

## Solution to Series 7

1. a) Maximizing  $\hat{\pi}_j$  (see (6.4)) over  $j$  for fixed  $x$  is equivalent to the maximization of  $\log(\hat{f}_{X|Y=j}(x)p_j)$  (the denominator does not depend on  $j$ ).

The normal density is given by

$$\hat{f}_{X|Y=j}(x) = \frac{1}{(2\pi)^{p/2} \det(\hat{\Sigma}_j)^{1/2}} \exp\left(-\frac{1}{2}(x - \hat{\mu}_j)^\top \hat{\Sigma}_j^{-1}(x - \hat{\mu}_j)\right).$$

We get

$$\log(\hat{f}_{X|Y=j}(x)p_j) = -\frac{p}{2} \log(2\pi) - \log(\det(\hat{\Sigma}_j))/2 - (x - \hat{\mu}_j)^\top \hat{\Sigma}_j^{-1}(x - \hat{\mu}_j)/2 + \log(\hat{p}_j).$$

Ignoring the leading constant term gives the solution  $\hat{\delta}_j^{QDA}$ .

- b) We can directly start at the solution of a) and replace  $\hat{\Sigma}_j$  by  $\hat{\Sigma}$ :

$$\hat{\delta}_j^{pre}(x) = -\log(\det(\hat{\Sigma}))/2 - x^\top \hat{\Sigma}^{-1}x/2 + x^\top \hat{\Sigma}^{-1}\hat{\mu}_j - \hat{\mu}_j^\top \hat{\Sigma}^{-1}\hat{\mu}_j/2 + \log(\hat{p}_j)$$

Again, by ignoring the two leading constant terms, we get

$$\hat{\delta}_j^{LDA}(x) = (x - \frac{\hat{\mu}_j}{2})^\top \hat{\Sigma}^{-1}\hat{\mu}_j + \log(\hat{p}_j). \quad (1)$$

- c) The boundary between the two groups is the set  $B = \{x \mid \hat{\delta}_0(x) = \hat{\delta}_1(x)\}$ . We get

$$\begin{aligned} B &= \{x \mid x^\top b_0 + c_0 = x^\top b_1 + c_1\} \\ &= \{x \mid x^\top (b_0 - b_1) + c_0 - c_1 = 0\} \end{aligned} \quad (2)$$

which is a  $p - 1$  dimensional subspace of  $\mathbb{R}^p$ , e.g. a straight line in  $\mathbb{R}^2$  or a plane in  $\mathbb{R}^3$ .

$B_0 = \{x \mid \hat{\delta}_0(x) > \hat{\delta}_1(x)\}$  is the set of points which are assigned to class 0. We can write  $B_0 = \{x \mid x^\top (b_0 - b_1) + c_0 - c_1 > 0\}$ . This is the set of points  $x$  which lie on "one side" of the boundary  $B$ . From equation (1), we find

$$b_j = \hat{\Sigma}^{-1}\hat{\mu}_j, \quad (3)$$

$$c_j = \log \hat{p}_j - \frac{1}{2}\hat{\mu}_j^\top \hat{\Sigma}^{-1}\hat{\mu}_j = \log \hat{p}_j - \frac{1}{2}\hat{\mu}_j^\top b_j. \quad (4)$$

Hence the decision boundary (2) is

$$\begin{aligned} B &= \{x \mid x^\top \underbrace{\hat{\Sigma}^{-1}(\hat{\mu}_0 - \hat{\mu}_1)}_{=:w} + \log \frac{\hat{p}_0}{\hat{p}_1} - \underbrace{\frac{1}{2}\hat{\mu}_0^\top \hat{\Sigma}^{-1}\hat{\mu}_0 + \frac{1}{2}\hat{\mu}_1^\top \hat{\Sigma}^{-1}\hat{\mu}_1}_{=: -a}\} \\ &= \{x \mid x^\top w = a\}. \end{aligned}$$

