Homework 1 Exercise 2 and 3

Ma Jingchun, 2020111235

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 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

```
library(ISLR2)
```

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

```
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            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
##
##
##
##
```

```
set.seed(1)
attach(Weekly)

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

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Weekly)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.623  -1.261   1.001   1.083   1.506
##
## 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
```

2. Fit a logistic regression model that predicts Direction using Lag1 and Lag2 using all but the first observation

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

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Weekly[-1,
##     ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6258  -1.2617   0.9999   1.0819   1.5071
##
## 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
```

3. 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.glm(glm.fit, Weekly[1, ], type = "response") > 0.5
```

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

4. 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:

- Fit a logistic regression model using all but the ith observation to predict Direction using Lag1 and Lag2

```
for (i in 1:(dim(Weekly)[1])) {
  glm.fit = glm(Direction ~ Lag1 + Lag2, data = Weekly[-i, ], family = binomial)
}
```

- Compute the posterior probability of the market moving up for the ith observation

```
for (i in 1:(dim(Weekly)[1])) {
  glm.fit = glm(Direction ~ Lag1 + Lag2, data = Weekly[-i, ], family = binomial)
  is_up = predict.glm(glm.fit, Weekly[i, ], type = "response") > 0.5
}
```

- Use the posterior probability for the ith observation in order to predict whether or not the market moves up

```
for (i in 1:(dim(Weekly)[1])) {
  glm.fit = glm(Direction ~ Lag1 + Lag2, data = Weekly[-i, ], family = binomial)
  is_up = predict.glm(glm.fit, Weekly[i, ], type = "response") > 0.5
  is_true_up = Weekly[i, ]$Direction == "Up"
}
```

- Determine whether or not an error was made in predicting the direction for the ith observation. If an error was made, then indicate this as a 1, and otherwise indicate it as a 0

```
count = rep(0, dim(Weekly)[1])
for (i in 1:(dim(Weekly)[1])) {
  glm.fit = glm(Direction ~ Lag1 + Lag2, data = Weekly[-i, ], family = binomial)
  is_up = predict.glm(glm.fit, Weekly[i, ], type = "response") > 0.5
  is_true_up = Weekly[i, ]$Direction == "Up"
  if (is_up != is_true_up)
    count[i] = 1
}
sum(count)
```

```
## [1] 490
```

8. 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(count)
```

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