LINEAR DISCRIMINANT ANALYSIS (LDA)

Introducción a la Ciencia de Datos

Linear Discriminant Analysis

- LDA undertakes the same task as Logistic Regression. It classifies data based on categorical variables
  - Making profit or not
  - Buy a product or not
  - Satisfied customer or not
  - Political party voting intention

Why Linear? Why Discriminant?

- LDA involves the determination of linear equation (just like linear regression) that will predict which group the case belongs to:

$$D = v_1 X_1 + v_2 X_2 + \ldots + v_i X_i + a$$

  - D: discriminant function
  - v: discriminant coefficient or weight for the variable
  - X: variable
  - a: constant

Purpose of LDA

- Choose the v's in a way to maximize the distance between the means of different categories
- We want to discriminate between the different categories
- *Think of food recipe*: changing the proportions (weights) of the ingredients will change the characteristics of the finished cakes. Hopefully that will produce different types of cake!

Why not Logistic Regression?

- Logistic regression is unstable when the classes are well separated
- In the case where **n** is small, and the distribution of predictors X is approximately normal, then LDA is more stable than Logistic Regression
- LDA is more popular when we have more than two response classes

Linear Discriminant Analysis

- The approach is to model the distribution of X in each of the classes separately, and the use Bayes theorem to flip things around and obtain P(Y|X).

Estimating Bayes' Classifier

- With Logistic Regression we modeled the probability of Y being from the k[th] class as:

$$p(X) = P(Y = k \mid X = x) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- However, Bayes' Theorem states

$$p(X) = P(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

π_k : Probability of coming from class k (prior probability)

$f_k(x)$: Density function for X given that X is an observation from class k

Estimate π_k and $f_k(x)$

- We can estimate π_k and $f_k(x)$ to compute $p(X)$

- The most common model for $f_k(x)$ is the Normal Density:

$$f_k(x) = \frac{1}{\sqrt{2\pi_k}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

- Using the density, we only need to estimate three quantities to compute *p(X)*:

$$\mu_k \qquad \sigma_k^2 \qquad \pi_k$$

Use Training Data set for Estimation

- The mean μ_k could be estimated by the average of all training observations from the kth class.
- The variance σ_k^2 could be estimated as the weighted average of variances of all k classes.
- And, π_k is estimated as the proportion of the training observations that belong to the kth class.

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}_k^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$\hat{\pi}_k = n_k / n$$

A Simple Example with One Predictor (p = 1)

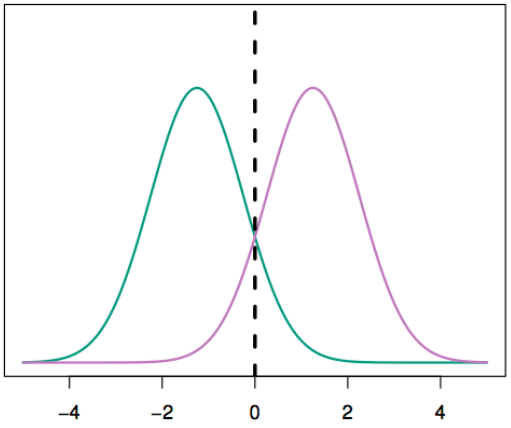
- Suppose we have only one predictor (p = 1)
- Two normal density function $f_1(x)$ and $f_2(x)$, represent two distinct classes
- The two density functions overlap, so there is some uncertainty about the class to which an observation with an unknown class belongs
- The dashed vertical line represents Bayes' decision boundary

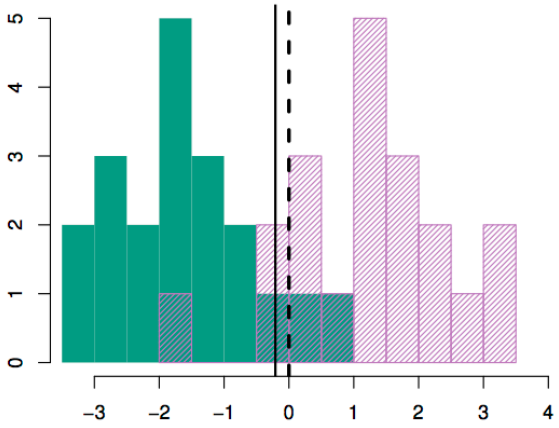
Apply LDA

- LDA starts by assuming that each class has a normal distribution with a common variance and the observations are a random sample
- The mean and the variance are estimated
- Finally, Bayes' theorem is used to compute p_k and the observation is assigned to the class with the maximum probability among all k probabilities