If we define

$$z = \frac{1}{2}(\hat{\mu}_0 + \hat{\mu}_1) + \frac{\hat{\mu}_0 - \hat{\mu}_1}{(\hat{\mu}_0 - \hat{\mu}_1)^\top \hat{\Sigma}^{-1}(\hat{\mu}_0 - \hat{\mu}_1)} \log \frac{\hat{p}_1}{\hat{p}_0}$$

and write  $B = \{x \mid (x - z)^\top w = 0\}$ , one can show that the decision boundary shifts towards  $\hat{\mu}_0$  (and away from  $\hat{\mu}_1$ ) as the fraction of the estimated prior probabilities  $\hat{p}_1/\hat{p}_0$  increases, and vice versa. Moreover, if the covariance matrix  $\hat{\Sigma}$  is diagonal and has constant components, then  $w$  is parallel to the straight line connecting the means  $\hat{\mu}_0$  and  $\hat{\mu}_1$ , and the decision boundary  $B$  is orthogonal to it.

2. Preparations for the exercise: set the seed, read in the data, parameters and functions:

```

> ## Read in dataset, set seed, load package
> Iris <- iris[,c("Petal.Length", "Petal.Width", "Species")]
> grIris <- as.integer(Iris[, "Species"])
> set.seed(16)
> library(MASS)
> ## Read n
> n <- nrow(Iris)
> ## Utility function for plotting boundaries
> predplot <- function(object, x, gr = grIris, main = "", lines.only=FALSE,
                        len = 42, colcont = "black", ...)
{
  ## gr : the true grouping/class vector
  stopifnot(length(gr) == nrow(x))
  xp <- seq(min(x[,1]), max(x[,1]), length=len)
  yp <- seq(min(x[,2]), max(x[,2]), length=len)
  grid <- expand.grid(xp, yp)
  colnames(grid) <- colnames(x)[-3]
  Z <- predict(object, grid, ...)
  zp <- Z$post[,3] - pmax(Z$post[,2], Z$post[,1])
  if(!lines.only)
    plot(x[,1], x[,2], col = gr, pch = gr,
         main = main, xlab=colnames(x)[1], ylab=colnames(x)[2])
  contour(xp, yp, matrix(zp, len),
         add = TRUE, levels = 0, drawlabels = FALSE, col=colcont)
  zp <- Z$post[,1] - pmax(Z$post[,2], Z$post[,3])
  contour(xp, yp, matrix(zp, len),
         add = TRUE, levels = 0, drawlabels = FALSE, col=colcont)
}
> ## Bootstrap size
> B <- 1000

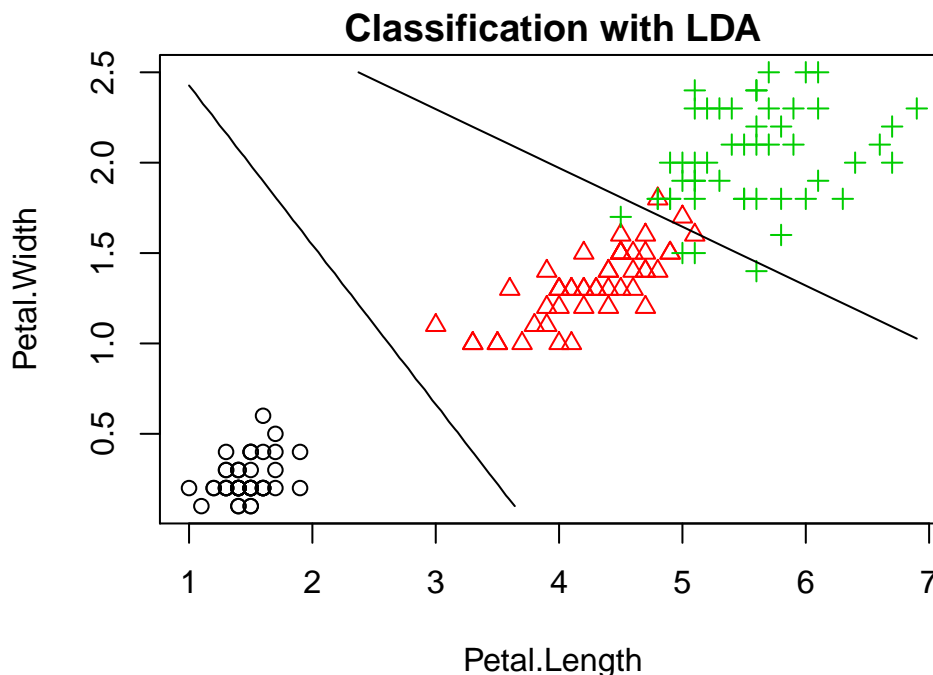
```

