Homework 7

Snigdha Peddi in Collaboration with John Herbert

Question 5.4.3 (on page 198)

We now review k-fold cross-validation.

(a) Explain how k-fold cross-validation is implemented.

Answer:

Unlike the LOOCV where n-1 observations are considered for the training and 1 observation used for testing each time, in K-fold cross validation the whole data set is partitioned into multiple data sets (k=5 or k=10 etc). If K=10,where data set is partitioned into 10 parts,one set of data is left out as validation set and remaining 9 data sets are used as training data, a MSE is calculated. Next time, a new set of data is left out as validation set and remaining 9 sets of data is used for training and a MSE is calculated. Similar process is followed until all 10 data sets act as validation data sets resulting in 10 different MSE. An average of this MSE is calculated and that gives the k-fold cross-validation accuracy.

$$Cross\ Validation\ Accuracy,\ CV(k) = K^{-}1\sum_{k1}^{K} MSE_k$$

(b) What are the advantages and disadvantages of k-fold cross-validation relative to:

Answer:

i. The validation set approach?

Advantages of k-fold CV:

- Less Bias,Less Variance.
- Performs well with few number of observations.
- Complete data will be used for training and testing the models.

Disadvantages of k-fold CV:

- validation set approach is easy to understand.
- computationally expensive compared to validation set approach where the model is trained and tested only once.

ii. LOOCV?

Advantages of k-fold CV:

- Computationally inexpensive.
- We can estimate the accuracy.
- More stable and less variable.

Disadvantages of k-fold CV:

- More bias when K<n, compared to LOOCV.

References

- STAT 602, Resampling Methods,Lecture,*Chapter 5 Part 1* by Dr.Saunders.
- STAT 602, Resampling Methods,Lecture,*Chapter 5 Part 2* by Dr.Saunders.

Question 5.4.5 (on page 198)

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.

```
## Dimensions of Default Datatset:
##  10000 4
```

```
##
##
## Summary of Default Dataset:
```

```
##  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
```

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

Answer:

A Logistic regression model is fit using income and balance as predictors and default variables as response.

```
Logistic Regression Model:
default.glm<-glm(default~income+balance,data=default,family="binomial")
```

```
##
## 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) Using the validation set approach, estimate the test error of this model. In order to do this, you must perform the following steps:

Answer:

i Split the sample set into a training set and a validation set.

A 70:30 split is done on the default dataset.

```
## Dimensions of Traing set(70:30 split):
##  7000 4
```

```
##
## Dimensions of Validation set(70:30 split):
##  3000 4
```

ii. Fit a multiple logistic regression model using only the training observations.

Using glm() function a multiple logistic regression model is fit on the training data.

```
Logistic Regression Model:
default.glm.train70<-glm(default~income+balance,
                    data=train70,family="binomial")
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = "binomial",
##     data = train70)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5091  -0.1393  -0.0537  -0.0193   3.7609
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.176e+01  5.325e-01 -22.083  < 2e-16 ***
## income       2.019e-05  6.014e-06   3.358 0.000785 ***
## balance      5.791e-03  2.792e-04  20.738  < 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: 2043.8  on 6999  degrees of freedom
## Residual deviance: 1081.8  on 6997  degrees of freedom
## AIC: 1087.8
##
## Number of Fisher Scoring iterations: 8
```

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.

Answer:

Predictions were made for each individual in the test set and classified the predicted observation as default if the prediction is more than 0.5.

iv. Compute the validation set error, which is the fraction of the observations in the validation set that are misclassified.

Answer:

Misclassification rate is calculated for the prediction made using validation set. with a 70:30 split, the misclassification rate observed is 2.77%.97.2% (accuracy) of the observations were correctly classified.

Table 1: *70/30 split*

Misclassification
0.02767

(c) 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.

Answer:

```
The default data is further analysed using 80:20, 65:35, 60:40 training and test splits.
```

```
## Dimensions of Traing set(80:20 split):
##  8000 4
```

```
##
```

```
## Dimensions of Validation set(80:20 split):
##  2000 4

##
## Logistic Regression Model (80:20 split):

## glm(formula = default ~ income + balance, family = "binomial",
##     data = train80)
```

Table 2: *80/20 split*

Misclassification
0.027

```
## Dimensions of Traing set(65:35 split):
##  6500 4

##
## Dimensions of Validation set(65:35 split):
##  3500 4

##
## Logistic Regression Model (65:35 split):

## glm(formula = default ~ income + balance, family = "binomial",
##     data = train65)
```

Table 3: *65/35 split*

Misclassification
0.02686

