Extending Linear Discriminant Analysis

Types of Errors

Let's say you work for a bank and you're tasked with building a model that will predict whether someone will default given their credit history (i.e. balance).

What could go wrong?

```
conf_log
```

```
##
## my_log_pred   No  Yes
##          No 9625  233
##         Yes   42  100
```

1. Deny credit to someone who would not have defaulted (false positive)
2. Give credit to somone who will default (false negative)

What could we change to lower our false positive rate?

```r
my_log_pred <- ifelse(m1$fit < 0.6, "No", "Yes")
conf_log_6 <- table(my_log_pred, Default$default)
conf_log_6
```

```
##
## my_log_pred   No  Yes
##          No 9643  258
##         Yes   24   75
```

```r
conf_log
```

```
##
## my_log_pred   No  Yes
##          No 9625  233
##         Yes   42  100
```

And if we raise the threshold a bit more?

```
my_log_pred <- ifelse(m1$fit < 0.7, "No", "Yes")
conf_log_7 <- table(my_log_pred, Default$default)
conf_log_7
```

```
##
## my_log_pred   No  Yes
##         No  9654  284
##         Yes   13   49
```

```
conf_log_6
```

```
##
## my_log_pred   No  Yes
##         No  9643  258
##         Yes   24   75
```

False positive rate

Of all of the actual negatives, how many did we declare positive?

$$FPR = FP/(FP + TN)$$

```
thresh <- c(0.5, 0.6, 0.7)
FPR <- c(conf_log["Yes", "No"]/sum(conf_log[, "No"]),
        conf_log_6["Yes", "No"]/sum(conf_log_6[, "No"]),
        conf_log_7["Yes", "No"]/sum(conf_log_7[, "No"]))
FPR
```

```
## [1] 0.004344678 0.002482673 0.001344781
```

True positive rate

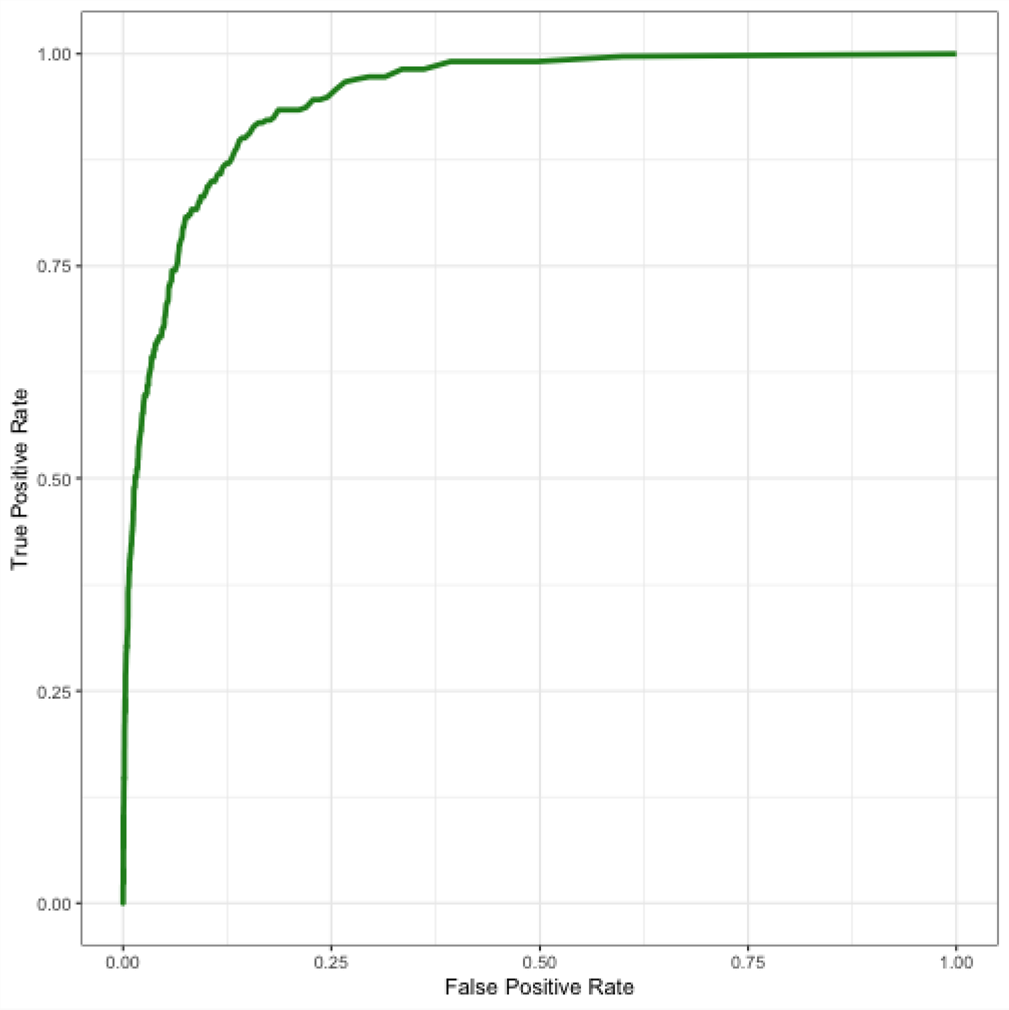
Of all of the actual positives, how many did we declare positive?

