Homework 5 Solution

Yi Han

ISLR ch.5: 5,6,9

Format [10pt]

CHR 5.5 [15pt]

In Chapter 4, we used logistic regression to predict the probability of default using income and balance on the Default data set. We will now estimate the test error of this logistic regression model using the validation set approach. Do not forget to set a random seed before beginning your analysis.

(a) [3pt]

Fit a logistic regression model that uses income and balance to predict default.

Solution

```
library(ISLR)
summary(Default)
```

```
##  default    student       balance           income     
##  No :9667   No :7056   Min.   :   0.0   Min.   :  772  
##  Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340  
##                        Median : 823.6   Median :34553  
##                        Mean   : 835.4   Mean   :33517  
##                        3rd Qu.:1166.3   3rd Qu.:43808  
##                        Max.   :2654.3   Max.   :73554  
```

```
set.seed(1)
glm.fit <-  glm(default ~ income + balance, data = Default, family = binomial)
summary(glm.fit)
```

```
## 
## Call:
## glm(formula = default ~ income + balance, family = binomial, 
##     data = Default)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245  
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
```

```
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8
```

(b) [4pt]

Using the validation set approach, estimate the test error of this model. In order to do this, you must perform the following steps: i. Split the sample set into a training set and a validation set. ii. Fit a multiple logistic regression model using only the training observations. iii. Obtain a prediction of default status for each individual in the validation set by computing the posterior probability of default for that individual, and classifying the individual to the default category if the posterior probability is greater than 0.5. iv. Compute the validation set error, which is the fraction of the observations in the validation set that are misclassified.

Solution

```r
# i.
set.seed(1)
train <- sample(nrow(Default), nrow(Default)/2)
# ii.
glm.fit <- glm(default ~ income + balance, data = Default, family = binomial, subset = train)
# iii.
glm.pred <- rep("No", nrow(Default)/2)
glm.probs <- predict(glm.fit, Default[-train, ], type = "response")
glm.pred[glm.probs > 0.5] <- "Yes"
# iv.
mean(glm.pred != Default[-train, ]$default)
```

```
## [1] 0.0254
```

The fraction of the observations in the validation set that are misclassified is 0.0286.

(c) [4pt]

Repeat the process in (b) three times, using three different splits of the observations into a training set and a validation set. Comment on the results obtained.

Solution

```r
error.rate <- c()
for (i in 2:4){
  set.seed(i)
  train <- sample(nrow(Default), nrow(Default)/2)
```

```
  glm.fit <- glm(default ~ income + balance, data = Default,family = binomial, subset = train)
  glm.pred <- rep("No", nrow(Default)/2)
  glm.probs <- predict(glm.fit, Default[-train, ], type = "response")
  glm.pred[glm.probs > 0.5] <- "Yes"
  error.rate <- c(error.rate,mean(glm.pred != Default[-train, ]$default))
}
error.rate
```

```
## [1] 0.0238 0.0264 0.0256
```

```
mean(error.rate)
```

```
## [1] 0.02526667
```

All the three test error rates are around 2.62%. The difference between the largest and the smallest test error rate is 0.0028%. $[max(error.rate) - min(error.rate)]$

(d) [4pt]

Now consider a logistic regression model that predicts the probability of default using income, balance, and a dummy variable for student. Estimate the test error for this model using the validation set approach. Comment on whether or not including a dummy variable for student leads to a reduction in the test error rate.

Solution

```
set.seed(1)
train <- sample(nrow(Default), nrow(Default)/2)
glm.fit <- glm(default ~ income + balance + student, data = Default, family = binomial,
    subset = train)
glm.pred <- rep("No", nrow(Default)/2)
glm.probs <- predict(glm.fit, Default[-train, ], type = "response")
glm.pred[glm.probs > 0.5] <- "Yes"
mean(glm.pred != Default[-train, ]$default)
```

```
## [1] 0.026
```

The test error rate is 2.88%.

Including a dummy variable for student does not improve the test error rate.

CHR 5.6 [15pt]

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 coefficients 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) [3pt]

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.

Solution

```r
library(ISLR)
summary(Default)
```

```
##   default    student       balance           income     
##   No :9667   No :7056   Min.   :   0.0   Min.   :  772  
##   Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340  
##                         Median : 823.6   Median :34553  
##                         Mean   : 835.4   Mean   :33517  
##                         3rd Qu.:1166.3   3rd Qu.:43808  
##                         Max.   :2654.3   Max.   :73554
```

```r
set.seed(1)
glm.fit <-  glm(default ~ income + balance, data = Default, family = binomial)
summary(glm.fit)
```

```
## 
## Call:
## glm(formula = default ~ income + balance, family = binomial, 
##     data = Default)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
## 
## Number of Fisher Scoring iterations: 8
```

(b) [4pt]

Write a function, boot.fn(), that takes as input the Default data set as well as an index of the observations, and that outputs the coefficient estimates for income and balance in the multiple logistic regression model.

