**Stat 407 Lab 9 Supervised Classification Fall 2012**
**Due: Wednesday, Oct 24 2012 in class** Hand in one solution per group.

**Purpose:** This lab is about supervised classification. We'll look at the music data.

**Data:** There are 54 data points (tracks), 30 are Rock tracks and 24 are Classical tracks. (There are also 3 tracks from Enya, and 5 unknown tracks. You'll need to remove these in order to compute the summary statistics and build the classification model.) The variables are:

| | |
|---|---|
| LVar, LAve, LMax | the average, variance, maximum of the frequencies of the left channels |
| LFEner | an indicator of the amplitude or loudness of the sound. |
| LFreq | Average of the top 15 frequencies as computed by the `periodogram` function. |

1. Compute the means, and variance-covariances for each group.
2. Make a scatterplot matrix, showing the two groups with different symbols/colors.
3. Is it reasonable to assume equal variance-covariances between the two groups for this data? Explain.
4. Is it reasonable to assume that the data from each group is a sample from a multivariate normal distribution? Explain.
5. Standardize the variables, and compute the means and variance-covariances for each group again, and the pooled variance-covariance matrix. We will use the standardized data to build the LDA rule, basically because it will be easier to write down the rule, and determine the important variables.
6. Calculate the linear discriminant analysis. Write down the LDA rule, including the constant term. (It is of the form *"If* $a_1x_1 + a_2x_2 + ... + a_px_p - constant > 0$ *allocate new observation to group 1 otherwise allocate to group 2."*)
7. Which variable(s) is the most important, based on magnitude of the coefficients?
8. Write down the misclassification table and report the error rate.
9. Which track(s) are missclassified?
10. Classify the 5 unknown tracks.

**Hints:**

- Data is in the file `music-plusnew-sub.csv`:

```
music<-read.csv("data/music-plusnew-sub.csv",row.names=1)
```

- Summary statistics

```
# Subset data for building rule
music.sub<-subset(music,Type=="Classical"|Type=="Rock",
    select=Type:LFreq)
type_to_num<-c("Classical"=1,"Rock"=2)
music.sub[,1]<-factor(music.sub[,1],exclude="New wave")
```

- Group means

```
options(digits=3)
apply(music.sub[music.sub[,1]=="Classical",-1],2,mean,na.rm=T)
apply(music.sub[music.sub[,1]=="Rock",-1],2,mean,na.rm=T)
var(music.sub[music.sub[,1]=="Classical",-1],na.rm=T)
var(music.sub[music.sub[,1]=="Rock",-1],na.rm=T)
```

- Scatterplot matrix

```
# This way, ....
library(ggplot2)
plotmatrix(music.sub[,-1]) + geom_point(aes(colour=music.sub$Type))

# Or this way.
library(GGally)
ggpairs(music.sub, columns=2:ncol(music.sub), colour="Type")
```

- Standardize variables

```
music.sub[,-1]<-scale(music.sub[,-1])
```

- Pooled variance-covariance

```
v1<-var(music.sub[music.sub[,1]=="Classical",-1],na.rm=T)
v2<-var(music.sub[music.sub[,1]=="Rock",-1],na.rm=T)
table(music.sub[,1])
(23*v1+29*v2)/52
```

- LDA

```
# LDA
library(MASS)
music.lda<-lda(Type~.,data=music.sub,prior=c(0.5,0.5))
music.lda

music.lda$scaling # these are the linear coefficients
music.lda$means   # group means

# Calculate the constant term
-(music.lda$means[1,]+music.lda$means[2,])%*%music.lda$scaling/2
```

- Misclassification table

```
table(music.sub[,1],predict(music.lda,music.sub)$class)
```

- Which tracks are misclassified?

```
data.frame(music.sub$Type, pred.Type=predict(music.lda,music.sub)$class)
```

- Classify the unknown cases.

```
music.mn<-apply(music[1:54,-c(1,2)],2,mean)
music.sd<-apply(music[1:54,-c(1,2)],2,sd)
music.new<-music[58:62,-c(1,2)]
for (i in 1:5) {
  music.new[,i]<-(music.new[,i]-music.mn[i])/music.sd[i]
}
predict(music.lda,music.new)$class
```

- Plot the data in the discriminant space

```
music.lda.model<-data.frame(music.sub, LD1=predict(music.lda,music.sub)$x,
  predType=predict(music.lda,music.sub)$class,
  posterior=predict(music.lda,music.sub)$posterior)
qplot(LD1, data=music.lda.model, geom="histogram", binwidth=0.5)+ facet_wrap(~Type, ncol=1)
```