Linear and Quadratic Discrimminant Analysis

Rebecca C. Steorts, Duke University

STA 325, Chapter 4 ISL

Agenda

- Classification revisited
- Issues with linear regression
- Linear discrimminant analysis
- Lab with tasks to complete for homework

Classification

Classification is a predictive task in which the response takes values across discrete categories (i.e., not continuous), and in the most fundamental case, two categories.

Examples:

- Predicting whether a patient will develop breast cancer or remain healthy, given genetic information
- Predicting whether or not a user will like a new product, based on user covariates and a history of his/her previous ratings
- Predicting the region of Italy in which a brand of olive oil was made, based on its chemical composition
- Predicting the next elected president, based on various social, political, and historical measurements

Classification

The point of classification methods is to accurately assign new, unlabeled examples, from the test data to these classes.

This is "supervised" learning because we can check the performance on the labeled training data.

The point of calculating information was to select features which made classification easier.

Classification versus Clustering

Similar to our usual setup, we observe pairs (x_i, y_i), $i = 1, \ldots n$, where y_i gives the class of the ith observation, and $x_i \in \mathbb{R}^p$ are the measurements of p predictor variables

Though the class labels may actually be $y_i \in \{\text{healthy}, \text{sick}\}$ or $y_i \in \{\text{Sardinia}, \text{Sicily}, ...\}$, but we can always encode them as

$$y_i \in \{1, 2, \ldots K\}$$

where K is the total number of classes

Why is classification different from clustering?

- In clustering there is not a pre-defined notion of class membership (and sometimes, not even K).
- We are not given the response variable y_i but only x_i, $i = 1, \ldots n$ (meaning we have no labeled data).

Classification versus clustering

Constructed from training data (x_i, y_i), $i = 1, \ldots n$, we denote our classification rule by $\hat{f}(x)$; given any $x \in \mathbb{R}^p$, this returns a class label $\hat{f}(x) \in \{1, \ldots K\}$

As before, we will see that there are two different ways of assessing the quality of $\hat{f}$: its predictive ability and interpretative ability

Binary classification and linear regression

Let's start off by supposing that $K = 2$, so that the response is $y_i \in \{1, 2\}$, for $i = 1, \ldots n$

You already know a tool that you could potentially use in this case for classification: linear regression.

Simply treat the response as if it were continuous, and find the linear regression coefficients of the response vector $y \in \mathbb{R}^n$ onto the predictors, i.e.,

$$\hat{\beta}_0, \hat{\beta} = \underset{\beta_0 \in \mathbb{R},\, \beta \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2$$

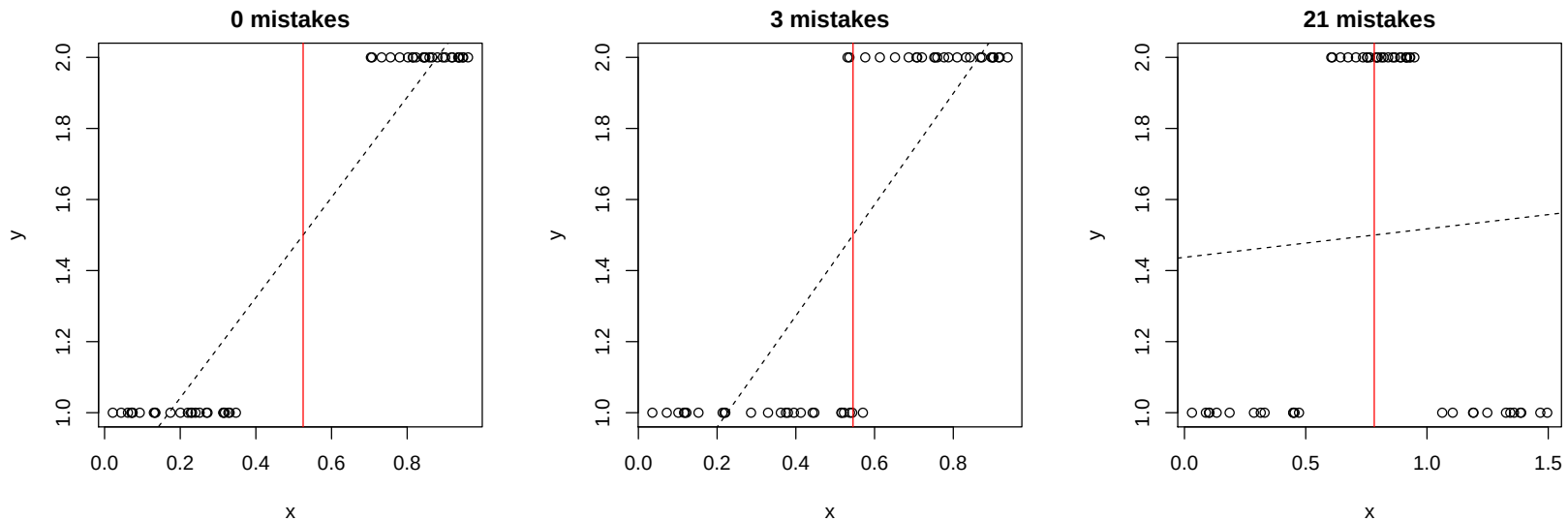
Then, given a new input $x_0 \in \mathbb{R}^p$, we predict the class to be

