

Models for classification

Logistic regression

A quick recap

During the last lectures, we did...

► Models for classification.

- Introduce the classification problem
- Introduce logistic regression models



Today's goal

Today, we going to do...

► Models for classification.

- Discuss a case study
- Implement in R

Reading list

-  Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*, Springer (2014)
-  Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*, Springer (2017)¹

¹The book is available for download at:
<http://www-bcf.usc.edu/~gareth/ISL/>

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Goal and motivation

- Predict whether the index will increase or decrease on a given day using the past 5 days' percentage changes in the index.
- It is a classification problem: Predicting whether a given day's stock market performance will fall into the *Up* bucket or the *Down* bucket.
- A model that could accurately predict the direction in which the market will move would be very useful!

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

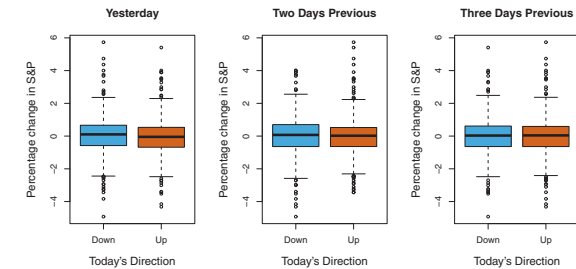
The data: 1250 observations on 9 variables

- **Year:** The year that the observation was recorded
- **Lag1:** Percentage return for previous day
- **Lag2:** Percentage return for 2 days previous
- **Lag3:** Percentage return for 3 days previous
- **Lag4:** Percentage return for 4 days previous
- **Lag5:** Percentage return for 5 days previous
- **Volume:** Volume of shares traded (number of daily shares traded in billions)
- **Today:** Percentage return for today
- **Direction:** A factor with levels `Down` and `Up` indicating whether the market had a positive or negative return on a given day

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

The data: 1250 observations on 9 variables

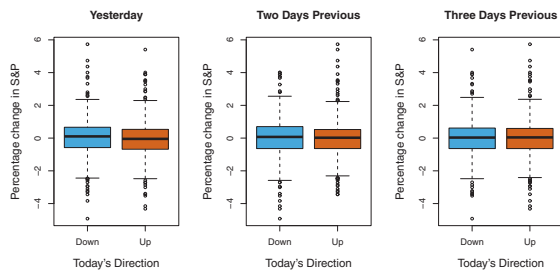


- Boxplots of the previous day's percentage change in the S&P index for the days for which the market increased or decreased.
- The two plots look almost identical: there is no simple strategy for using the previous days movement in the S&P to predict today's returns.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

The data: 1250 observations on 9 variables

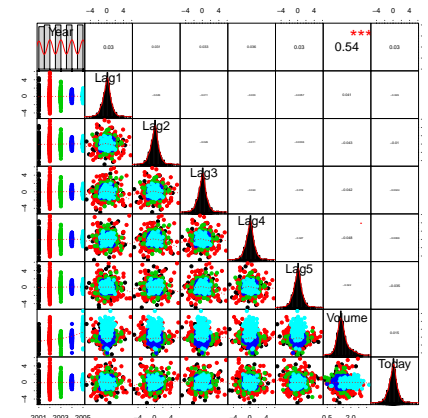


- This lack of pattern is to be expected: in the presence of strong correlations between successive days' returns, one could adopt a simple trading strategy to generate profits from the market.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

The data: 1250 observations on 9 variables

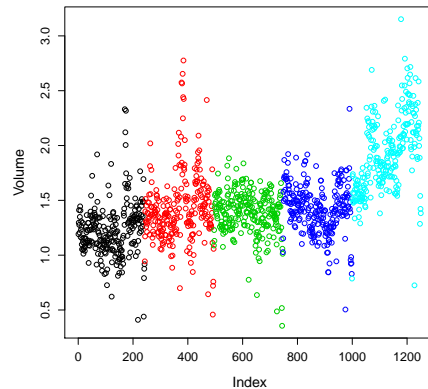


- The correlations between the lag variables and today's returns are close to zero.
- There appears to be little correlation between today's returns and previous days' returns.
- The only substantial correlation is between `Year` and `Volume`.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

The data: 1250 observations on 9 variables



- ▶ By plotting the data we see that Volume is increasing over time.
- ▶ In other words, the average number of shares traded daily increased from 2001 to 2005.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume

- ▶ `glm()` fits generalized linear models and the argument `family=binomial` tells R to run a logistic regression.

```
glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
             data=Smarket, family=binomial)

coef(glm.fits) # Model parameters

(Intercept)   Lag1    Lag2    Lag3    Lag4    Lag5   Volume
   -0.126   -0.073   -0.042    0.011    0.009    0.010    0.135

summary(glm.fits)$coef[,4] # P-values for the model parameters
(Intercept)   Lag1    Lag2    Lag3    Lag4    Lag5   Volume
   0.60     0.15     0.40     0.82     0.85     0.83    0.39
```