- a) Here we fit the data with the LDA and QDA methods and plot the resulting classifications. We begin with the LDA method.

```

> class_lda <- lda(x=Iris[,c("Petal.Length", "Petal.Width")], grouping=Iris[, "Species"])
> predplot(class_lda, x=Iris, main="Classification with LDA")

```

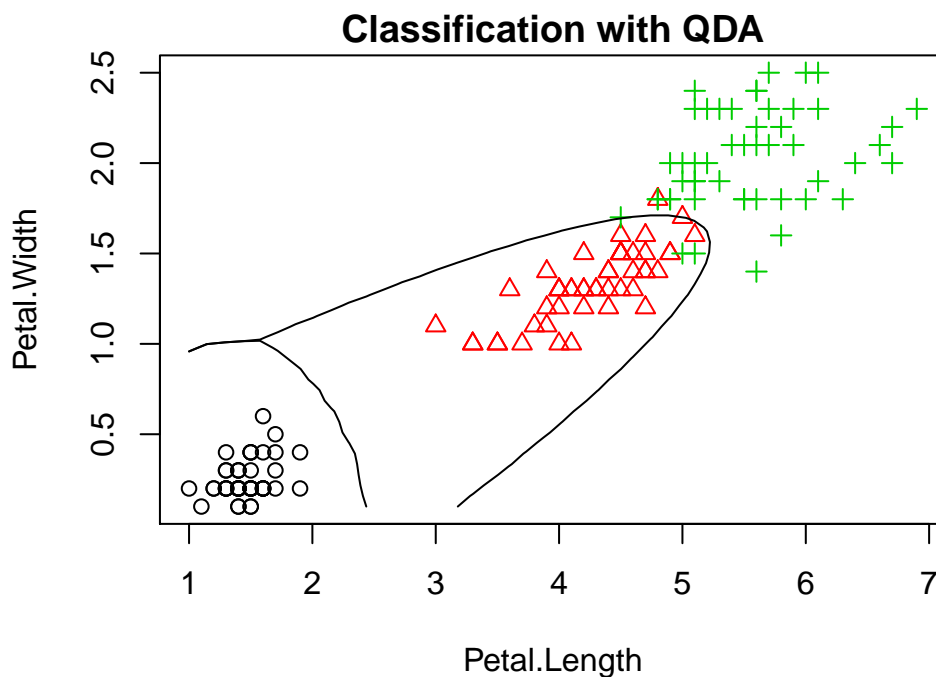


Now we have the QDA method.

```

> class_qda <- qda(x=Iris[,c("Petal.Length", "Petal.Width")], grouping=Iris[, "Species"])
> predplot(class_qda, Iris, main="Classification with QDA")

