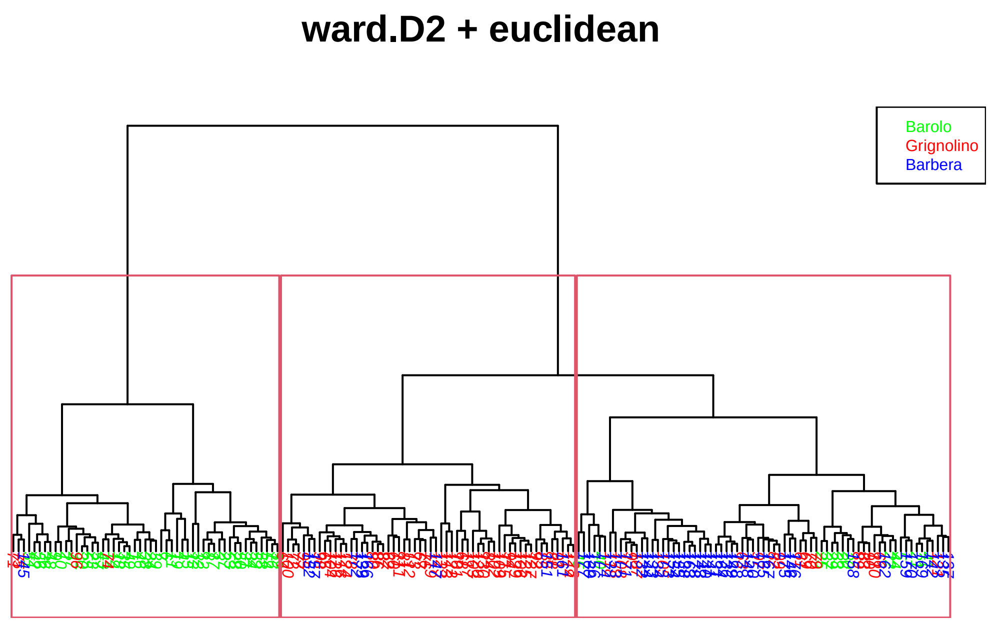
```
print(table(gmm.out$classification, wine.type))
```

```
##    wine.type
##     Barolo Grignolino Barbera
##   1     58          7       0
##   2      1         62       0
##   3      0          2      48
```

Finally, compute a hierarchical clustering with Ward's algorithm (ward.D2) on Euclidean distances, both for the original and standardised data.

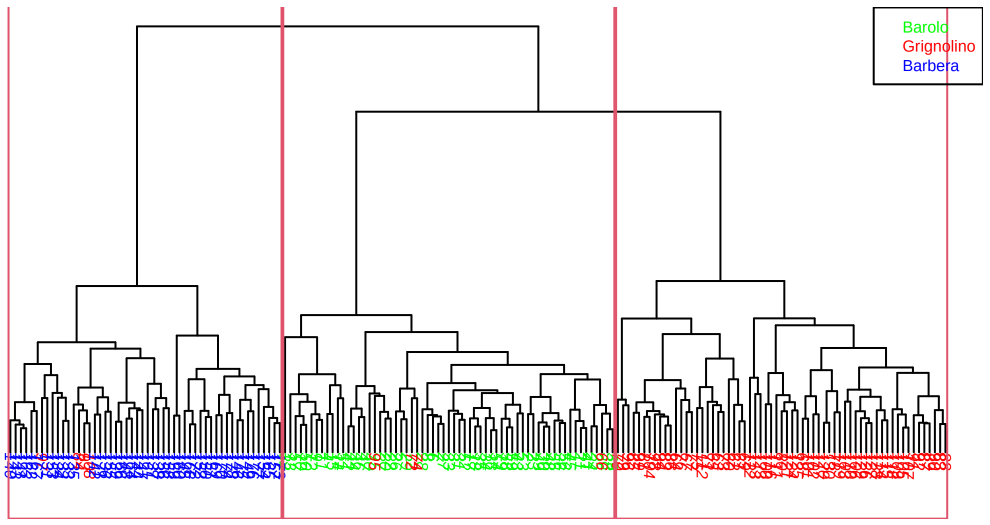
```
X = wine.attrib
X.std = scale(X, scale=TRUE)

dist = dist(X, method="euclidean")
dist.std = dist(X.std, method="euclidean")

hclust = hclust(dist, method = "ward.D2")
hclust.std = hclust(dist.std, method="ward.D2")

library("ape")

col.wine = c("green", "red", "blue")
colMap.wine = col.wine[as.integer(wine.type)]
```

```r
plot(as.phylo(hclust), tip.color = colMap.wine, cex = 0.5,
     direction = "downward", main=paste(hclust$method, "+", hclust$dist.method))
rect.hclust(hclust, k=3)
legend("topright", legend=levels(wine.type), text.col=col.wine, cex=0.5)
```



```r
plot(as.phylo(hclust.std), tip.color = colMap.wine, cex = 0.5,
     direction = "downward", main=paste(hclust.std$method, "+", hclust.std$dist.method))
rect.hclust(hclust.std, k=3)
legend("topright", legend=levels(wine.type), text.col=col.wine, cex=0.5)
```

ward.D2 + euclidean


Barolo
Grignolino
Barbera