Solution

```r
boot.fn <- function(data, index)
  return(coef(glm(default ~ income + balance,
    data = data, family = binomial, subset = index)))
```

(c) [4pt]

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

Solution

```
library(boot)
boot(Default, boot.fn, R=500)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Default, statistic = boot.fn, R = 500)
##
##
## Bootstrap Statistics :
##          original        bias     std. error
## t1* -1.154047e+01 -2.298134e-02 4.212334e-01
## t2*  2.080898e-05 -2.053140e-07 5.184890e-06
## t3*  5.647103e-03  1.678038e-05 2.178846e-04
```

The standard errors of the logistic regression coefficients for income and balance are 4.72×10^{-6} and 2.22×10^{-4}, when the number of bootstrap replicates equals to 500.

(d) [4pt]

Comment on the estimated standard errors obtained using the glm() function and using your bootstrap function.

Solution

The estimated standard errors obtained using the glm() function is 4.99×10^{-6} and 2.27×10^{-4}. They are very close to those in using the bootstrap function.

CHR 5.9 [24pt]

We will now consider the Boston housing data set, from the MASS library.

(a) [1pt]

Based on this data set, provide an estimate for the population mean of medv. Call this estimate $\hat{\mu}$.

Solution

```
library(MASS)
summary(Boston)
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
```

```
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
medv.mean <- mean(Boston$medv)
medv.mean
```

```
## [1] 22.53281
```

The estimate $\hat{\mu}$ is 22.53.

(b) [2pt]

Provide an estimate of the standard error of $\hat{\mu}$. Interpret this result. Hint: We can compute the standard error of the sample mean by dividing the sample standard deviation by the square root of the number of observations.

Solution

```r
medv.err <- sd(Boston$medv)/sqrt(nrow(Boston))
medv.err
```

```
## [1] 0.4088611
```

The estimate of the standard error of $\hat{\mu}$ is 0.4089.

(c) [5pt]

Now estimate the standard error of $\hat{\mu}$ using the bootstrap. How does this compare to your answer from (b)?

Solution

```r
set.seed(1)
boot.fn <- function(data, index)
```

```
  return(mean(data[index]))
library(boot)
bstrap <- boot(Boston$medv, boot.fn, R=1000)
bstrap
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston$medv, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 22.53281 0.007650791   0.4106622
```

The estimate of the standard error of $\hat{\mu}$ using the bootstrap is 0.4112.

It is similar to answer from (b).

(d) [2pt]

Based on your bootstrap estimate from (c), provide a 95 % confidence interval for the mean of medv. Compare it to the results obtained using t.test(Boston$medv). Hint: You can approximate a 95 % confidence interval using the formula $[\hat{\mu} - 2SE(\hat{\mu}), \hat{\mu} + 2SE(\hat{\mu})]$.

Solution

```
c(bstrap$t0 - 2*sd(bstrap$t),   # lower bound
bstrap$t0 + 2*sd(bstrap$t) ) # upper bound
```

```
## [1] 21.71148 23.35413
```

```
t.test(Boston$medv)$conf.int
```

```
## [1] 21.72953 23.33608
## attr(,"conf.level")
## [1] 0.95
```

Based on the bootstrap estimate from (c), a 95 % confidence interval for the mean of medv is [21.70893, 23.35668].

The 95 % confidence interval obtained using t.test(Boston$medv) is [21.72953, 23.33608].

Bootstrap estimate only 0.02 away for t.test estimate.

(e) [2pt]

Based on this data set, provide an estimate, $\hat{\mu}_{med}$, for the median value of medv in the population.

Solution

```
medv.med <- median(Boston$medv)
medv.med
```

```
## [1] 21.2
```

The estimate, $\hat{\mu}_{med}$, for the median value of medv in the population is 21.2

(f) [5pt]

We now would like to estimate the standard error of $\hat{\mu}_{med}$.Unfortunately, there is no simple formula for computing the standard error of the median. Instead, estimate the standard error of the median using the bootstrap. Comment on your findings.

Solution

```r
boot.fn <- function(data, index)
  return(median(data[index]))
boot(Boston$medv, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston$medv, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original   bias    std. error
## t1*     21.2 -0.0386   0.3770241
```

The estimate of the standard error of the median using the bootstrap is 0.3874. The standard error to the median value is smaller than the mean value.

(g) [2pt]

Based on this data set, provide an estimate for the tenth percentile of medv in Boston suburbs. Call this quantity $\hat{\mu}_{0.1}$. (You can use the quantile() function.)

Solution

```r
medv.tenth <- quantile(Boston$medv, 0.1)
medv.tenth
```

```
##   10%
## 12.75
```

An estimate for the tenth percentile of medv in Boston suburbs is 12.75.

(h) [5pt]

Use the bootstrap to estimate the standard error of $\hat{\mu}_{0.1}$. Comment on your findings.

Solution

```r
boot.fn <- function(data, index) return(quantile(data[index], 0.1))
boot(Boston$medv, boot.fn, 1000)
```

```
##
```

```
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston$medv, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*    12.75  0.0186   0.4925766
```

The standard error of $\hat{\mu}_{0.1}$ is estimated to be 0.5113 using the bootstrap.

The standard error of $\hat{\mu}_{0.1}$ is larger than the standard error to the median value and the mean value.