Apply LDA

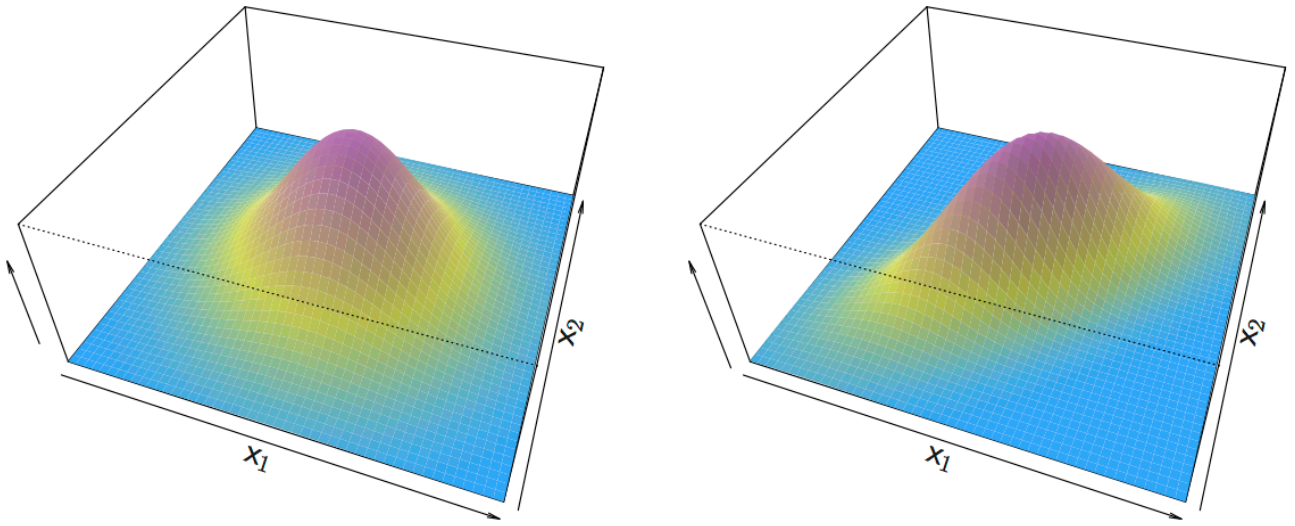
- 20 observations were drawn from each of the two classes
- The dashed vertical line is the Bayes' decision boundary
- The solid vertical line is the LDA decision boundary
  - Bayes' error rate: 10.6%
  - LDA error rate: 11.1%
- Thus, LDA is performing pretty well!





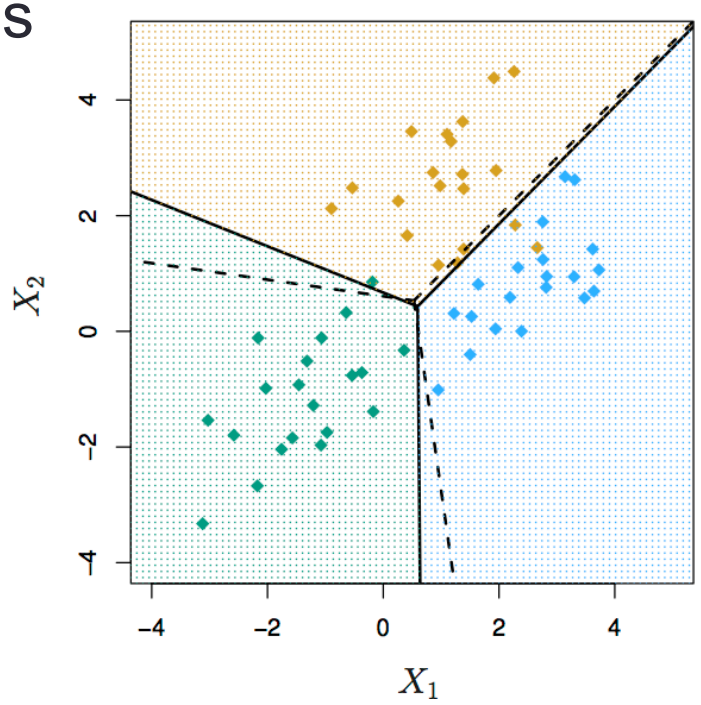
An Example When p > 1

- If X is multidimensional (p > 1), we use exactly the same approach except the density function f(x) is modeled using the multivariate normal density



