chapter 5

Resampling Methods

03/10/2024

```
library(boot)
```

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.3.3
```

#chp 5(6) 6. We continue to consider the use of a logistic regression model to predict the probability of default using income and balance on the Default data set. In particular, we will now compute estimates for the standard errors of the income and balance logistic regression co efficients in two different ways: (1) using the bootstrap, and (2) using the standard formula for computing the standard errors in the glm() function. Do not forget to set a random seed before beginning your analysis.

a. Using the summary() and glm() functions, determine the estimated standard errors for the coefficients associated with income and balance in a multiple logistic regression model that uses both predictors.

```
fit14=glm(default~income+balance , data = Default , family = binomial)
summary(fit14)$coefficients[, 2]
```

```
##  (Intercept)       income      balance 
## 4.347564e-01 4.985167e-06 2.273731e-04
```

INTERPRETATION #The coefficient for income, approximately 0.000004985, that the each one-unit increase in income, the dependent variable is expected to increase by 0.000004985 units, indicating a very small positive relationship between income and the outcome #the coefficient for balance, about 0.0002273731, implies that a one-unit increase in balance results in an expected increase of 0.0002273731 units in the dependent variable (b) Write a function, boot.fn(), that takes as input the Default dataset as well as an index of the observations, and that outputs the coefficient estimates for income and balance in the multiple logistic regression model.

```
boot.fn <- function(x, i) {
  fit14 <- glm(default ~ income + balance, data = x[i, ], family = "binomial")
  coef(fit14)[-1]
}
```

INTERPRETATION #the change in the log-odds of default for each unit increase in income, holding balance constant #a positive coefficient indicates that higher balances are associated with a higher likelihood of default

c. Use the boot() function together with your boot.fn() function to estimate the standard errors of the logistic regression coefficients for income and balance.

```
boot(boot.fn, data=Default, 100)
```

```
## 
## ORDINARY NONPARAMETRIC BOOTSTRAP
## 
## 
## Call:
## boot(data = Default, statistic = boot.fn, R = 100)
## 
## 
## Bootstrap Statistics :
##         original        bias     std. error
## t1* 2.080898e-05 -4.847706e-07 4.403669e-06
## t2* 5.647103e-03 -4.753193e-06 1.917629e-04
```

INTERPRETATION #it involves resampling the Default dataset 100 times with replacement (d) Comment on the estimated standard errors obtained using the glm() function and using your bootstrap function. boot() Method:

glm() Method:

```
summary(fit14)$coefficients[2:3, 2]
```

```
##       income      balance 
## 4.985167e-06 2.273731e-04
```

INTERPRETATION #a loan increase by approximately (4.985167*10^-6) when holding the balance constant #a loan increase by approximately 2.273731*10^-4 when holding income constant #it indicates a positive relationship between balance and the likelihood of default

#chp 5(7) 7. In Sections 5.3.2 and 5.3.3, we saw that the cv.glm() function can be used in order to compute the LOOCV test error estimate. Alternatively, one could compute those quantities using just the glm() and 200 predict.glm() functions, and a for loop. You will now take this approach in order to compute the LOOCV error for a simple logistic regression model on the Weekly data set. Recall that in the context of classification problems, the LOOCV error is given in (5.4).

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
glimpse(Weekly)
```

```
## Rows: 1,089
## Columns: 9
## $ Year      <dbl> 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, …
## $ Lag1      <dbl> 0.816, -0.270, -2.576, 3.514, 0.712, 1.178, -1.372, 0.807, 0…
## $ Lag2      <dbl> 1.572, 0.816, -0.270, -2.576, 3.514, 0.712, 1.178, -1.372, 0…
## $ Lag3      <dbl> -3.936, 1.572, 0.816, -0.270, -2.576, 3.514, 0.712, 1.178, -…
## $ Lag4      <dbl> -0.229, -3.936, 1.572, 0.816, -0.270, -2.576, 3.514, 0.712, …
## $ Lag5      <dbl> -3.484, -0.229, -3.936, 1.572, 0.816, -0.270, -2.576, 3.514,…
## $ Volume    <dbl> 0.1549760, 0.1485740, 0.1598375, 0.1616300, 0.1537280, 0.154…
## $ Today     <dbl> -0.270, -2.576, 3.514, 0.712, 1.178, -1.372, 0.807, 0.041, 1…
## $ Direction <fct> Down, Down, Up, Up, Up, Down, Up, Up, Up, Down, Down, Up, Up…
```