```

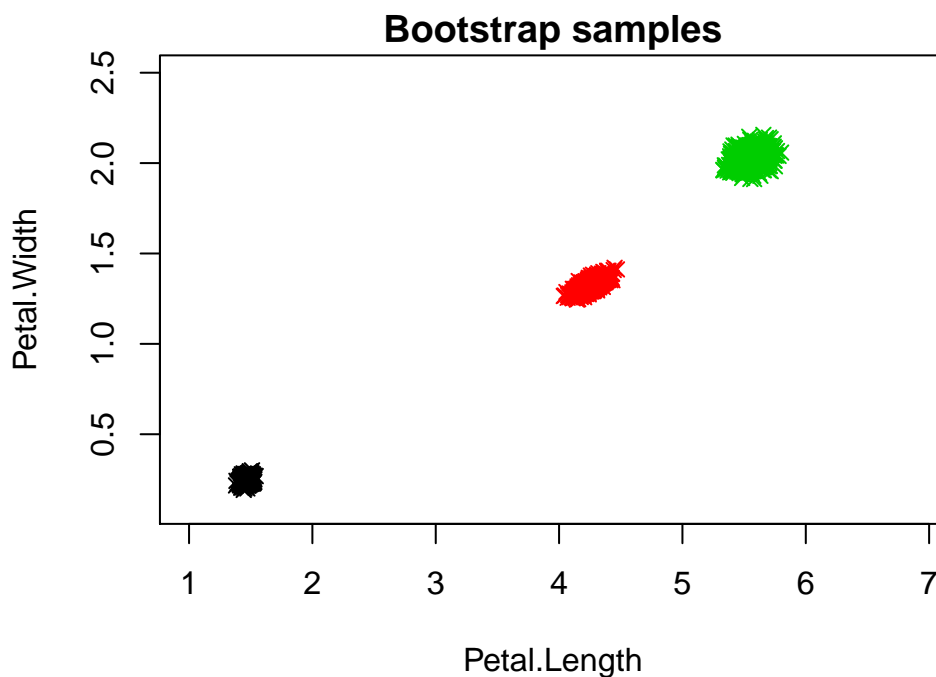


- b) We begin by generating an indexes matrix for the bootstrap samples. Then we save the bootstrapped fits and resulting means.

```
> #random index to generate bootstrap
> index <- matrix(sample.int(n,n*B,replace=TRUE),nrow=n,ncol=B)
> #initial list for LDA nad QDA fits
> fit_lda <- vector("list",B)
> fit_qda <- vector("list",B)
> #use both methods on the bootstrap samples
> for(i in 1:B){
  ind <- index[,i]
  fit_lda[[i]] <-
    lda(x=Iris[ind,c("Petal.Length","Petal.Width")],grouping=Iris[ind,"Species"])
  fit_qda[[i]] <-
    qda(x=Iris[ind,c("Petal.Length","Petal.Width")],grouping=Iris[ind,"Species"])
}
> #initialize the mu_hat bootstrap estimates
> mu_hat_1 <- mu_hat_2 <- mu_hat_3 <- matrix(0,ncol=B,nrow=2)
> #determine the mu_hat bootstrap estimates
>
> for(i in 1:B){
  mu_hat_temp <- fit_lda[[i]]$means
  mu_hat_1[i] <- mu_hat_temp[1,]
  mu_hat_2[i] <- mu_hat_temp[2,]
  mu_hat_3[i] <- mu_hat_temp[3,]}
```