An Example When p = 2

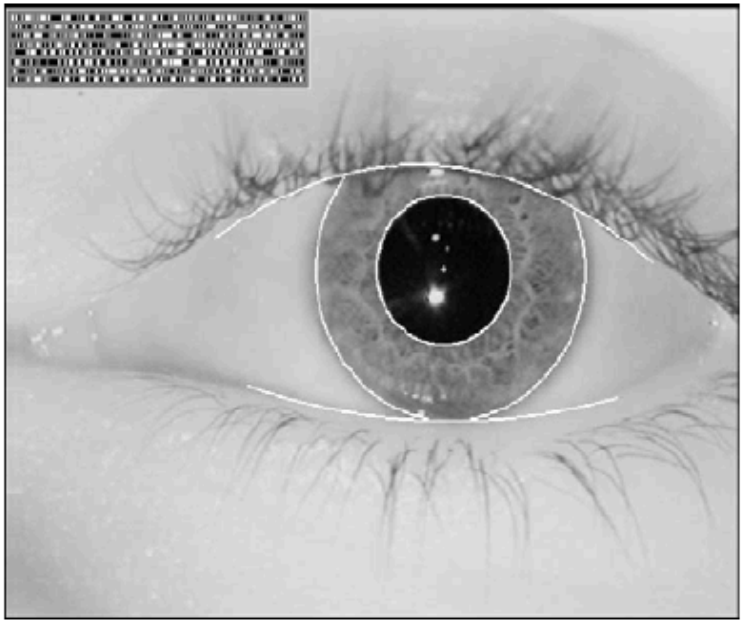
- We have two predictors (p = 2)
- Three classes
- 20 observations were generated from each class
- The solid lines are Bayes' boundaries
- The dashed lines are LDA boundaries

