LDA and QDA Examples In R

Cathy Poliak

Example of Linear Discriminant Analysis in R

This is an excerpt from the textbook Lab section for chapter 4.

The Data

We will be looking at the data `Smarket` data part of the `ISLR` package.
This data set consists of percentage returns for the SP 500 stock index over 1250 days from the beginning of 2001 until the end of 2005. For each data we have the following variables:

- `Lag1 - Lag5`: percentage returns for each of the five previous trading days *`Volume`: the number of shares traded on previous day, in billions
- `Today`: the percentage return on the date in question
- `Direction`: whether the market was `Up` or `Down` on this date

Lab Question:

1. Type and run in `R`, `cor(Smarket)` what is your output?
   a. Values between zero and 1.

   b. A list of names.

   c. An error message appears.

   d. Nothing happens.

2. Type and run in `R`, `cor(Smarket[,-9])` (This removes the categorical variable `Direction`) where is there a correlation above absolute value of 0.5?
   a. `Year` and `Lag1`

   b. `Year` and `Volume`

   c. `Year` and `Today`

   d. There are no correlation above absolute value of 0.5.

Separating the Data into Training/Test Data sets

- Error rates for the training data, called **training error** rate will always be lower than the error rate for the test data (**test error rate**).

- Reason: We specifically adjust the parameters of our model to do well on the training data.
- We will use part of the data, instead of of random sample of 75%/25% split we are going to *hold out* one year, 2005. Why?
- Type and run the following in `R`

```
library(ISLR)
attach(Smarket)
train = (Year < 2005)
Smarket.2005 = Smarket[!train,]
dim(Smarket.2005)
Direction.2005 = Smarket.2005$Direction
```

Lab Question:

3. How many observations occurred in 2005?

    a. 252

    b. 1250

    c. 9

    d. 2005

The Model

We will perform LDA to predict the `Direction` of the stock based on `Lag1` and `Lag2` and use only the observations before 2005.

In `R` type and run:

```
library(MASS)
lda.fit = lda(Direction ~ Lag1 + Lag2, data = Smarket, subset=train)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2
## Down  0.04279022  0.03389409
## Up   -0.03954635 -0.03132544
##
## Coefficients of linear discriminants:
##             LD1
## Lag1 -0.6420190
## Lag2 -0.5135293
```

The LDA output gives three objects:

- This estimated prior probabilities: $\hat{\pi}_1$ and $\hat{\pi}_2$.
- The group means, the average of each predictor within each class
- The **coefficients of linear discriminates** the linear combination of the variables that are used to form the LDA decision rule.

The Predictions

The `predict()` function returns a list with three objects.

- **Class** contains the LDA's predictions about the classification.
- **Posterior** is a matrix whose kth column contains the posterior probability that the corresponding observation belongs to the kth class.
- **X** contains the linear discriminates based on the coefficients.

Type and run the following in `R`

```
lda.pred = predict(lda.fit,Smarket.2005)
lda.class = lda.pred$class
table(lda.class,Direction.2005)
```

