An Introduction to Linear Discriminant Analysis

Arjun Panda

October 11, 2016

Abstract

In this project, I look at R.A. Fisher's canonical **Iris** dataset, which describes four features of three species of the Iris flower. There are 50 observations per species totalling to 150 observations. His question was this: is there a way to use a combination of the features in order to classify each species of Iris? As a solution, he developed *inear discriminant analysis* (LDA). I explain the logic of this method in three parts, culiminating in a LDA on the entire dataset. In the first part, we discriminate bewteen two species with one descriptive variable, or feature. This is not a good method, because of significant overlap in the response. In section two, we create a linear discriminator to classify the same two species, using all features. The model correctly classified all of the holdout observations correctly. In section three, we use multiple linear discriminators to classify all three Iris species. The model was trained on half the dataset and correctly classified 73 of 75 holdouts.

As a note, I will be using the following R packages to streamline data manipulation and analysis:

```
library(ggplot2)
library(reshape2)
library(dplyr)
library(MASS)
library(scatterplot3d)
```

1. Can we seperate I.Versicolor and I. Setosa using one variable?

We start with a univariate method of distinguishing two of the three species. First, we will visualize the distribution of sepal lengths and petal lengths between I. versicolor and I. setosa.

```
# Iris subset with only two species
vers.set <- iris %>% filter(Species == "versicolor" | Species == "setosa")

#Sepal Length
ggplot(vers.set, aes(x = Species, y = Sepal.Length)) +
  geom_point()  + geom_jitter(width = .05) +
  labs(title = "Sepal Lengths (cm) in I. Versicolor and I. Setosa")
```


```
#Petal Length
ggplot(vers.set, aes(x = Species, y = Petal.Length)) +
  geom_point()  + geom_jitter(width = .05) +
  labs(title = "Petal Lengths (cm) in I. Versicolor and I. Setosa")
```

It is clear that while the petal lengths can discriminate between the two species in question, the sepal lengths provide more ambiguous information because of large overlap. We use the pooled t-test with the null hypothesis that both data came from the same distribution to quantify our observation.

```
t.test(vers.set$Sepal.Length ~ vers.set$Species) #Sepal Lengths
```

```
##
##  Welch Two Sample t-test
##
## data:  vers.set$Sepal.Length by vers.set$Species
## t = -10.521, df = 86.538, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.1057074 -0.7542926
## sample estimates:
##     mean in group setosa mean in group versicolor
##                    5.006                    5.936
```

```
t.test(vers.set$Petal.Length ~ vers.set$Species) #Petal Lengths
```

```
##
##  Welch Two Sample t-test
##
## data:  vers.set$Petal.Length by vers.set$Species
## t = -39.493, df = 62.14, p-value < 2.2e-16
```