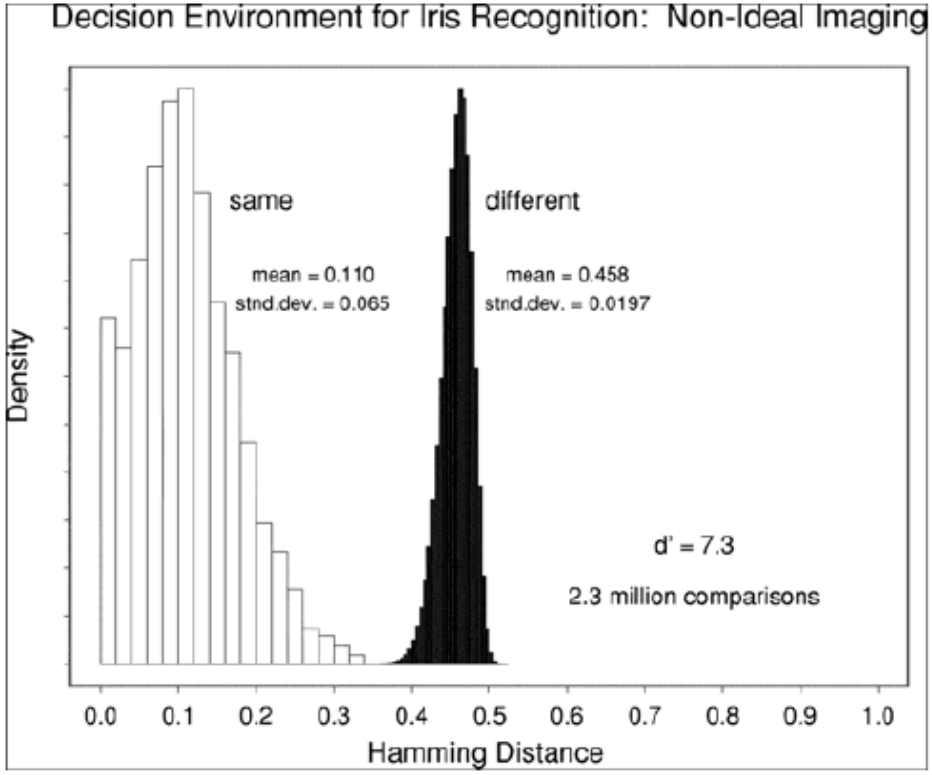


Recommendations

- The LDA solution depends on inverting a covariance matrix, thus a unique solution exists
- The data must contain more samples than predictors, and the predictors **must be independent**
- It is recommended that predictors be centered and scaled and that near-zero variance predictors be removed
- It is also recommended that LDA be used on data sets that have at least 5-10 times more samples than predictors

Example





How Iris Recognition Works, John Daugman

IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. VOL. 14. NO. 1. JANUARY 2004

QUADRATIC DISCRIMINANT ANALYSIS (QDA)

Introducción a la Ciencia de Datos

Quadratic Discriminant Analysis (QDA)

- LDA assumed that every class has the same variance / covariance
  - However, LDA may perform poorly if this assumption is far from true
- QDA works identically as LDA except that it estimates separate variances/covariance for each class
  - Now, the number of predictors must be less than the number of cases within each class.
  - Also, the predictors within each class must not have pathological levels of collinearity

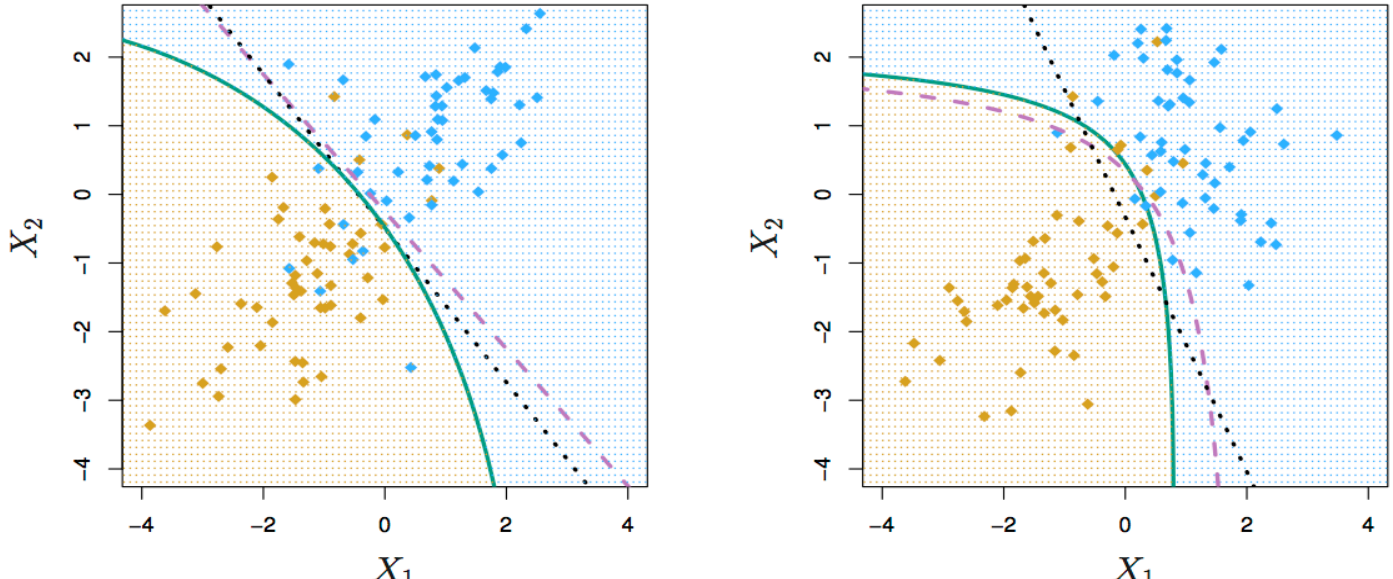
Which is better? LDA or QDA?

- Since QDA allows for different variances among classes, the resulting boundaries become quadratic!