$$\hat{f}^{\mathrm{LS}}(x_0) = \begin{cases} 1 & \text{if } \hat{\beta}_0 + x_0^T \hat{\beta} \leq 1.5 \\ 2 & \text{if } \hat{\beta}_0 + x_0^T \hat{\beta} > 1.5 \end{cases}$$

Linear regression continued for classification

(Note: since we included an intercept term in the regression, it doesn't matter whether we code the class labels as $\{1, 2\}$ or $\{0, 1\}$, etc.)

In many instances, this actually works reasonably well. Examples:



Overall, using linear regression in this way for binary classification is not a crazy idea. But how about if there are more than 2 classes?

Linear regression (K>2)

Given K classes, define the indicator matrix $Y \in \mathbb{R}^{n \times K}$ to be the matrix whose columns indicate class membership.

That is, its jth column satisfies $Y_{ij} = 1$ if $y_i = j$ (observation i is in class j) and $Y_{ij} = 0$ otherwise.

E.g., with $n = 6$ observations and $K = 3$ classes, the matrix

$$Y = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{6 \times 3}$$

corresponds to having the first two observations in class 1, the next two in class 2, and the final 2 in class 3

Linear regression of indicators

To construct a prediction rule, we regress each column $Y_j \in \mathbb{R}^n$ (indicating the jth class versus all else) onto the predictors:

$$\hat{\beta}_{j,0}, \hat{\beta}_j = \underset{\beta_{j,0} \in \mathbb{R},\, \beta_j \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} (Y_{ij} - \beta_{0,j} - \beta_j^T x_i)^2$$

Now, given a new input $x_0 \in \mathbb{R}^p$, we compute

$$\hat{\beta}_{0,j} + x_0^T \hat{\beta}_j, \quad j = 1, \ldots K$$

take predict the class j that corresponds to the highest score. I.e., we let each of the K linear models make its own prediction, and then we take the strongest. Formally,

$$\hat{f}^{\mathrm{LS}}(x_0) = \underset{j=1,\ldots K}{\arg\max}\ \hat{\beta}_{0,j} + x_0^T \hat{\beta}_j$$

Linear regression for classification

The decision boundary between any two classes j, k are the values of $x \in \mathbb{R}^p$ for which

$$\hat{\beta}_{0,j} + x^T \hat{\beta}_j = \hat{\beta}_{0,k} + x^T \hat{\beta}_k$$

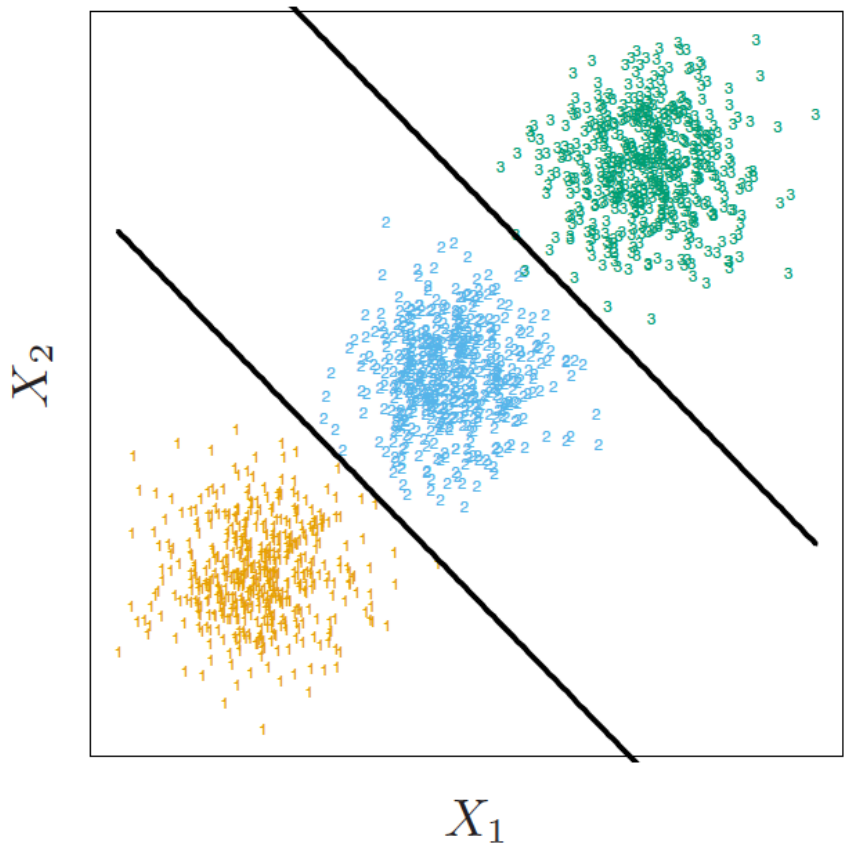
i.e., $\hat{\beta}_{0,j} - \hat{\beta}_{0,k} + (\hat{\beta}_j - \hat{\beta}_k)^T x = 0$

