Cross Validation and Bootstrap

Lab 9 - MATH 4322

Problem 1

We will use the data Weekly in the `ISLR2` package to predict Direction using Lag1 and Lag2.

Description

Weekly percentage returns for the S&P 500 stock index between 1990 and 2010.

Lag1 Percentage return for previous week
Lag2 Percentage return for 2 weeks previous
Direction A factor with levels Down and Up indicating whether the market had a positive or negative return on a given week

```
     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            Today
 Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
 1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
 Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
 Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
 3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
 Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
 Direction
 Down:484
 Up  :605
```

Question 1: Is this a regression or classification problem?

This is a classification problem because the response variable is binary

Question 2: Which is the correct model, linear regression or logistic regression?

Between the two models the logistic regression is the correct model.

Question 3: Write out the equation of the correct model.

$$P(\hat{\text{Direction}} \overset{=\mu(x)}{=} \frac{exp(\beta_0 + \beta_1 \times \text{Lag1} + \beta_2 \times \text{Lag2})}{1 + exp(\beta_0 + \beta_1 \times \text{Lag1} + \beta_2 \times \text{Lag2})} + \epsilon$$

1. Fit a logistic regression model that predicts Direction using Lag1 and Lag2 on half of the data. This is the **trainning** data.

Question 4: In R what code do we use to separate the data into a train and test data.

```
library(ISLR2)
set.seed(100)
sample = sample(1:nrow(Weekly),nrow(Weekly)/2)
train = Weekly[sample,]
test = Weekly[-sample,]
```

Question 5: In R what code do would we use to get a model to predict direction based on lag1 and lag2?

```
direction.glm = glm(Direction ~ Lag1 + Lag2, data = train,
                    family = binomial)
direction.glm
```

```
Call:  glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = train)

Coefficients:
(Intercept)         Lag1         Lag2
    0.21145     -0.01450      0.05513

Degrees of Freedom: 543 Total (i.e. Null);  541 Residual
Null Deviance:      747.9
Residual Deviance: 745.6    AIC: 751.6
```

```r
summary(direction.glm)
```

```
Call:
glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = train)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept)  0.21145    0.08666   2.440   0.0147 *
Lag1        -0.01450    0.03647  -0.398   0.6910  
Lag2         0.05513    0.03914   1.408   0.1590  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 747.95  on 543  degrees of freedom
Residual deviance: 745.59  on 541  degrees of freedom
AIC: 751.59

Number of Fisher Scoring iterations: 4
```

Use this model to predict the direction of the first observation. You can do this by predicting that the first observation will go up if $P(Direction = \text{"Up"}|Lag1, Lag2) > 0.5$

Question 6: What is the code to get a prediction of the first observation?

```r
predict.glm(direction.glm,newdata = Weekly[1,],type = "response")
```

```
        1 
0.5710867 
```

= P (Direction = "Up" | X)

```r
Weekly[1,]$Direction
```

```
[1] Down
Levels: Down Up
```

= Actual value of Direction

Question 7: Would we predict the first observation to go up or down?

We would predict the first observation to go up

Question 8: Is the a correct prediction or miss classified?

The original observation is going Down, so this is miss classified.

We want to create a confusion matrix to determine the proportion of miss classified observations. This is called the *error rate*.

Question 9: What is the code to create this confusion matrix?

```
pred.train = predict.glm(direction.glm,type = "response")
pred.direction.train = ifelse(pred.train>0.5,"Up","Down")
(confmat.train = table(train$Direction,pred.direction.train))
```

row, column

```
      pred.direction.train
       Down  Up
  Down   14 229
  Up     10 291
```

Question 10: What is the error rate based on the training data?

```
(confmat.train[1,2]+confmat.train[2,1])/sum(confmat.train)
```

```
[1] 0.4393382
```

2. Use this trained model to make a prediction from the data that we did not use. This is the **test** data.

```
pred.test = predict.glm(direction.glm,
                        type = "response",
                        newdata = test)
pred.direction.test = ifelse(pred.test>0.5,"Up","Down")
(confmat.test = table(test$Direction,pred.direction.test))
```

```
      pred.direction.test
       Down  Up
  Down   10 231
  Up     13 291
```

Question 11: What is the test error rate?

```
(confmat.test[1,2]+confmat.test[2,1])/sum(confmat.test)
```

```
[1] 0.4477064
```

3. We want to create a LOOCV test error rate for the whole data using a for loop.

Write a for loop from $i = 1$ to $i = n$, where n 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 i^{th} observation to predict Direction using Lag1 and Lag2.
ii. Compute the posterior probability of the market moving up for the i^{th} observation.
iii. Use the posterior probability for the i^th observation in order to predict whether or not the market moves up.
iv. Determine whether or not an error was made in predicting the direction for the i^{th} observation. If an error was made, then indicate this as a 1, and otherwise indicate it as a 0.

```
loocv.err = NA
for (i in 1:nrow(Weekly)){
  sample = i
  fit.glm = glm(Direction ~ Lag1 + Lag2,
                data = Weekly[-sample,],
                family = binomial)
  pred.fit = predict.glm(fit.glm,Weekly[sample,],
                         type = "response")
  pred.direction = ifelse(pred.fit>0.5,"Up","Down")
  loocv.err[i] = (Weekly[sample,]$Direction != pred.direction)
}
```