- Which approach is better: LDA or QDA?
  - QDA will work best when the variances are very different between classes and we have enough observations to accurately estimate the variances
  - LDA will work best when the variances are similar among classes or we don't have enough data to accurately estimate the variances

Comparing LDA to QDA

- Black dotted: LDA boundary
- Purple dashed: Bayes' boundary
- Green solid: QDA boundary
- Left: variances of the classes are equal (LDA is better fit)
- Right: variances of the classes are not equal (QDA is better fit)



LDA & QDA

R session

The Stock Market Data

```
library(MASS)
library(ISLR)

# First check LDA assumtions!

# The observations are a random sample: we will assume there are...
# Each predictor variable is normally distributed
shapiro.test(Smarket$Lag1)
shapiro.test(Smarket$Lag2)

qqnorm(y = Smarket$Lag1)
qqline(y = Smarket$Lag1)

# Predictors have a common variance
boxplot(Smarket[,2:3])

var(Smarket$Lag1)
var(Smarket$Lag2)

# Linear Discriminant Analysis
lda.fit <- lda(Direction~Lag1+Lag2,data=Smarket, subset=Year<2005)
lda.fit

plot(lda.fit, type="both")

Smarket.2005 <- subset(Smarket,Year==2005)
lda.pred <- predict(lda.fit,Smarket.2005)
class(lda.pred)