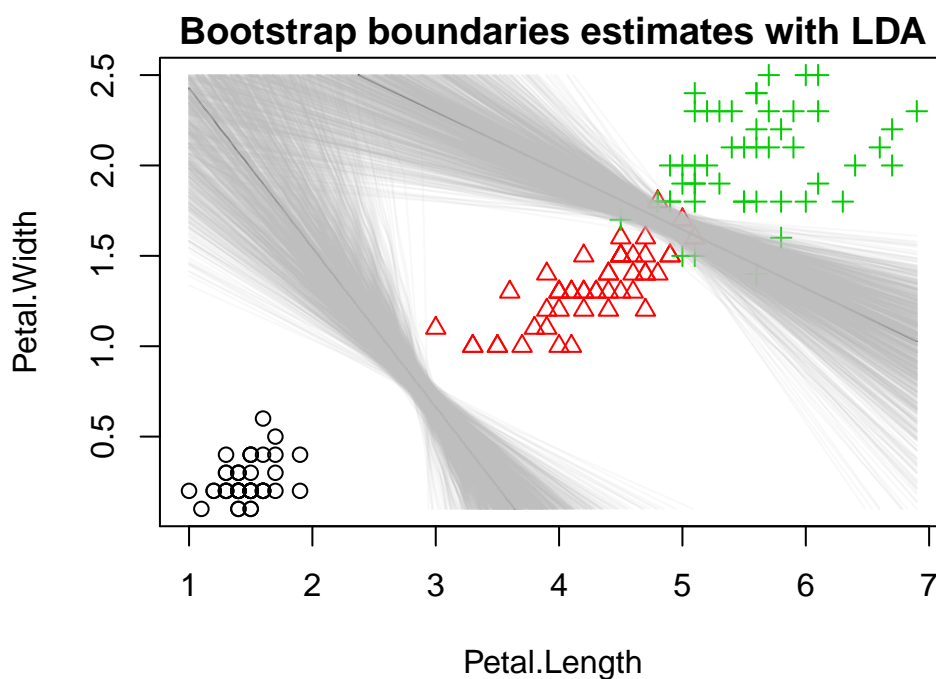
Finally we plot the bootstrapped means in a single plot.

```
> xmin <- min(Iris[, "Petal.Length"])
> xmax <- max(Iris[, "Petal.Length"])
> ymin <- min(Iris[, "Petal.Width"])
> ymax <- max(Iris[, "Petal.Width"])
> plot(mu_hat_1[1, ], mu_hat_1[2, ], xlim = c(xmin,
  xmax), ylim = c(ymin, ymax), xlab = colnames(Iris)[1],
  ylab = colnames(Iris)[2], pch = 4, main = "Bootstrap samples")
> points(mu_hat_2[1, ], mu_hat_2[2, ], col = 2,
  pch = 4)
> points(mu_hat_3[1, ], mu_hat_3[2, ], col = 3,
  pch = 4)
```



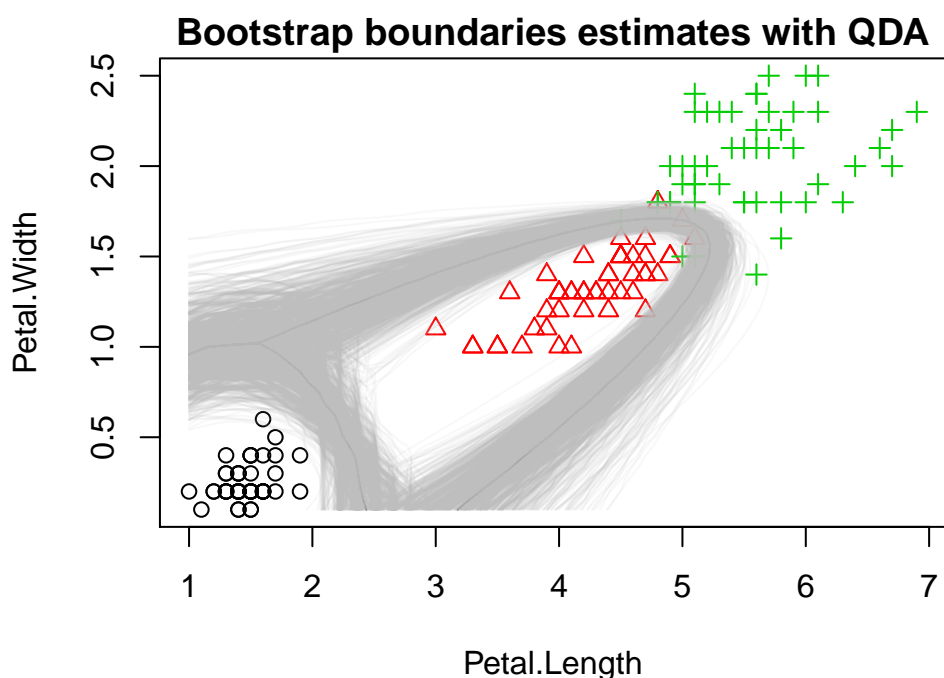
c) Here we plot the classification boundaries of the bootstrap samples. We begin with the LDA method.

```
> predplot(class_lda,Iris,main="Bootstrap boundaries estimates with LDA")
> for(i in 1:B){
  fit<-fit_lda[[i]]
  predplot(fit,Iris,lines.only=TRUE, colcont = adjustcolor("gray",0.1))}
```