Lab Question

4. What is the test error rate for this LDA?

   a. 0.44

   b. 0.56

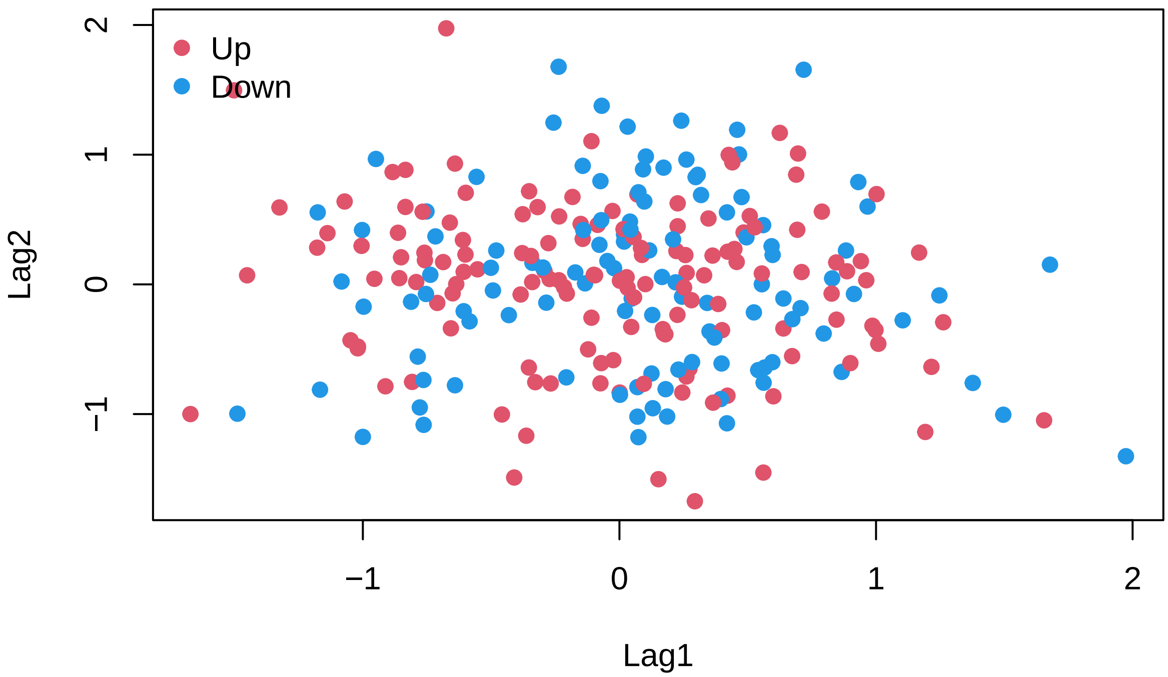
   c. 0.33

   d. 0.68

We could also get the acceptance rate by:

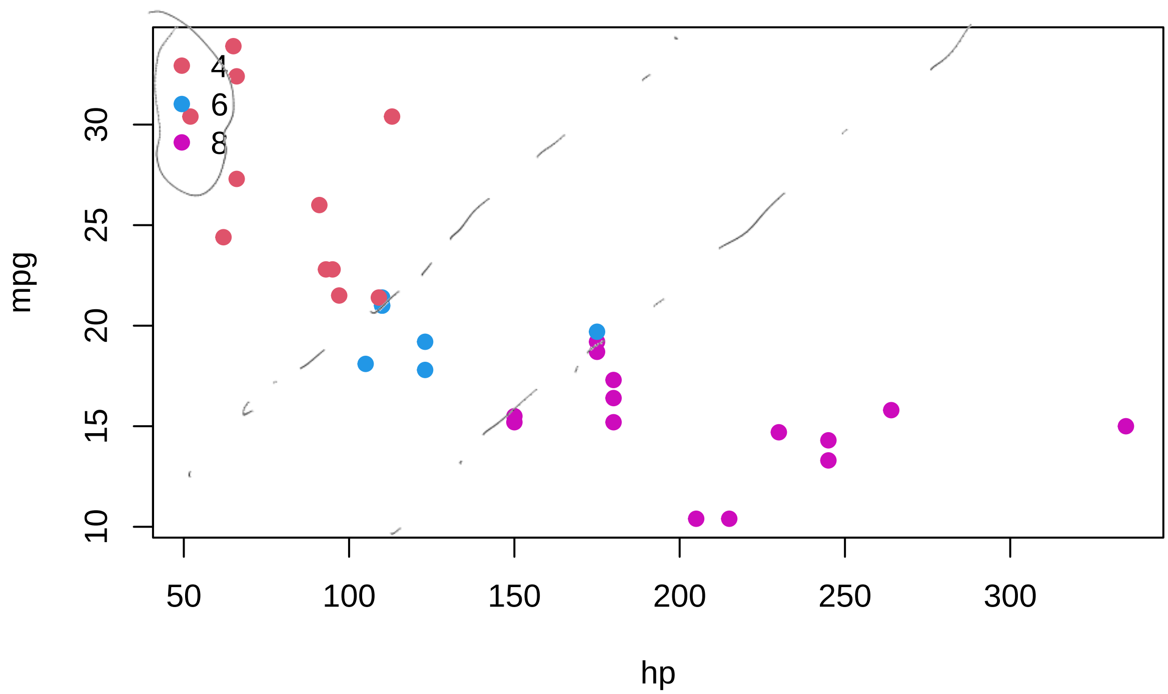
```
mean(lda.class == Direction.2005)
```

```
## [1] 0.5595238
```