This defines a $(p-1)$-dimensional affine subspace in $\mathbb{R}^p$. To one side, we would always predict class j over k; to the other, we would always predict class k over j

For K classes total, there are $\binom{K}{2} = \frac{K(K-1)}{2}$ decision boundaries

Linear regression for classification

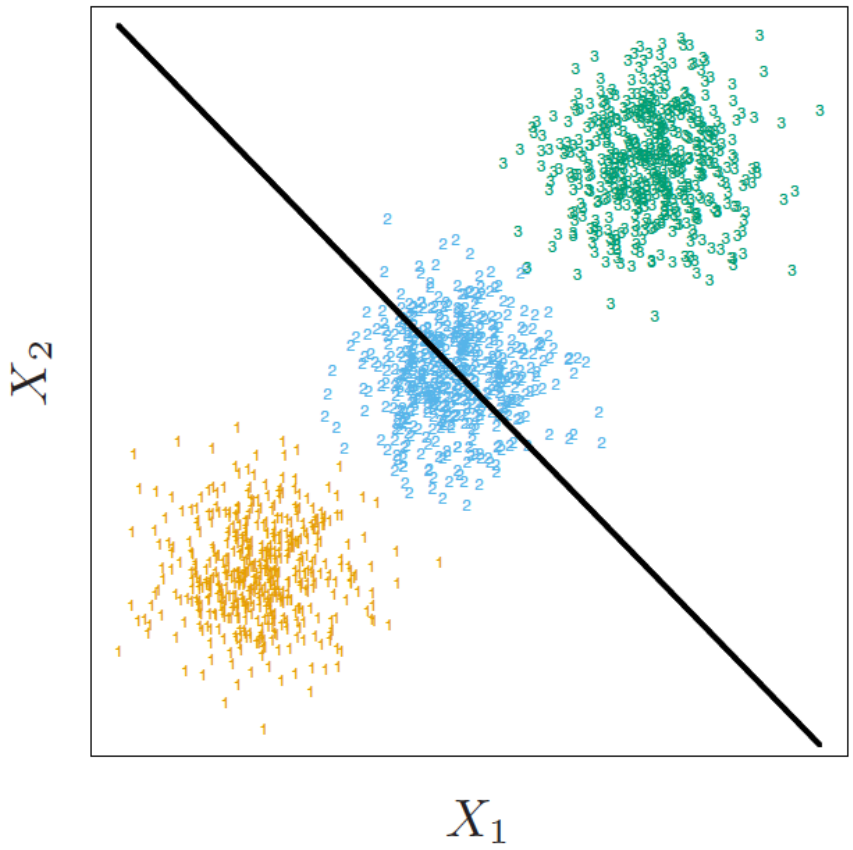
What we'd like to see when we use linear regression for a 3-way classification (from ESL page 105):



The plotted lines are the decision boundaries between classes 1 and 2, and 2 and 3 (the decision boundary between classes 1 and 3 never matters)

Linear regression for classification

What actually happens when we use linear regression for this 3-way classification (from ESL page 105):



The decision boundaries between 1 and 2 and between 2 and 3 are the same, so we would never predict class 2. This problem is called masking (and it is not uncommon for moderate K and small p)

General Setup

Response categories are coded as an indicator variable. Suppose $\mathcal{G}$ has K classes, then $\boldsymbol{Y_1}$ is a vector of 0's and 1's indicating for example whether each observation is in class 1.

- The indicator response matrix is defined as $Y = (\boldsymbol{Y_1}, \ldots, \boldsymbol{Y_K})$.
- Y is a matrix of 0's and 1's with each row having a single 1 indicating an observation is in class k.
- The i^{th} observation of interest has covariate values $\boldsymbol{x_{i1}}, \ldots, \boldsymbol{x_{ip}}$ that will be represented by $X_{N \times p}$.
- Our goal is to predict what class each observation is in given its covariate values.