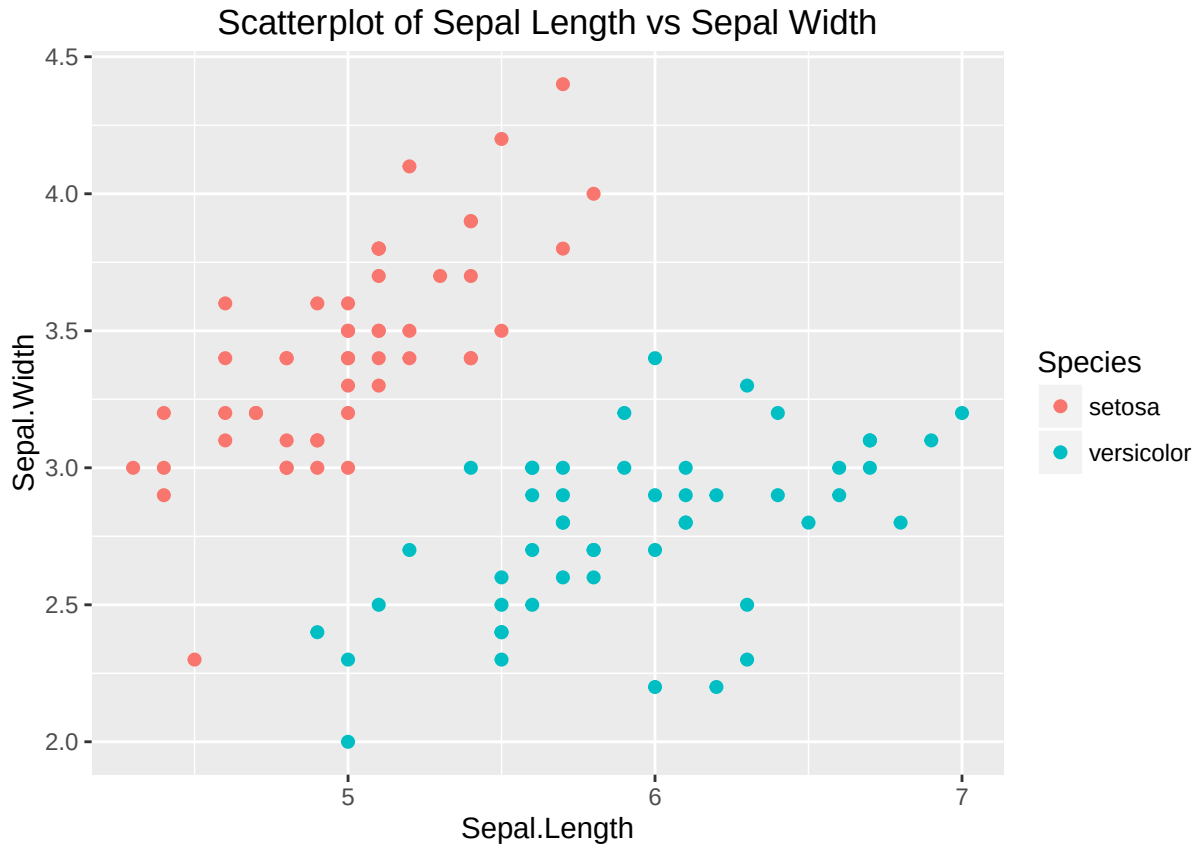
```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.939618 -2.656382
## sample estimates:
##     mean in group setosa mean in group versicolor
##                    1.462                    4.260
```

Since both p-values are much less than alpha = 0.05, we reject the null and assume that the mean sepal and petal lengths are different. However, this does not clarify our classification difficulty from before. **Linear Discriminant Analysis** is a multivariate method that R.A. Fisher developed as a solution to this problem.

2. Linear Discriminant Analysis: I. Versicolor and I. Setosa

The underlying concept of LDA is clear with a simple example. We first plot the bivariate scatterplot of sepal length and sepal width for the two species, neither of which is a good discriminator on its own.

```
vers.set %>% group_by(Species) %>%
  ggplot(data=., aes(x = Sepal.Length, y = Sepal.Width), color = Species) +
  geom_point(aes(color = Species), size = 1.8) +
  ggtitle("Scatterplot of Sepal Length vs Sepal Width")
```



In non-mathematical terms, the line perpendicular to the one which crosses the "center" of each species cluster can best seperate the two. The coefficients for each feature are found by taking the difference of the linear discriminant functions for each class.

```r
lda(formula = Species ~ Sepal.Length + Sepal.Width, data = vers.set)
```

```
## Call:
## lda(Species ~ Sepal.Length + Sepal.Width, data = vers.set)
##
## Prior probabilities of groups:
##     setosa versicolor
##        0.5        0.5
##
## Group means:
##            Sepal.Length Sepal.Width
## setosa            5.006       3.428
## versicolor        5.936       2.770
##
## Coefficients of linear discriminants:
##                     LD1
## Sepal.Length  2.560968
## Sepal.Width  -3.167079
```

The actual discriminant functions corresponding to each class cannot be reconstructed with **MASS::lda()**, but the model can be saved as an object and used to classify the holdouts using **MASS::predict()**.