lda.pred

data.frame(lda.pred)

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

The Stock Market Data

```
library(klaR)
partimat(Direction~Lag1+Lag2, data=Smarket ,method="lda")

# Check same variance but this time for each class
var(Smarket[Smarket$Direction == "Up",]$Lag1)
var(Smarket[Smarket$Direction == "Up",]$Lag2)
var(Smarket[Smarket$Direction == "Down",]$Lag1)
var(Smarket[Smarket$Direction == "Down",]$Lag2)

# QDA
qda.fit <- qda(Direction~Lag1+Lag2, data=Smarket, subset=Year<2005)
qda.fit

qda.pred <- predict(qda.fit,Smarket.2005)
class(qda.pred)
data.frame(qda.pred)

table(qda.pred$class,Smarket.2005$Direction)
mean(qda.pred$class==Smarket.2005$Direction)

partimat(Direction~Lag1+Lag2, data=Smarket ,method="qda")
```

The Iris Data

- When there are more than two groups we can estimate more than one discriminant function:

$$D_1 = v_1 X_1 + v_2 X_2 + \ldots + v_i X_i + a$$

$$D_2 = w_1 X_1 + w_2 X_2 + \ldots + w_i X_i + a$$

- For example, when there are three groups, we could estimate (1) a function for discriminating between group 1 and groups 2 and 3 combined, and (2) another function for discriminating between group 2 and group 3.

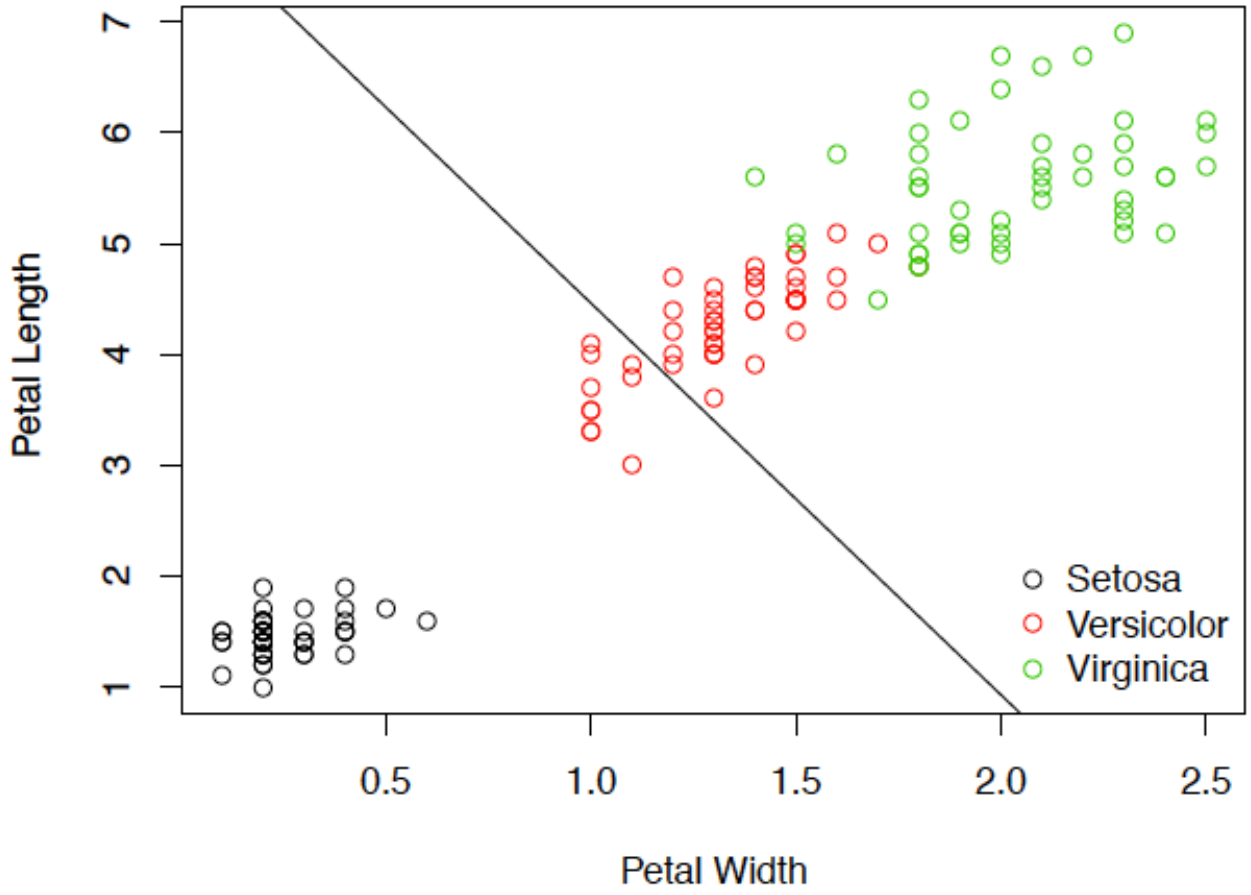
The Iris Data