Naive method

- Let's proceed blindly and use a naive method of linear regression.
- Fit a linear regression to each column of Y.
- The coefficient matrix is $\hat{B} = (X'X)^{-1}X'Y$.
- $\hat{Y} = X(X'X)^{-1}X'Y$
- The k^{th} column of $\hat{B}$ contains the estimates corresponding to the linear regression coefficients that we get from regressing $\boldsymbol{X_1}, \ldots, \boldsymbol{X_p}$ onto $\boldsymbol{Y_K}$.

Linear regression of an indicator matrix

Look at $\hat{Y}$ corresponding to the indicator variable for each class k. Assign each observation to the class for which $\hat{Y}$ is the largest.

More formally stated, a new observation with covariate $\boldsymbol{x}$ is classified as follows:

- Compute the fitted output $\hat{\boldsymbol{Y}}_{\boldsymbol{new}}(\boldsymbol{x}) = [(1, \boldsymbol{x})^T \hat{B}]^T$.
- Identify the largest component of $\hat{\boldsymbol{Y}}_{\boldsymbol{new}}(\boldsymbol{x})$ and classify according to

$$\hat{G}(\boldsymbol{x}) = \arg\max_k \hat{\boldsymbol{Y}}_{\boldsymbol{new}}(\boldsymbol{x}).$$

Does this approach make sense?

- The regression line estimates $\mathrm{E}(Y_k|\boldsymbol{X} = \boldsymbol{x}) = \mathrm{P}(G = k|\boldsymbol{X} = \boldsymbol{x})$ so the method seems somewhat sensible at first.
- Although $\sum_k \hat{Y}_k(\boldsymbol{x}) = 1$ for any $\boldsymbol{x}$ as long as there is an intercept in the model (exercise), $\hat{Y}_k(\boldsymbol{x})$ can be negative or greater than 1 which is nonsensical to the initial problem statement.
- Worse problems can occur when classes are masked by others due to the rigid nature of the regression model.

LDA and QDA

- How do we fix the issues with regression?
- We could do logististic regression, but it doesn't work for more than 2 classes.
- We'll now introduce linear and quadratic discrimminant analysis.
- You'll look at the details of this more in your homework.