$$TPR = TP/(TP + FN)$$

```
thresh <- c(0.5, 0.6, 0.7)
TPR <- c(conf_log["Yes", "Yes"]/sum(conf_log[, "Yes"]),
         conf_log_6["Yes", "Yes"]/sum(conf_log_6[, "Yes"]),
         conf_log_7["Yes", "Yes"]/sum(conf_log_7[, "Yes"]))
TPR
```

```
## [1] 0.3003003 0.2252252 0.1471471
```

ROC curve



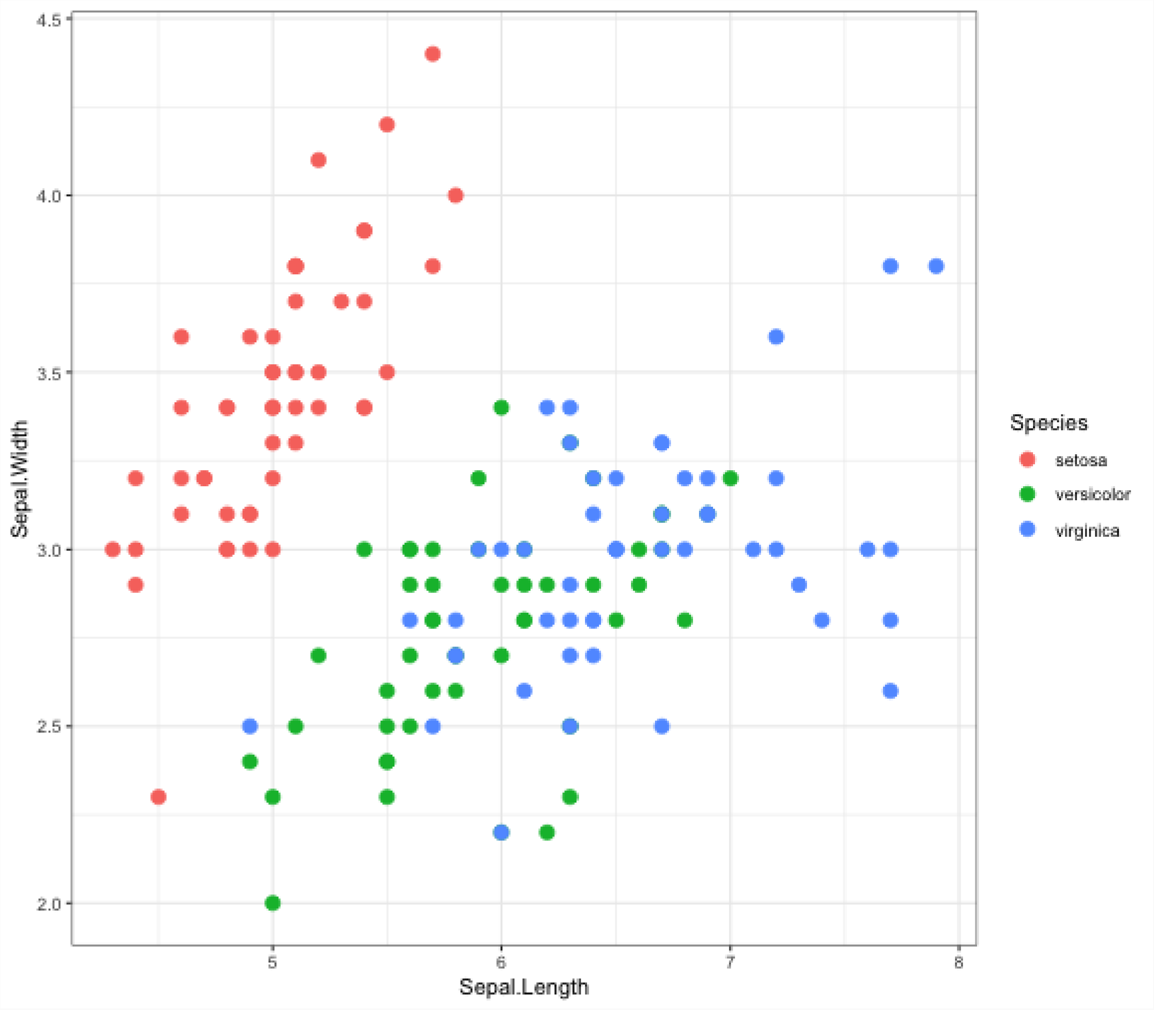
1.00
0.75
0.50
0.25
0.00
True Positive Rate
0.00
0.25
0.50
0.75
1.00
False Positive Rate