```
data(iris)
iris.lda <- lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,  data = iris)
iris.lda

partimat(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data=iris,
method="lda")

TrainData <- iris[,1:4]
TrainClasses <- iris[,5]
library(caret)
ldaFit <- train(TrainData, TrainClasses,
                method = "lda",
                preProcess = c("center", "scale"),
                tuneLength = 10,
                trControl = trainControl(method = "cv"))
ldaFit
confusionMatrix(ldaFit)

qdaFit <- train(TrainData, TrainClasses,
                method = "qda",
                preProcess = c("center", "scale"),
                tuneLength = 10,
                trControl = trainControl(method = "cv"))
qdaFit
confusionMatrix(qdaFit)
```

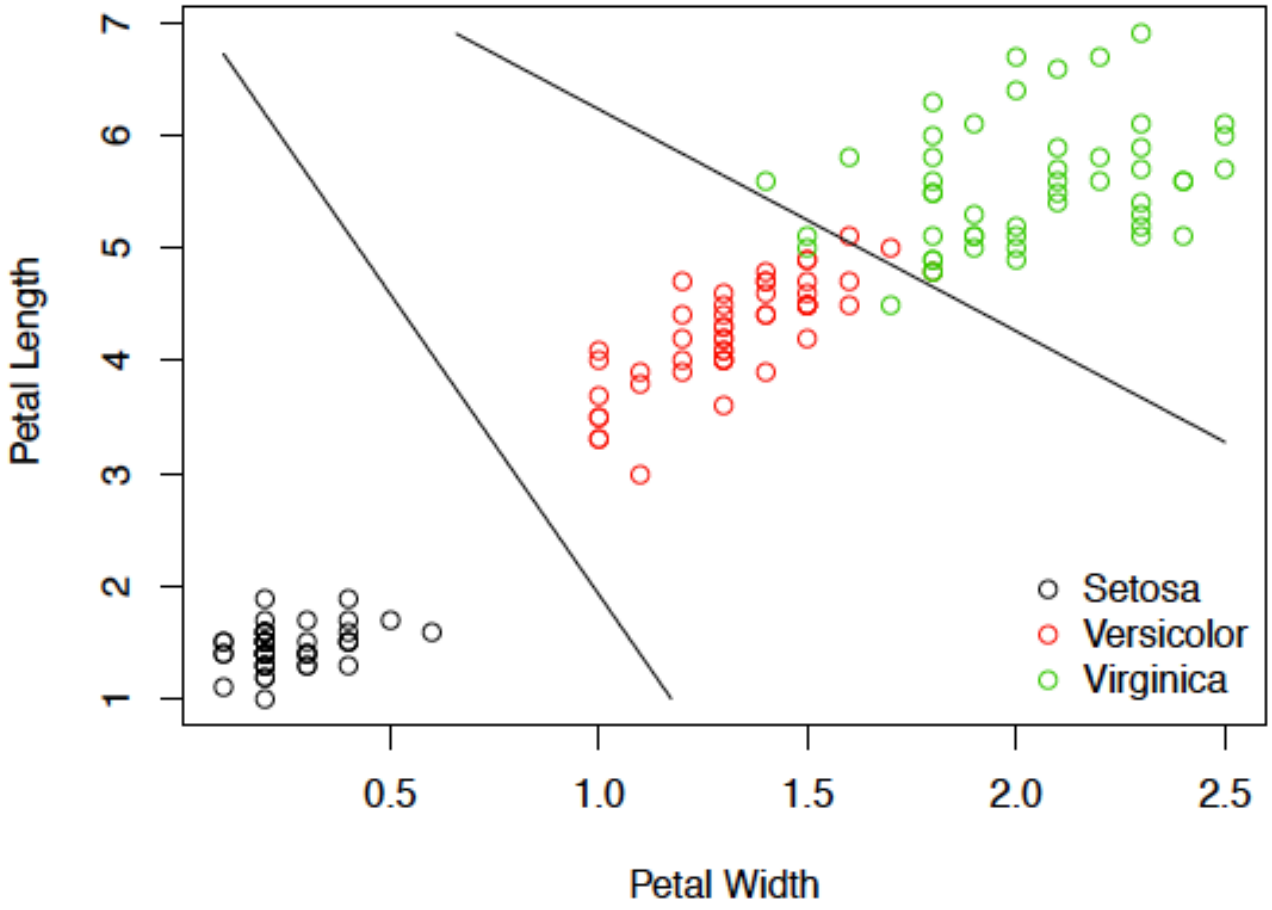
COMPARISON OF CLASSIFICATION METHODS

Introducción a la Ciencia de Datos

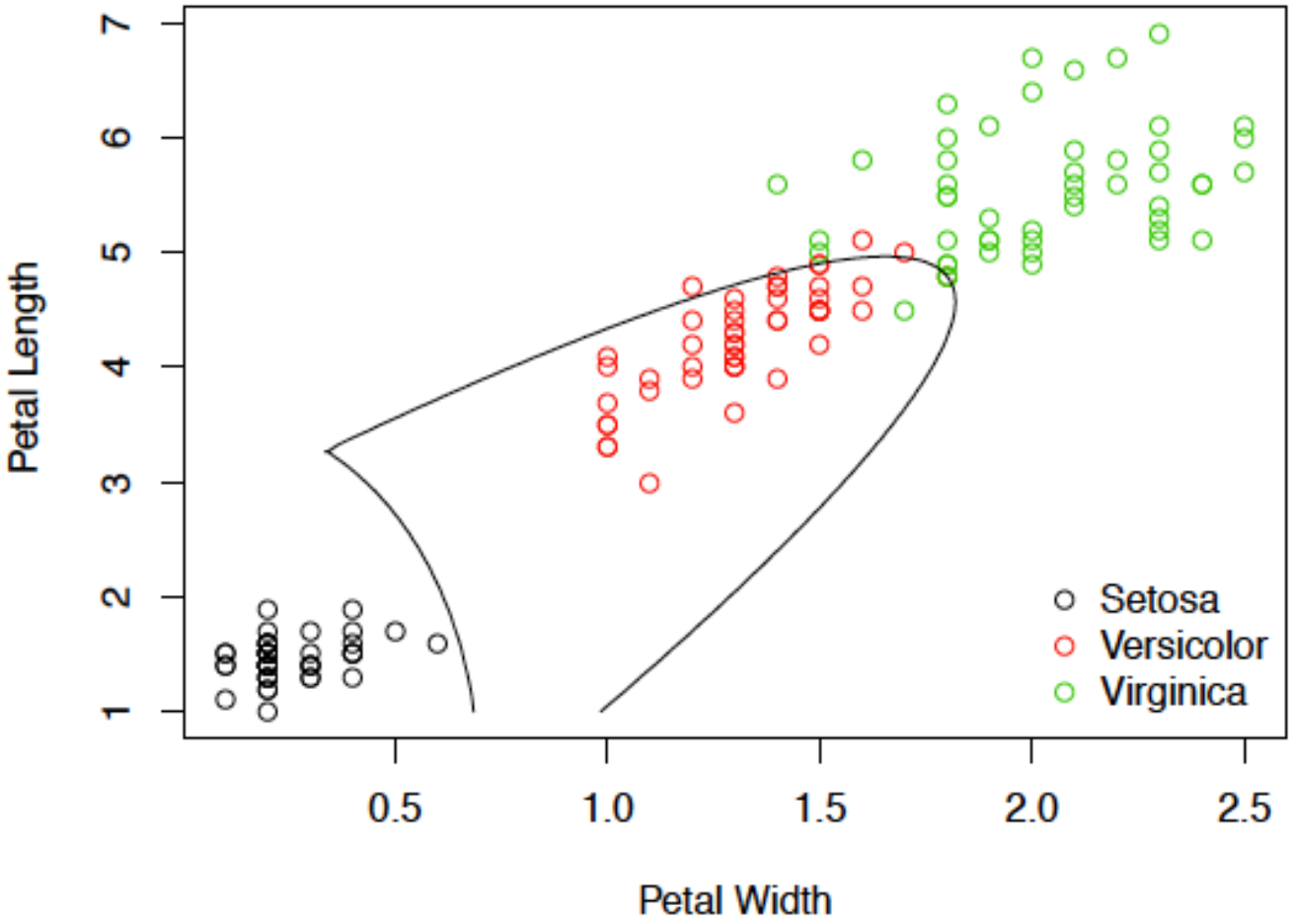
Iris data (glm)



Iris data (lda)



Iris data (qda)



Comparison of Classification Methods

- k-NN
- Logistic Regression
- LDA
- QDA

Logistic Regression vs. LDA

- Similarity: Both Logistic Regression and LDA produce linear boundaries
- Difference: LDA assumes that the observations are drawn from the normal distribution with common variance, while logistic regression does not have this assumption. LDA would do better than Logistic Regression if the assumption of normality holds, otherwise logistic regression can outperform LDA

k-NN vs. (LDA and Logistic Regression)

- k-NN takes a completely different approach