a. Fit a logistic regression model that predicts Direction using Lag1 and Lag2.

```
log_dir = glm(Direction ~ Lag1 + Lag2, data = Weekly, family = "binomial")
summary(log_dir)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = "binomial", data = Weekly)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22122    0.06147   3.599 0.000319 ***
## Lag1        -0.03872    0.02622  -1.477 0.139672
## Lag2         0.06025    0.02655   2.270 0.023232 *
## ---
## 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: 1488.2  on 1086  degrees of freedom
## AIC: 1494.2
##
## Number of Fisher Scoring iterations: 4
```

INTERPRETATION #The coefficient for Lag1 is -0.03872, meaning that for each unit increase in Lag1 this is not statistically significant (p-value = 0.139672) #The model’s null deviance is 1496.2, and the residual deviance is 1488.2, indicating a modest improvement in fit when including the lag variables (b) Fit a logistic regression model that predicts Direction using Lag1 and Lag2 using all but the first observation.

```
log_dir_2 <- glm(Direction ~ Lag1 + Lag2, data = Weekly[-1, ], family = "binomial")
summary(log_dir_2)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = "binomial", data = Weekly[-1,
##     ])
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22324    0.06150   3.630 0.000283 ***
## Lag1        -0.03843    0.02622  -1.466 0.142683
## Lag2         0.06085    0.02656   2.291 0.021971 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1494.6  on 1087  degrees of freedom
## Residual deviance: 1486.5  on 1085  degrees of freedom
## AIC: 1492.5
##
## Number of Fisher Scoring iterations: 4
```

INTERPRETATION # Lag2 has a positive coefficient of 0.06085 #This relationship is statistically significant (p-value = 0.021971)

c. Use the model from (b) to predict the direction of the first observation. You can do this by predicting that the first observation will go up if P(Direction="Up"|Lag1, Lag2) > 0.5.Was this observation correctly classified?

```
predict(log_dir_2, newdata = Weekly[1, , drop = FALSE], type = "response") > 0.5
```

```
##    1
## TRUE
```

####Yes the observation was correctly classified.

d. Write a for loop from i =1toi = n,wheren is the number of observations in the data set, that performs each of the following steps:i. Fit a logistic regression model using all but the ith observation to predict Direction using Lag1 and Lag2.

ii. Compute the posterior probability of the market moving up for the ith observation.
iii. Use the posterior probability for the ith observation in orderto predict whether or not the market moves up.
iv. Determine whether or not an error was made in predictingthe direction for the ith observation. If an error was made,then indicate this as a 1, and otherwise indicate it as a 0.

```
error <- c()

for (i in 1:nrow(Weekly)) {
  log_dir <- glm(Direction ~ Lag1 + Lag2, data = Weekly[-i, ], family = "binomial") # i.

  prediction <- ifelse(predict(log_dir, newdata = Weekly[i, ], type = "response") > 0.5, "Up", "Down") # ii. & ii
i.

  error[i] <- as.numeric(prediction != Weekly[i, "Direction"]) # iv.
}

error[1:10]
```

```
##  [1] 1 1 0 1 0 1 0 0 0 1
```

INTERPRETATION #storing a 1 for incorrect predictions and a 0 for correct ones in the error vector #probability exceeds 0.5, the prediction is classified as "Up"; otherwise, it is classified as "Down." (e) Take the average of the n numbers obtained in (d)iv in order to obtain the LOOCV estimate for the test error.Comment on the results.

```
mean(error)
```

```
## [1] 0.4499541
```

The LOOCV estimate for the test error is ≈0.45.

```
prop.table(table(Weekly$Direction))
```

```
##
##      Down        Up
## 0.4444444 0.5555556
```