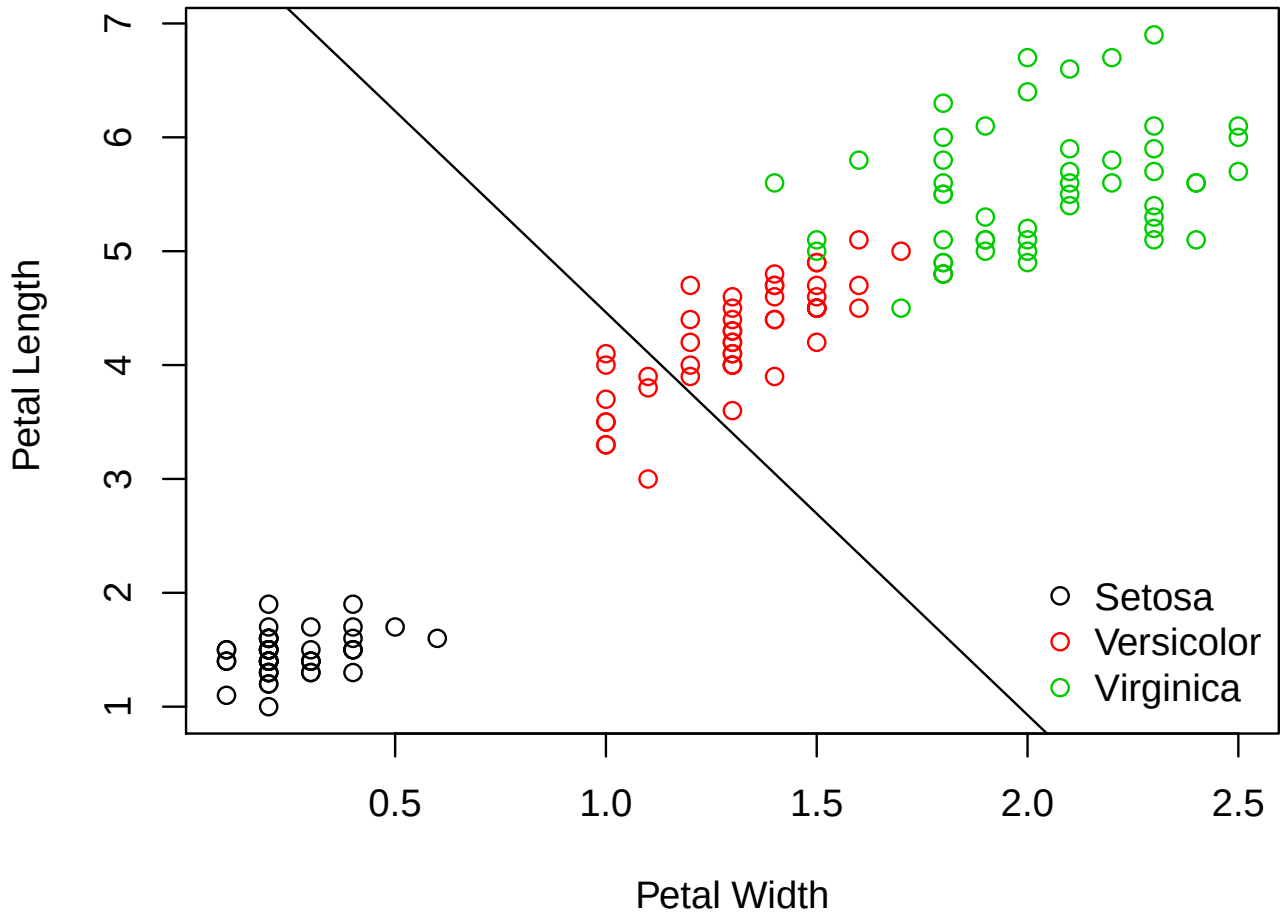
Iris Data

- This data set (Fisher, Annals of Eugenics,1936) gives the measurements of sepal and petal length and width for 150 flowers using 3 species of iris (50 flowers per species).
- The species considered are setosa, versicolor, and virginica.
- To best illustrate the methods of classification, we considered how petal width and length predict the species of a flower.

Iris Data

Sepal L	Sepal W	Petal L	Petal W	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
⋮	⋮	⋮	⋮	⋮
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
⋮	⋮	⋮	⋮	⋮
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica

Illustration of Masking



Why does this illustrate masking?

To recap the previous picture, we can see that using linear regression to predict for different classes can lead to a masking effect of one group or more. This occurs for the following reasons:

1. There is a plane that is high in the bottom left corner (setosa) and low in the top right corner (virginica).
2. There is a second plane that is high in the top right corner (virginica) but low in the bottom left corner (setosa).
3. The third plane is approximately flat since it tries to linearly fit a collection of points that is high in the middle (versicolor) and low on both ends.

Linear discriminant analysis

For each observation, conditional on them being in class k, we assume $\boldsymbol{X}|G = k \sim N_p(\boldsymbol{\mu_k}, \Sigma_k)$. That is,

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_k})'\Sigma_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu_k})\right\}.$$

Linear Discriminant Analysis (LDA) assumes $\Sigma_k = \Sigma$ for all k.

Linear discriminant analysis

In practice the parameters of the Gaussian distribution are unknown and must be estimated by:

- $\hat{\pi_k} = N_k/N$, where N_k is the number of observations in class k
- $\hat{\boldsymbol{\mu}}_{\boldsymbol{k}} = \sum_{i:g_i=k} \boldsymbol{x_i}/N_k$
- $\hat{\boldsymbol{\Sigma}} = \sum_{k=1}^{K} \sum_{i:g_i=k} (\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}_{\boldsymbol{k}})(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}}_{\boldsymbol{k}})'/(N-K)$,

where $\pi_k = P(G = k)$.

Derivations

We're interested in computing

$$P(G = k|\boldsymbol{X} = \boldsymbol{x}) = \frac{P(G = k, \boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{X} = \boldsymbol{x})}$$

$$= \frac{P(\boldsymbol{X} = \boldsymbol{x}|G = k)P(G = k)}{\sum_{k=1}^{K} P(\boldsymbol{X} = \boldsymbol{x}, G = k)}$$

$$= \frac{f_k(\boldsymbol{x})\pi_k}{\sum_{j=1}^{K} f_j(\boldsymbol{x})\pi_j}.$$

Derivations

We will compute $P(G = k|\boldsymbol{X} = \boldsymbol{x})$ for each class k.

Consider comparing $P(G = k_1|\boldsymbol{X} = \boldsymbol{x})$ and $P(G = k_2|\boldsymbol{X} = \boldsymbol{x})$.