Next we classify the two species utilizing on all four features of the Iris dataset. This can no longer be visualized because it is in four dimentions. However, we can generalize the procedure above and say that instead of a line separating the classes, there is a *hyperplane* of *N-1* dimentions acting as the decision boundary. We will train the algorithm on the 50 random observations, and validate the model on the other 50.

```r
training.obs <- sample(1:100, 50, replace = F) #random observations
model.1 <- lda(formula = Species~., data = vers.set[training.obs, ]) #training
validate.1 <- predict(object = model.1, newdata = vers.set[-training.obs, ]) #validation

# How many were classified correctly?
test <- data.frame(Actual = vers.set$Species[-training.obs], Predicted = validate.1$class)
table(test$Actual, test$Predicted)
```

```
##
##              setosa versicolor virginica
##   setosa         30          0         0
##   versicolor      0         20         0
##   virginica       0          0         0
```

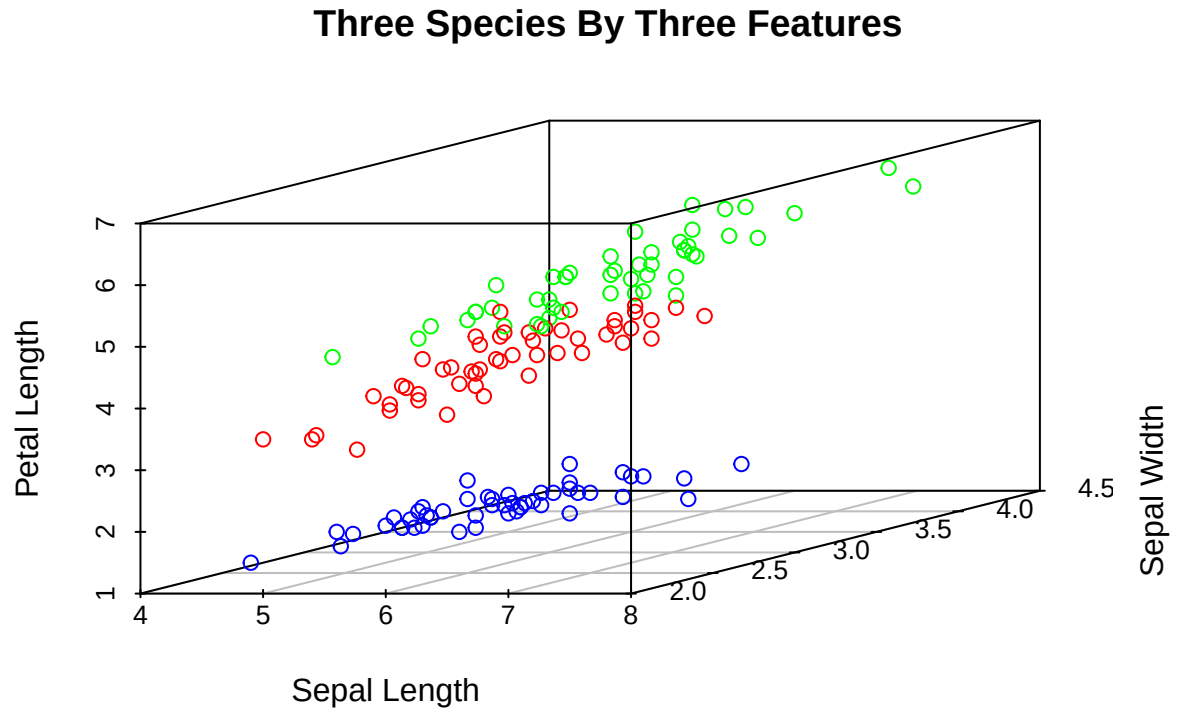
The table shows that our model was able to correctly predict the species of every holdout.