```
## Dimensions of Traing set(60:40 split):
##  6000 4

##
## Dimensions of Validation set(60:40 split):
##  4000 4

##
## Logistic Regression Model (60:40 split):

## glm(formula = default ~ income + balance, family = "binomial",
##     data = train60)
```

Table 4: *60/40 split*

Misclassification
0.027

These splits resulted in misclassifiaction rates of 2.7%,2.69% and 2.7% respectively. The comparative error rate is presented in the table below.

Table 5: Comparitive Error rate for Default Data

60:40 Split	*65:35 Split*	*70:30 Split*	*80:20 Split*
0.027	0.0269	0.0277	0.027

(d) 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.

Answer:

Logistic regression models were fit using all predictor variables.

```
## Dimensions of Traing set(65:35 split):
##  6500 4
```

```
##
## Dimensions of Validation set(65:35 split):
##  3500 4
```

```
##
## Logistic Regression Model (65:35 split):
```

```
## glm(formula = default ~ ., family = "binomial", data = train.new)
```

```
##
## Coefficients of the model with 65:35 split:
```

```
##                  Estimate   Std. Error     z value     Pr(>|z|)
## (Intercept) -1.113877e+01 6.214615e-01 -17.9235157 7.728576e-72
## studentYes  -4.914147e-01 2.916841e-01  -1.6847498 9.203685e-02
## balance      5.794466e-03 2.905551e-04  19.9427465 1.732844e-88
## income       6.751676e-06 1.025013e-05   0.6586918 5.100937e-01
```

Table 6: *All Predictors:65:35 Split*

Misclassification
0.02743

```
## Dimensions of Traing set(70:30 split):
##  7000 4

##
## Dimensions of Validation set(70:30 split):
##  3000 4

##
## Logistic Regression Model (70:30 split):

## glm(formula = default ~ ., family = "binomial", data = train.new1)

##
## Coefficients of the model with 70:30 split:

##                  Estimate   Std. Error     z value     Pr(>|z|)
## (Intercept) -1.117542e+01 6.027456e-01 -18.5408610 9.666173e-77
## studentYes  -5.549462e-01 2.831578e-01  -1.9598482 5.001354e-02
## balance      5.867588e-03 2.841631e-04  20.6486588 1.003677e-94
## income       4.774176e-06 9.923888e-06   0.4810792 6.304602e-01
```

Table 7: *All Predictors:70:30 split*

Misclassification
0.028

```
## Dimensions of Traing set(60:40 split):
##  6000 4

##
## Dimensions of Validation set(60:40 split):
##  4000 4

##
## Logistic Regression Model (60:40 split):

## glm(formula = default ~ ., family = "binomial", data = train.new2)

##
## Coefficients of the model with 60:40 split:

##                  Estimate   Std. Error     z value     Pr(>|z|)
## (Intercept) -1.120609e+01 6.445558e-01 -17.3857554 1.057842e-67
```

```
## studentYes  -2.582602e-01 3.058363e-01  -0.8444395 3.984239e-01
## balance      5.644638e-03 2.933083e-04  19.2447251 1.562904e-82
## income       1.396703e-05 1.068892e-05   1.3066827 1.913205e-01
```

Table 8: *All Predictors:60:40 split*

Misclassification
0.02725

```
## Dimensions of Traing set(80:20 split):
##  8000 4

##
## Dimensions of Validation set(80:20 split):
##  2000 4

##
## Logistic Regression Model (80:20 split):

## glm(formula = default ~ ., family = "binomial", data = train.new3)

##
## Coefficients of the model with 80:20 split:

##                  Estimate   Std. Error     z value      Pr(>|z|)
## (Intercept) -1.104859e+01 5.572855e-01 -19.8257183  1.786127e-87
## studentYes  -6.330709e-01 2.634320e-01  -2.4031666  1.625378e-02
## balance      5.805091e-03 2.631615e-04  22.0590407 7.822416e-108
## income       4.883858e-06 9.158856e-06   0.5332389  5.938682e-01
```

Table 9: *All Predictors:80:20 split*

Misclassification
0.04325

The p value of income variable at 0.05% confidence interval is larger on addition of dummy variable "Student:Yes" to all the models (in all the different splits). Even the P value of the dummy variables is higher that 0.05 except for 70:30 split where it is marginal with 0.05. Missclassification rate of all splits is presented in table below.the 80:20 split has highest error rate of 4.32% and 60:40 split has lowest,2.72%.

Table 10: Comparitive Error rate for: All Predictors

60:40 Split	*65:35 Split*	*70:30 Split*	*80:20 Split*
0.0272	0.0274	0.028	0.0432

On comparison, the addition of the dummy variable student:yes to the models has an effect when the default data is split 80:20. In all other cases the error rates are pretty close (2.7%) and seems to have not much

effect.

Question 5.4.7 (page 200)

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 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).