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume

- ▶ `glm()` fits generalized linear models and the argument `family=binomial` tells R to run a logistic regression.
- ▶ The smallest p-value here is associated with Lag1.
- ▶ The negative coefficient for this predictor suggests that if the market had a positive return yesterday, then it is less likely to go up today.
- ▶ However, at a value of 0.15, the p-value is still relatively large, and so there is no clear evidence of a real association between Lag1 and Direction.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- ▶ `predict()` function can be used to predict the probability that the market goes up.
- ▶ `type="response"` tells R to output probabilities of the form $P(Y = 1|X)$, as opposed to other information such as the logit.
- ▶ If no data set is supplied, then the probabilities are computed for the training data that was used to fit the logistic regression model.

```
glm.probs=predict(glm.fits,type="response")

glm.probs[1:10] # Print only the first 10 prob
 1    2    3    4    5    6    7    8    9   10
0.51 0.48 0.48 0.52 0.51 0.51 0.49 0.51 0.52 0.49
```

- ▶ These values correspond to the market probability going up, rather than down.
- ▶ `contrasts()` gives the contrasts associated with a factor.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- `predict()` function can be used to predict the probability that the market goes up.
- `type="response"` tells R to output probabilities of the form $P(Y = 1|X)$, as opposed to other information such as the logit.
- If no data set is supplied, then the probabilities are computed for the training data that was used to fit the logistic regression model.

```
contrasts(Direction)
      Up
Down  0
Up    1
```

- With `contrasts()`, R has created a dummy variable with a 1 for Up.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- `predict()` function can be used to predict the probability that the market goes up.
- `type="response"` tells R to output probabilities of the form $P(Y = 1|X)$, as opposed to other information such as the logit.
- If no data set is supplied, then the probabilities are computed for the training data that was used to fit the logistic regression model.

```
contrasts(Direction)
      Up
Down  0
Up    1
```

- With `contrasts()`, R has created a dummy variable with a 1 for Up.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- To make a prediction as to whether the market will go up or down on a particular day, we must convert these predicted probabilities into class labels, Up or Down.

```
plim <- 0.5
glm.pred = rep("Down", length(Direction))
glm.pred[glm.probs > plim] = "Up"
head(glm.pred)
[1] "Up" "Down" "Down" "Up" "Up" "Up"
```

- We transform to Up all of the elements for which the predicted probability of a market increase exceeds 0.5.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- To make a prediction as to whether the market will go up or down on a particular day, we must convert these predicted probabilities into class labels, Up or Down.

```
table(glm.pred, Direction)
      Direction
glm.pred Down Up
Down   145 141
Up     457 507

mean(glm.pred == Direction)
[1] 0.52
```

- The diagonal elements of the confusion matrix indicate correct predictions, while the off-diagonals represent incorrect predictions.
- The model correctly predicted that the market would go up on 507 days and that it would go down on 145 days, for a total of $507 + 145 = 652$ correct predictions.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- To make a prediction as to whether the market will go up or down on a particular day, we must convert these predicted probabilities into class labels, Up or Down.

```
mean(glm.pred==Direction)
[1] 0.52
```

- We compute the fraction of days for which the prediction was correct.
- Logistic regression correctly predicted the market movement 52% of the time.
→ Only a little better than a random guess!

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- This result is misleading: we trained and tested the model on the same set of 1250 observations.
- In other words, $100 - 52.2 = 47.8\%$ is the training error rate.
- We know that the training error rate is often overly optimistic: it tends to underestimate the test error rate.
- In order to better assess the accuracy of the logistic regression model in this setting, we can fit the model using part of the data, and then examine how well it predicts the held out data.
- This will yield a more realistic error rate, in the sense that in practice we will be interested in our model's performance not on the data that we used to fit the model, but rather on days in the future for which the market's movements are unknown.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- We split the data in training and test.
- We first create a vector corresponding to the observations from 2001 through 2004.
- We will then use this vector to create a held out data set of observations from 2005.

```
train=(Year<2005)
Smarket.2005=Smarket[!train,]
dim(Smarket.2005)

Direction.2005=Direction[!train]
```

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- We now fit a logistic regression model using only the subset of the observations that correspond to dates before 2005.
- We then obtain predicted probabilities of the stock market going up for each of the days in our test set - that is, for the days in 2005.

```
glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,↵
family=binomial,subset=train)

glm.probs=predict(glm.fits,Smarket.2005,type="response")
```

- The training and test are completely different data sets.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- We compute the predictions for 2005.
- We compare them to the actual movements of the market over that training period.

```
glm.pred=rep("Down", nrow(Smarket.2005))
glm.pred[glm.probs>plim]="Up"
table(glm.pred,Direction.2005)
      Direction.2005
glm.pred Down Up
Down      77  97
Up        34  44
mean(glm.pred==Direction.2005)
[1] 0.48
mean(glm.pred !=Direction.2005)
[1] 0.52
```

- One would not generally expect to be able to predict previous days' returns to predict future market performance.

Case study: The stock market data

We examine a stock market data set that contains the daily movements in the Standard & Poor's 500 (S&P) stock index over a 5-year period between 2001 and 2005.

Logistic regression in R: we fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.

- We try to remove the variables that appear not to be helpful in predicting Direction and we obtain a more effective model.

```
glm.fits=glm(Direction~Lag1+Lag2,data=Smarket ,family=binomial, <-
             subset=train)
glm.probs=predict(glm.fits,Smarket.2005,type="response")

glm.pred[glm.probs>plim]="Up"
table(glm.pred,Direction.2005)
      Direction.2005
glm.pred Down Up
Down      35  35
Up        76 106
mean(glm.pred==Direction.2005)
[1] 0.5595238
```