- k-NN is completely non-parametric: No assumptions are made about the shape of the decision boundary!
- Advantage of k-NN: We can expect k-NN to dominate both LDA and Logistic Regression when the decision boundary is highly non-linear
- Disadvantage of k-NN: k-NN does not tell us which predictors are important (no table of coefficients!)

QDA vs. (LDA, Logistic Regression, and k-NN)

- QDA is a compromise between non-parametric k-NN method and the linear LDA and logistic regression

- If the true decision boundary is:
  - Linear: LDA and Logistic outperforms
  - Moderately Non-linear: QDA outperforms
  - More complicated: k-NN is superior

Exercise 1 (Smarket data)

- Try `lda` with all Lag variables
- Make a quick comparison between logistic regression and `lda`.

- Try with `qda` and compare all three methods. Plot the results.

Exercise 2

- Using only the information in file clasif_train_alumnos.csv:
  - Compare lda and qda using Wilcoxon.
  - Perform a multiple comparison using Friedman.
  - Using Holm see if there is a winning algorithm (even if Friedman says there is no chance…).

Bibliography

- DSO 530: Applied Modern Statistical Learning Techniques. Abbass Al Sharif. http://www.alsharif.info/#!iom530/c21o7
- An Introduction to Statistical Learning. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. http://www-bcf.usc.edu/~gareth/ISL/index.html
- Applied Predictive Modeling. Max Kuhn and Kjell Johnson. 2013th Edition. Springer. http://appliedpredictivemodeling.com