LDA with $p > 1$, $K > 2$

```r
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Fisher's Irises



LDA Classification

Can be done quickly using the `lda()` function in the `MASS` package.

```
library(MASS)
mlda <- lda(Species ~ Sepal.Length + Sepal.Width,
            data = iris)
mlda_pred <- predict(mlda)
(conf_mlda <- table(mlda_pred$class,
                    iris$Species))
```

```
##
##             setosa versicolor virginica
##   setosa         49          0         0
##   versicolor      1         36        15
##   virginica       0         14        35
```

LDA Misclassification Rate

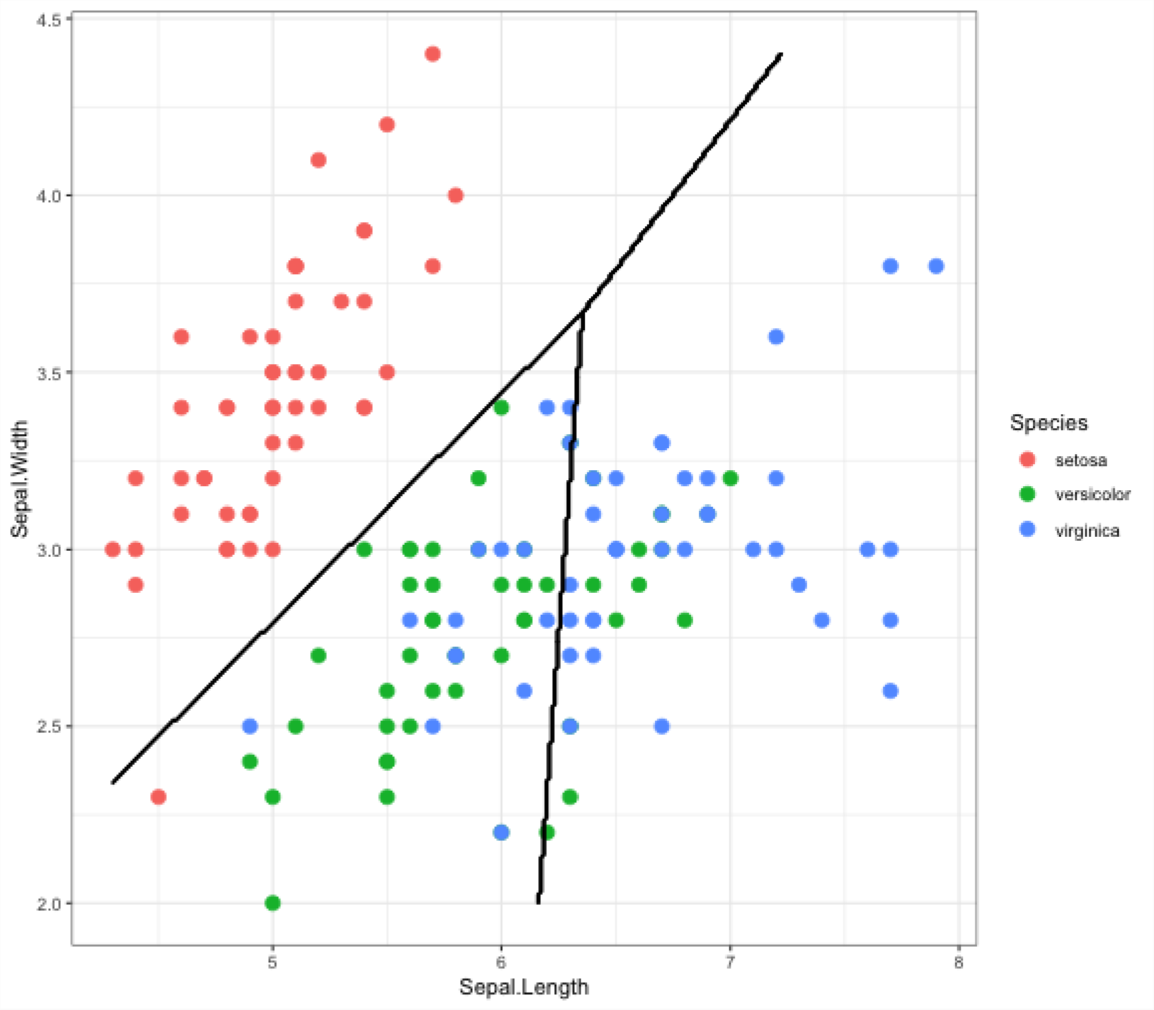
```
conf_mlda
```

```
##
##             setosa versicolor virginica
##   setosa         49          0         0
##   versicolor      1         36        15
##   virginica       0         14        35
```

```
(sum(conf_mlda) - sum(diag(conf_mlda)))/
  sum(conf_mlda)
```

```
## [1] 0.2
```

LDA decision boundaries



LDA summary

- Focuses on modeling the predictors: $f_k(X) = \text{Normal}(\mu_k, \sum)$
- Uses Bayes Rule to find the probabilities that an observation in is each class given the probabilities of all the $\pi_k f_k(X)$.

Note

- Allows each class to have its own μ_k.
- Constrains $\sum$ to be shared between the classes (inducing linear decision boundaries).

Question

On data set with 15 predictors and 1000 observations, would you worry more about the *bias* or the *variance* of this method?

Quadratic Discriminant Analysis (QDA)

Focuses on modeling the predictors: $f_k(X) = \text{Normal}(\mu_k, \sum_k)$

Allow each class to have it's own covariance matrix

QDA

```
mqda <- qda(Species ~ Sepal.Length + Sepal.Width,
            data = iris)
mqda_pred <- predict(mqda)
(conf_mqda <- table(mqda_pred$class,
                    iris$Species))
```

```
## 
##              setosa versicolor virginica
##   setosa         49          0         0
##   versicolor      1         37        16
##   virginica       0         13        34
```

QDA Misclassification Rate

```
conf_mqda
```

```
##
##               setosa versicolor virginica
##   setosa          49          0         0
##   versicolor       1         37        16
##   virginica        0         13        34
```

```
(sum(conf_mqda) - sum(diag(conf_mqda)))/sum(conf_m
```

```
## [1] 0.2
```

LDA decision boundaries