```
## Dimensions of Weekly Data Set:

## [1] 1089    9
```

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

Answer:

A logistic regression model is fit for weekly data using Lag1 and Lag2 variables as predictors.

```
##
## Logistic Regression model of weekly dataset:

## glm(formula = Direction ~ Lag1 + Lag2, family = "binomial", data = weekly)
```

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

Answer:

A logistic regression model is fit using Lag1 and Lag2 variables as predictors for observations of weekly data except the first variable.

```
##
## Logistic Regression model of weekly dataset-first observation:

## glm(formula = Direction ~ Lag1 + Lag2, family = "binomial", data = (weekly[-1,
##     ]))
```

(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?

Answer:

The predictions were made on the first observation using the logistic regression model fit above.The prediction made for first observation is missclassified as direction "Up" instead of "Down".

Table 11: Direction of 1st Observation

Predicted	Actual
1	0

(d) 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 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 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 ith observation. If an error was made, then indicate this as a 1, and otherwise indicate it as a 0. *(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.

Answer:

LOOCV is estimated by using *glm()* and *predict.glm()* functions using a *for loop* where each time one observation (nth) is used as test set and n-1 observations are used as training set. This process is continued until all the observations in the dataset are used as validation sets. The estimated error rate using LOOCV is 0.45.

```
##
##  LOOCV estimate for error rate:
##  0.4499541
```

References

- Chapter 5,Resampling Methods,*An Introduction to Statistical Learning with Applications in R* by Gareth James.

Question 4:

Write your own code (similar to Exercise #3 above) to estimate test error using k-fold cross validation for fitting a linear regression model of the form

$$mpg = \beta_0 + \beta_1 * X_1 + \beta_2 * X_1^2$$

from the **Auto** data in the **ISLR** library, with X_1 = horsepower. Use `echo = T` to show the code. Test this code with `k = 5` and `k = 30`. Discuss the computational trade-off between the two choices of `k`. Do not use the `cv.glm` function.

Answer:

K-Fold cross validation error rate is estimated by using *glm()* and *predict.glm()* functions using a *for loop* where each time one fold of observations (for k=5, 1 set is used as test set) are used as test set and k-1 folds of observations are used as training set. This process is continued until all the folds are used as validation sets. The estimated error rate using k=5 is 19.1 and k=30 is 19.2.

```r
#Reading Auto data
auto1<-Auto
#Setting seed for reproducibility
set.seed(16489)
#Randomly shuffling the Auto data
auto1<-auto1[sample(nrow(auto1)),]

#Create 5 equally size folds
folds<-cut(seq(1,nrow(auto1)),breaks=5,labels=FALSE)
#No. of subsets,k=5
k=5
#Creating Empty Vector of length k
V<-rep(0,length(k))
#For loop to get K-fold cross validation error rate
for(i in 1:5){
  auto.mod<-lm(mpg~poly(horsepower,2),data=auto1[folds!=i,])
  auto.pred<-predict.lm(auto.mod,auto1[folds==i,])
V[i]<-mean((auto.pred-auto1$mpg[folds==i])^2)
test.error<-mean(V)
}
cat("\nK=5,estimate for error rate:\n\n",test.error)
```

```
##
## K=5,estimate for error rate:
##
##  19.10675
```

```r
####

#Create 30 equally size folds
folds<-cut(seq(1,nrow(auto1)),breaks=30,labels=FALSE)
#No. of subsets,k=30
k=30
#Creating Empty Vector of length k
V<-rep(0,length(k))
#For loop to get K-fold cross validation error rate
for(i in 1:30){
  auto.mod<-lm(mpg~poly(horsepower,2),data=auto1[folds!=i,])
  auto.pred<-predict.lm(auto.mod,auto1[folds==i,])
V[i]<-mean((auto.pred-auto1$mpg[folds==i])^2)
test.error<-mean(V)
}
cat("\nK=30,estimate for error rate:\n\n",test.error)
```

```
##
## K=30,estimate for error rate:
##
##  19.1829
```

K-fold cross validation error rate for both k=5 and k=30 is very close.However, k=30 will take more computational power than k=3 as the loop has to run 25 times more.

REFERENCES

- StackExchange blog post on *How to split data set to do 10-fold cross validation.*