Then

$$\log\left[\frac{P(G = k_1|\boldsymbol{X} = \boldsymbol{x})}{P(G = k_2|\boldsymbol{X} = \boldsymbol{x})}\right] = \log\left[\frac{f_{k_1}(\boldsymbol{x})\pi_{k_1}}{f_{k_2}(\boldsymbol{x})\pi_{k_2}}\right]$$

$$= -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_{k_1}})'\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu_{k_1}}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_{k_2}})'\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu_{k_2}}) + \log\left[\frac{\pi_{k_1}}{\pi_{k_2}}\right]$$

$$= (\boldsymbol{\mu_{k_1}} - \boldsymbol{\mu_{k_2}})'\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}'_{\boldsymbol{k_1}}\Sigma^{-1}\boldsymbol{\mu_{k_1}} + \frac{1}{2}\boldsymbol{\mu}'_{\boldsymbol{k_2}}\Sigma^{-1}\boldsymbol{\mu_{k_2}} + \log\left[\frac{\pi_{k_1}}{\pi_{k_2}}\right]$$

Derivations

Now let's consider the boundary between predicting someone to be in class k_1 or class k_2. To be on the the boundary, we must decide what $\boldsymbol{x}$ would need to be if we think that an observation is equally likely to be in class k_1 or k_2.

This reduces to solving

$$(\boldsymbol{\mu_{k_1}} - \boldsymbol{\mu_{k_2}})'\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}'_{\boldsymbol{k_1}}\Sigma^{-1}\boldsymbol{\mu_{k_1}} + \frac{1}{2}\boldsymbol{\mu}'_{\boldsymbol{k_2}}\Sigma^{-1}\boldsymbol{\mu_{k_2}} + \log\left[\frac{\pi_{k_1}}{\pi_{k_2}}\right] = 0,$$

which is linear in $\boldsymbol{x}$.

- The boundary will be a line for two dimensional problems.
- The boundary will be a hyperplane for three dimensional problems.

Derivations

The linear log-odds function implies that our decision boundary between classes k_1 and k_2 will be the set where

$$P(G = k_1|\boldsymbol{X} = \boldsymbol{x}) = P(G = k_2|\boldsymbol{X} = \boldsymbol{x}),$$

which is linear in $\boldsymbol{x}$. In p dimensions, this is a hyperplane.

We can then say that class k_1 is more likely than class k_2 if

$$P(G = k_1|\boldsymbol{X} = \boldsymbol{x}) > P(G = k_2|\boldsymbol{X} = \boldsymbol{x}) \implies$$

$$\log\left[\frac{P(G = k_1|\boldsymbol{X} = \boldsymbol{x})}{P(G = k_2|\boldsymbol{X} = \boldsymbol{x})}\right] > 0 \implies$$

Derivations

$$(\boldsymbol{\mu_{k_1}} - \boldsymbol{\mu_{k_2}})' \Sigma^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}'_{\boldsymbol{k_1}} \Sigma^{-1} \boldsymbol{\mu_{k_1}} + \frac{1}{2} \boldsymbol{\mu}'_{\boldsymbol{k_2}} \Sigma^{-1} \boldsymbol{\mu_{k_2}} + \log \left[\frac{\pi_{k_1}}{\pi_{k_2}} \right] > 0 \implies$$

$$\boldsymbol{\mu}'_{\boldsymbol{k_1}} \Sigma^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}'_{\boldsymbol{k_1}} \Sigma^{-1} \boldsymbol{\mu_{k_1}} + \log(\pi_{k_1}) > \boldsymbol{\mu}'_{\boldsymbol{k_2}} \Sigma^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}'_{\boldsymbol{k_2}} \Sigma^{-1} \boldsymbol{\mu_{k_2}} + \log(\pi_{k_2}).$$

The linear discriminant function $\delta_k^L(\boldsymbol{x})$ is defined as

$$\delta_k^L(\boldsymbol{x}) = \boldsymbol{\mu}'_{\boldsymbol{k}} \Sigma^{-1} \boldsymbol{x} - \boldsymbol{\mu}'_{\boldsymbol{k}} \Sigma^{-1} \boldsymbol{\mu_k} + \log(\pi_k).$$

We can tell which class is more likely for a particular value of $\boldsymbol{x}$ by comparing the classes' linear discriminant functions.

QDA

- If the Σ_k are not assumed to be equal, then convenient cancellations in our derivations earlier do not occur.
- The quadratic pieces in $\boldsymbol{x}$ end up remaining leading to quadratic discriminant functions (QDA).
- QDA is similar to LDA except a covariance matrix must be estimated for each class k.

The quadratic discriminant function $\delta_k^Q(\boldsymbol{x})$ is defined as

$$\delta_k^Q(\boldsymbol{x}) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_k})'\Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu_k}) + \log(\pi_k).$$

Properties of LDA and QDA

LDA and QDA seem to be widely accepted due to a bias variance trade off that leads to stability of the models.

That is, we want our model to have low variance, so we are willing to sacrifice some bias of a linear decision boundary in order for our model to be more stable.