3. Linear Discriminant Analysis: Classifying All Three Species

The linear discriminant in the two-class case is a single decision boundary. However, when more classes are inroduced, multiple decision boundaries are necessary. Although the full Iris dataset has four features, we will start by visualizing I.Versicolor, I. Setosa, and I. Virginica with three defining fearures.

```r
#add color vector to iris
iris$color[iris$Species == "setosa"] <- "blue"
iris$color[iris$Species == "versicolor"] <- "red"
iris$color[iris$Species == "virginica"] <- "green"


scatterplot3d(iris$Sepal.Length,
              y = iris$Sepal.Width,
              z = iris$Petal.Length,
              color = iris$color,
              main = "Three Species By Three Features",
              xlab = "Sepal Length", ylab = "Sepal Width",
              zlab = "Petal Length", angle = 30
              )
```



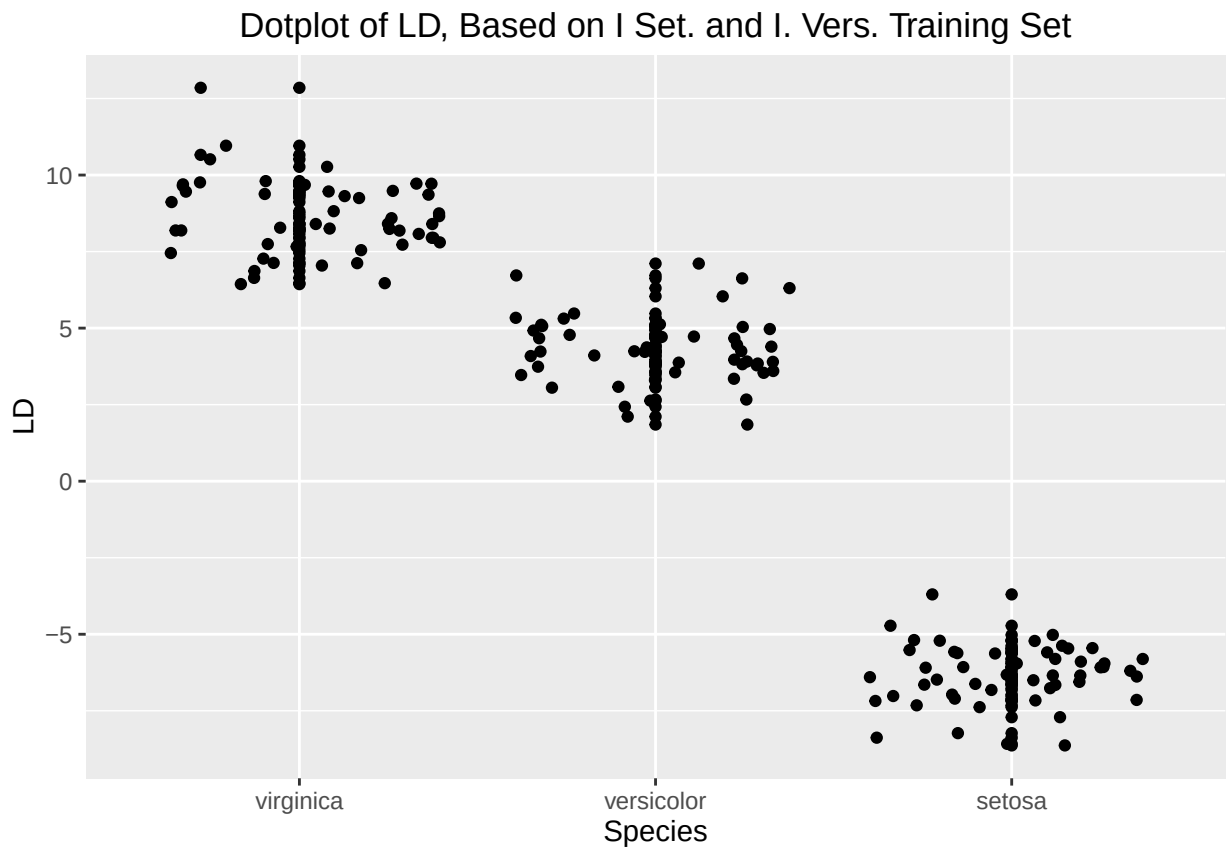
There is clearly a seperation of classes in three dimentional feature space. But before building an LDA model with the full Iris dataset, its worth asking if we even need one in order to classify the third species. Maybe the one from above, with one decision boundary in four dimensions is good enough? Lets check:

```r
virginica.ld <- predict(object = model.1, newdata = iris[iris$Species == "virginica",-6])
setosa.ld <- predict(object = model.1, newdata = iris[iris$Species == "setosa",-6])
versicolor.ld <- predict(object = model.1, newdata = iris[iris$Species == "versicolor",-6])


lds <- data.frame(virginica.ld$x, versicolor.ld$x, setosa.ld$x)
colnames(lds) <- c("virginica", "versicolor", "setosa")
```

```
lds <- melt(lds)

ggplot(lds, aes(x = variable, y= value )) +
  geom_point() +
  geom_jitter() +
  labs(title = "Dotplot of LD, Based on I Set. and I. Vers. Training Set",
       y = "LD",
       x= "Species")
```