Up
Down
2
1
0
−1
Lag2
−1
0
1
2
Lag1

Example 2

Lets go back to the `mtcars` data set. Can we determine the number of cylinders by the hp and mpg?




```r
cars.lda = lda(cylinders ~ mtcars$mpg + mtcars$hp)
cars.lda
```

```
## Call:
## lda(cylinders ~ mtcars$mpg + mtcars$hp)
##
## Prior probabilities of groups:
##       4       6       8
## 0.34375 0.21875 0.43750
##
## Group means:
##   mtcars$mpg mtcars$hp
## 4   26.66364  82.63636
## 6   19.74286 122.28571
## 8   15.10000 209.21429
##
## Coefficients of linear discriminants:
##                   LD1        LD2
## mtcars$mpg -0.2020452 0.25260148
## mtcars$hp   0.0157379 0.02254518
##
## Proportion of trace:
##    LD1    LD2
```

```
## 0.9694 0.0306
```

```
cars.pred = predict(cars.lda)
table(cylinders,cars.pred$class)
```

```
##
## cylinders  4  6  8
##         4  9  2  0
##         6  0  6  1
##         8  0  0 14
```

##Lab Question

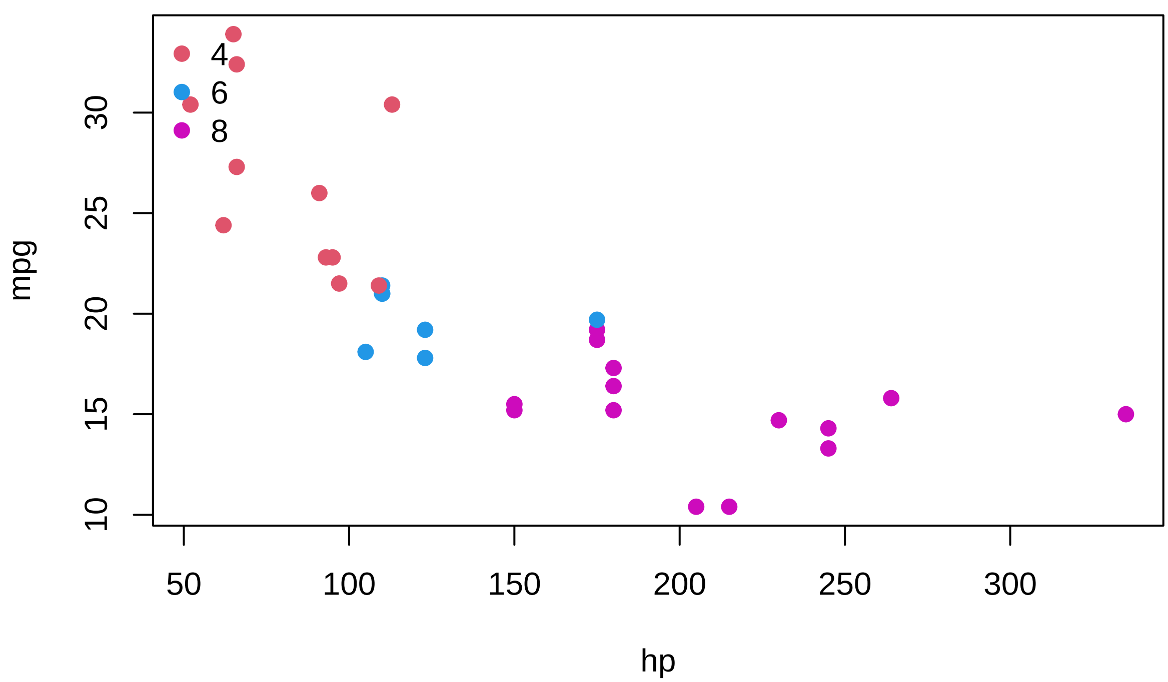
5. What is the error rate for this LDA?

   a. 1

   b. 0.91

   c. 0.09

   d. 0.65



Quadratic Discriminant Analysis

- The **Quadratic Discriminant Analysis** (QDA) assumes that the variance - covaraince matrix is not the same for all K classes.