Question 12: Take the average of the n numbers obtained in **iv** in order to obtain the LOOCV estimate for the test error. What is the value?

```
mean(loocv.err)
```

```
[1] 0.4499541
```

4. We will use the `cv.glm` function to determine LOOCV estimate for the test error. Since the response is binary, we will have to create a **cost** function to determine what probability we want to use as a cut off for "Up".

```r
library(boot)
```

```
Warning: package 'boot' was built under R version 4.3.2
```

```r
#Since the response is a binary variable an appropriate cost function is
cost <- function(r, pi = 0) mean(abs(r-pi) > 0.5)
direction.glm =  glm(Direction ~ Lag1 + Lag2,
                      data = Weekly,family = binomial)
cv.glm(Weekly,direction.glm,cost)$delta[1]
```

```
[1] 0.4499541
```

Question 13: Give the cross validation estimate from this method. Compare this to the value in Task 3, is it the same, higher or lower?

This is the same as previous value

5. We will do a 10-fold cross validation

```r
set.seed(10)
cv.glm(Weekly,direction.glm,cost, K = 10)$delta[1]
```

```
[1] 0.4481175
```

Question 14: Is this the same value as the loocv error?

This is not the same value

Question 15: Repeat the `cv.glm` code again. Do you get the same value?

```r
cv.glm(Weekly,direction.glm,cost, K = 10)$delta[1]
```

```
[1] 0.4508724
```

We do not get the same value as before

Question 16: What does the CV value represent in this scenario?

The CV is from the 10 samples, the average error rate

Problem 2

Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively, where X and Y are random quantities. We will invest a fraction, α, of our money in X, and will invest the remaining $1-\alpha$ in Y . Since there is variability associated with the returns on these two assets, we wish to choose α to minimize the total risk, or variance, of our investment. In other words, we want to minimize $Var(\alpha X+(1-\alpha)Y)$. One can show that the value that minimizes the risk is given by

$Var(\alpha X + (1-\alpha)Y)$
$= \alpha^2 Var(X) + (1-\alpha)^2 Var(Y) + 2\alpha(1-\alpha) cov(X,Y)$

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

Where, $\sigma_X^2 = Var(X)$, $\sigma_Y^2 = Var(Y)$, and $\sigma_{XY} = Cov(X,Y)$.

In reality the population variances and covariance is unknown so we have to use estimates, using a data set that contains past measurements for X and Y. We can then estimate the value of α that minimizes the variance of our investment using

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

1. Install and/or call the ISLR2 library, we will be using the Portfolio data set.

```
#install.packages("ISLR") #(Remove # if you have not installed this package)
library(ISLR2)
```

2. Create the function which takes as input (X,Y) data as well as a vector indicating which observations should be used to estimate α. The function then outputs the estimate for α based on the selected observations. The function is as follows:

```
alpha.fn = function(data,index) {
  X = data$X[index]
  Y = data$Y[index]
  return((var(Y) - cov(X,Y))/(var(X) + var(Y) - 2*cov(X,Y)))
}
```

Be careful about capitalization and lower case in these variables.

3. This function *returns* or outputs an estimate for α based on a applying the formula to the observations indexed by the argument index. For instance, the following command tells R to estimate α using all 100 observations.

```r
(alpha.hat = alpha.fn(Portfolio,1:100))
```

```
[1] 0.5758321
```

Question ~~1~~ 17: From this command, give an estimate of α.

$$\hat{\alpha} = 0.5758$$

4. The following command uses the sample function to randomly select 100 observations from the range 1 to 100, with replacement. This is equivalent to constructing a new bootstrap data set and recomputing α based on the new data set.

```r
set.seed(10)
(alpha.hat2 = alpha.fn(Portfolio,sample(100,100,replace = TRUE)))
```

```
[1] 0.508921
```

Question ~~2~~ 18: From this command, give an estimate of α.

$$\hat{\alpha}_2 = 0.5089$$

We can implement a bootstrap analysis by performing this command many times, recording all of the corresponding estimate for α, and computing the resulting standard deviation. However, the boot() function automates this approach. Below, is the function to produce $R = 1000$ bootstrap estimates for α.

```r
# install.packages("boot") #(Remove # if you have not installed this package)
library(boot)
alpha.boot = boot(Portfolio,alpha.fn,R = 1000)
alpha.boot
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Portfolio, statistic = alpha.fn, R = 1000)
```

```
Bootstrap Statistics :
     original      bias    std. error
t1* 0.5758321 0.00702054  0.09237541
```

Question 19: Original $\alpha = 0.5758$
$SE(\hat{\alpha}) = 0.0923$

```
(mean.boot = mean(alpha.boot$t))
```

```
[1] 0.5828526
```

$= 0.5758 + 0.007$

Question 20:
$SD(\hat{\alpha}) = SE(\hat{\alpha})$
we are off by about 9.23%

```
(sd.boot = sd(alpha.boot$t))
```

```
[1] 0.09237541
```