Regularized Discriminant Analysis

- Friedman (1989) proposed a compromise between LDA and QDA.
- This method says that we should shrink the covariance matrices of QDA toward a common covariance matrix as done in LDA.
- Regularized covariance matrices take the form

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}, \quad 0 \leq \alpha \leq 1.$$

- In practice, α is chosen based on performance of the model on validation data or by using cross-validation.

Application to Weekly Data

We're going to work with the Weekly data set that is part of the ISLR package.

It contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

Task 1

Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

Solution to Task 1

```
library(ISLR)
data(Weekly)
names(Weekly)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

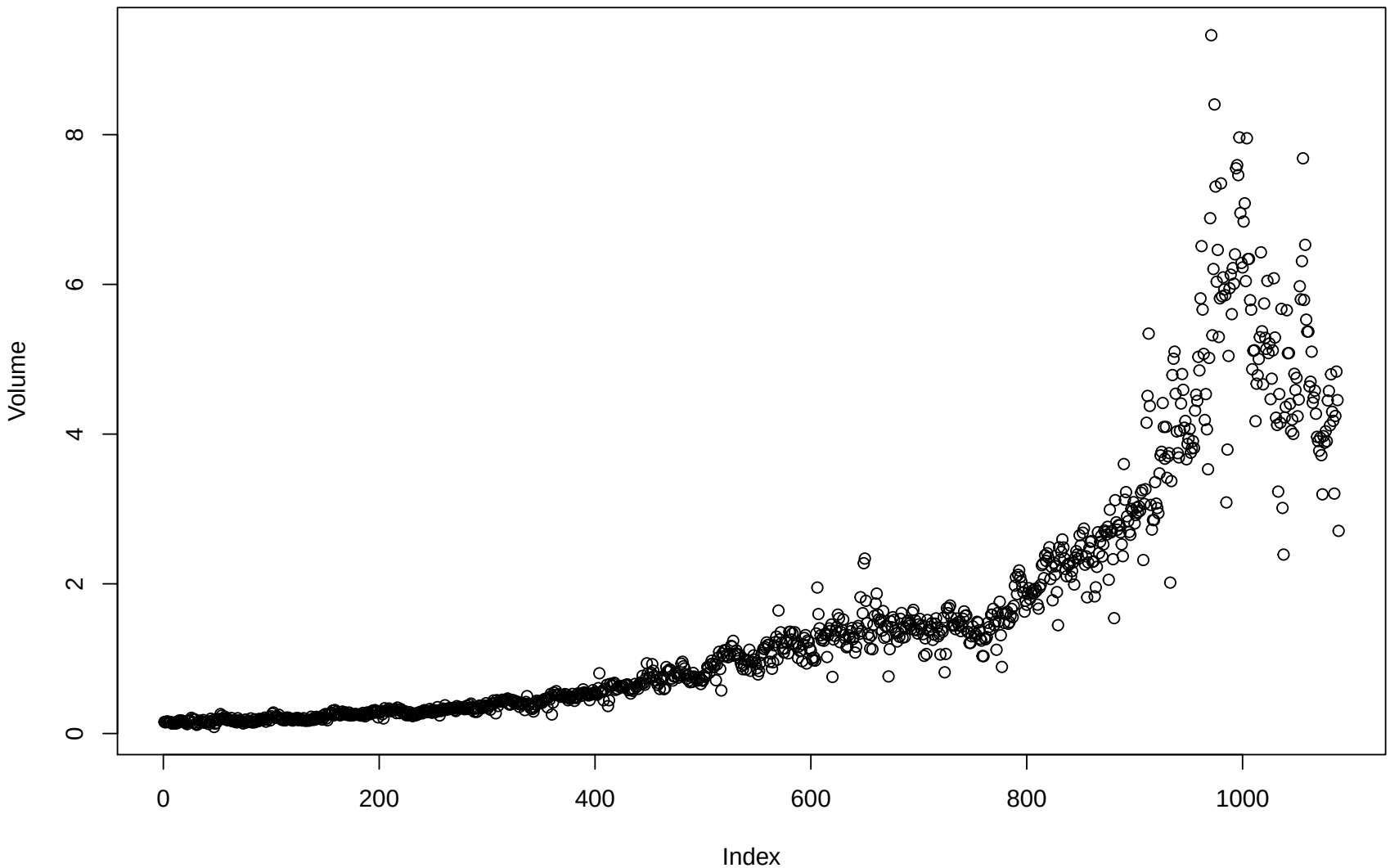
```
summary(Weekly)
```