We can usually tell the difference between I. Virginica and I. Versicolor with this model, but not always because of some overlap. So as a final analysis, we will use all four features to classify three species. We will train the algorithm on half of the data (N = 75), and verify on the rest (N = 75).

```
iris <- iris[ ,-6] #remove color vector from earlier
training.2 <- sample(1:150, 75, replace = F) #random observations
model.2 <- lda(formula = Species~., data = iris[training.2, ], prior = c(1,1,1)/3) #training
validate.2 <- predict(object = model.2, newdata = iris[-training.obs, ]) #validation
print(model.2)
```

```
## Call:
## lda(Species ~ ., data = iris[training.2, ], prior = c(1, 1, 1)/3)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
```

```
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa         5.052381    3.404762     1.533333   0.2761905
## versicolor     5.812000    2.780000     4.168000   1.3080000
## virginica      6.665517    2.982759     5.603448   2.0586207
##
## Coefficients of linear discriminants:
##                     LD1       LD2
## Sepal.Length  0.6815802  1.198453
## Sepal.Width   1.4614689  1.212930
## Petal.Length -1.7839285 -1.249851
## Petal.Width  -2.9166430  2.188699
##
## Proportion of trace:
##    LD1    LD2
## 0.9908 0.0092
```

The model uses two LDs to classify the three species. Proportion of trace tells us that LD1 accounts for 99.4 percent of the between-group variance, ands LD2 accounts for the rest. Intuitively this makes sense, because we were more or less able to classify I. Virginica based on the single LD derived from the other two species. However, two LDs allows a very accurate classification, as evidenced by the contingency table below.

```
table(iris$Species[-training.obs], validate.2$class)
```

```
##
##              setosa versicolor virginica
##   setosa         30          0         0
##   versicolor      0         18         2
##   virginica       0          1        49
```

We see here that one I. Versicolor was misclassified as I. Virginica, and one I. Virginicas as I. Versicolor. Therefore, 97.33 percent of the holdouts were classified correctly. We conclude that LDA is a very effective classification method.

As a postscript, I want to mention that LDA is a *supervised* machine learning method, because it utilizes the pre-assigned classes in the dataset to find the classification rule. An *unsupervised* method, such as clustering, looks for categories that minimize the distances from each observation to the determined centroid for each class in a stochastic manner.

References

Texts Trumbo, Bruce E. "8: Classifying Irises." Learning Statistics with Real Data. Pacific Grove, CA: Duxbury/Thomson Learning, 2002. N. pag. Print.

Mitchell, Melanie. "Linear Classification." Machine Learning. Web. 16 May 2016.

Schalkoff R. J. (1997). Artificial Neural Networks. The McGraw-Hill Press, USA

Software

H. Wickham. ggplot2: elegant graphics for data analysis. Springer New York, 2009.

Hadley Wickham and Romain Francois (2015). dplyr: A Grammar of Data Manipulation. R package version 0.4.3.

Hadley Wickham (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20.

Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN0-387-95457-0

Ligges, U. and M?chler, M. (2003). Scatterplot3d - an R Package for Visualizing Multivariate Data. Journal of Statistical Software 8(11), 1-20.