- That is, it assumes that an observation from the kth class is of the form $X \sim N(\mu_k, \mathbf{\Sigma}_k)$, where $\mathbf{\Sigma}_k$ is a covariance matrix for the kth class.
- Under this assumption, the Bayes classifier assigna an observation $X = x$ to the class for which

$$\delta_k(x) = -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}_k^T\mu_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log(\pi_k)$$

  is the largest.

Fit a QDA Model In R

We will now fit a QDA model to the `Smarket` data using the `qda()` funcition.

```
library(ISLR)
library(MASS)
attach(Smarket)
train = (Year < 2005)
Smarket.2005 = Smarket[!train,]
Direction.2005 = Smarket.2005$Direction
qda.fit = qda(Direction~Lag1 + Lag2, data = Smarket, subset = train)
qda.fit
```

```
## Call:
## qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2
## Down  0.04279022  0.03389409
## Up   -0.03954635 -0.03132544
```

Notice that the output does not contain the coefficients of the lienar discriminant, because this involves quadratic function of the predictors.

Now lets look at the confusion matrix with the test data.

```
library(MASS)
qda.class = predict(qda.fit,Smarket.2005)$class
table(qda.class,Direction.2005)
```

```
##           Direction.2005
## qda.class Down  Up
##      Down   30  20
##      Up     81 121
```

```
mean(qda.class == Direction.2005)
```

```
## [1] 0.5992063
```

Lab Questions

6. What is the test error rate?

    a. 0.599

    b. 0.0794

    c. 0.3214

    d. 0.401

7. What is the accuracy rate based on the test data?

    a. 0.599

    b. 0.0794

    c. 0.3214

    d. 0.401

Comparing Logistic Regression, LDA and QDA

We will use the `Boston` data set in order to predict whether a given suburb has a crime rate above or below the median based on the predictors `age` and `medv`.

```
library(ISLR)
data("Boston")
#Separate the data between training and test
set.seed(10)
sample = sample.int(n = nrow(Boston),
                    size = floor(.75*nrow(Boston)),
                    replace = F)
train = Boston[sample,]
test = Boston[-sample,]
#Create a new variable crim01 that is 1 if above the medain 0 if below the median
train$crim01 = (train$crim > median(train$crim))
test$crim01 = (test$crim > median(test$crim))

#Logistic Regression
fit.glm = glm(crim01 ~ age + medv, data = train, family = "binomial")
glm.pred = predict.glm(fit.glm,test,type = "response")
yHat = glm.pred > 0.5
table(test$crim01,yHat)
```

```
##        yHat
##         FALSE TRUE
##   FALSE    49   15
##   TRUE     13   50
```

```
#LDA results
fit.lda = lda(crim01 ~ age + medv, data = train)
table(test$crim01,predict(fit.lda,test)$class)
```

```
##
##         FALSE TRUE
##   FALSE    45   19
##   TRUE     13   50
```

```
#QDA results
fit.qda = qda(crim01 ~ age + medv, data = train)
table(test$crim01,predict(fit.qda,test)$class)
```

```
##
##         FALSE TRUE
##   FALSE    49   15
##   TRUE     11   52
```

Lab Questions

8. What is the error rate for the logistic regression?

   a. 0.22

   b. 0.25

   c. 0.20

   d. 0.385

9. What is the error rate for the LDA?

   a. 0.22

   b. 0.25

   c. 0.20

   d. 0.385

10. What is the error rate for the QDA?

    a. 0.22

    b. 0.25

    c. 0.20

    d. 0.385