Now we plot the boundaries resulting from the QDA method.

```
> predplot(class_qda,Iris,main="Bootstrap boundaries estimates with QDA")
> for(i in 1:B){
  fit<-fit_qda[[i]]
  predplot(fit,Iris,lines.only=TRUE,colcont = adjustcolor("gray",0.1))}
```



d) and e\*) We calculate the bootstrap and the OOB estimate of the generalization error for both methods.

```
> error_lda <- rep(0,B)
> error_lda_OOB <- rep(0,B)
> error_qda <- rep(0,B)
> error_qda_OOB <- rep(0,B)
> for(i in 1:B){

  ind <- index[,i]

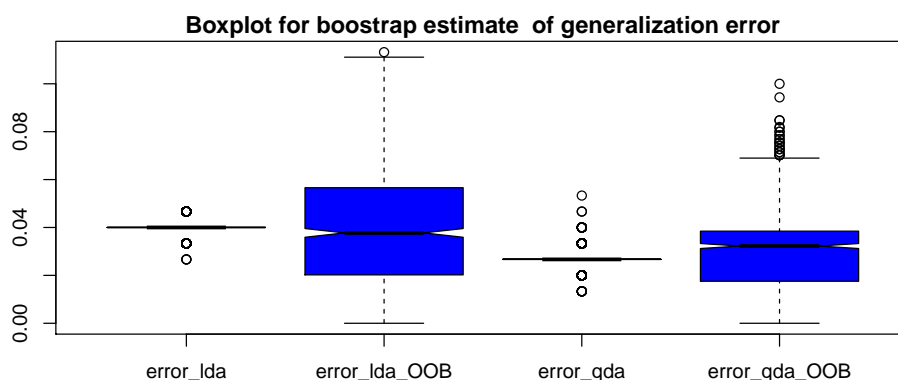
  fit_temp1 <- fit_lda[[i]]
  error_lda[i] <-
  mean(predict(fit_temp1,Iris[,c("Petal.Length","Petal.Width")])$class!=Iris[, "Species"])
  error_lda_OOB[i] <-
  mean(predict(fit_temp1,Iris[-ind,c("Petal.Length","Petal.Width")])$class!=Iris[-ind,"Species"])

  fit_temp2 <- fit_qda[[i]]
  error_qda[i] <-
  mean(predict(fit_temp2,Iris[,c("Petal.Length","Petal.Width")])$class!=Iris[, "Species"])
  error_qda_OOB[i] <-
  mean(predict(fit_temp2,Iris[- ind,c("Petal.Length","Petal.Width") ])$class!=Iris[- ind,"Species"])

}
> ##print the error
> cat("Generalized error for LDA:",format(mean(error_lda),digits=4))
Generalized error for LDA: 0.03986
> cat("Generalized OOB error for LDA:",format(mean(error_lda_OOB),digits=4))
Generalized OOB error for LDA: 0.04198
> cat("Generalized error for QDA:",format(mean(error_qda),digits=4))
Generalized error for QDA: 0.02641
> cat("Generalized OOB error for QDA:",format(mean(error_qda_OOB),digits=4))
Generalized OOB error for QDA: 0.02889
```

We now use a boxplot to determine if there's a significant difference between the medians of the errors.

```
> ##plot boxplot
> boxplot(cbind(error_lda,error_lda_OOB,error_qda,error_qda_OOB),notch=TRUE,col="blue",
          main="Boxplot for bootstrap estimate of generalization error")
```



The OOB estimates are calculated using data not included in the bootstrap sample, so the resulting error has a bigger variance and a smaller bias as compared to the ordinary estimate of the generalization error, as is evident from the boxplot. Judging only by the ordinary estimates of the generalization error, we would conclude that the QDA method is significantly better than the LDA method for the Iris data. However, looking at the OOB estimates the difference between the LDA and the QDA method for the Iris data is not as dramatic. The QDA method still appears to produce a smaller OOB estimate of the generalization error on average. The QDA method also has a lower variance of the OOB estimate of the generalization error, as compared to the LDA method, so we would still choose the QDA as the preferable method for this dataset.