```
##       Year           Lag1                Lag2                Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4                Lag5              Volume
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
##  Median :  0.2380   Median :  0.2340   Median :1.00268
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
##      Today          Direction
##  Min.   :-18.1950   Down:484
##  1st Qu.: -1.1540   Up  :605
##  Median :  0.2410
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
```

```
attach(Weekly)
```

Solution to Task 1

```
plot(Volume)
```



Yes, there is a pattern of increasing volume over time.

Task 2

Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

Solution to Task 2

```
glm.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 +
        Lag5 + Volume, family="binomial", data=Weekly)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = "binomial", data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Solution to Task 2

From the analysis on the previous slide, Lag 2 is statistically significant with a p-value of 0.03. The postive coefficient for this predictor suggests that if the market had a positive return yesterday, then it is more likely to go up today. Since the p-value is small, the is clear evidence of an association between Lag 2 and Direction.

Task 3

Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.[1] Calculate the false positive and false negative rates here.

[1] A confusion matrix is explained on p. 145 of the James et. al (2013) text. See Table 4.5.

Solution to Task 3

Recall back in the LSH module we looked at classifications as being:

1. Pairs of data can be linked in both the handmatched training data (which we refer to as 'truth') and under the estimated linked data. We refer to this situation as true positives (TP).
2. Pairs of data can be linked under the truth but not linked under the estimate, which are called false negatives (FN).
3. Pairs of data can be not linked under the truth but linked under the estimate, which are called false positives (FP).
4. Pairs of data can be not linked under the truth and also not linked under the estimate, which we refer to as true negatives (TN).

Table 1: Confusion Matrix on Full Data for Direction

Pred (Estimated)		True	False
Truth	True	True positive	False negative
	False	False positive	True negative

Solution to Task 3

You can now calculate the FNR and FPR using the following formulas:

$$FPR = \frac{\text{false positives}}{(\text{false positives} + \text{true negatives})}.$$

Also,

$$FNR = \frac{\text{false negatives}}{(\text{false negatives} + \text{true positives})}.$$

Give the confusion matrix and calculate the FNR and FPR.

Task 4

Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010). Is there an improvement and if so in what sense? (Think about how the false positive and false negative rates have changed).

Task 5

Repeat Task 4 using LDA.

Use the function lda() in R.

Task 6

Repeat Task 4 using QDA. Which method is better: logistic regression, LDA, or QDA. Make sure you can defend your answer.

Use the function qda() in `R`.

Summary

As we remarked earlier, both LDA and logistic regression model the log odds as a linear function of the predictors $x \in \mathbb{R}^p$

$$\text{Linear discriminant analysis:} \quad \log\left\{\frac{\mathcal{P}(C=1|X=x)}{\mathcal{P}(C=2|X=x)}\right\} = \alpha_0 + \alpha^T x$$

$$\text{Logistic regression:} \quad \log\left\{\frac{\mathcal{P}(C=1|X=x)}{\mathcal{P}(C=2|X=x)}\right\} = \beta_0 + \beta^T x$$

where for LDA we form $\hat{\alpha}_0, \hat{\alpha}$ based on estimates $\hat{\pi}_j, \hat{\mu}_j, \hat{\Sigma}$ (easy!), and for logistic regression we estimate $\hat{\beta}_0, \hat{\beta}$ directly based on maximum likelihood (harder)

This is what leads to linear decision boundaries for each method

Careful inspection (or simply comparing them in R) shows that the estimates $\hat{\alpha}_0, \hat{\beta}_0$ and $\hat{\alpha}, \hat{\beta}$ are different. So how do they compare?

Summary

Generally speaking, logistic regression is more flexible because it doesn't assume anything about the distribution of X. LDA assumes that X is normally distributed within each class, so that its marginal distribution is a mixture of normal distributions, hence still normal:

$$X \sim \sum_{j=1}^{K} \pi_j N(\mu_j, \Sigma)$$

This means that logistic regression is more robust to situations in which the class conditional densities are not normal (and outliers)

On the other side, if the true class conditional densities are normal, or close to it, LDA will be more efficient, meaning that for logistic regression to perform comparably it will need more data

In practice they tend to perform similarly in a variety of situations (as claimed by the ESL book on page 128)

Model free classification

We could instead take a model free classification approach.

The downside: these methods are essentially a black box for classification, in that they typically don't provide any insight into how the predictors and the response are related

The upside: they can work well for prediction in a wide variety of situations, since they don't make any real assumptions

These procedures also typically have tuning parameters that need to be properly tuned in order for them to work well (for this we can use cross-validation)

Model free classification

1. K-nearest neighbors
2. Prototype classification
3. K-means classification

We won't have time to cover these, but you could read about these more on your own if you want to know about model-free (non-parametric approaches).

Some of these are briefly covered in ISLR, but not in great detail.