Chapter 5 HW

Nehemya McCarter-Ribakoff

4 April 2017

Conceptual Questions

Exercise 3: We now review k-fold cross-validation.

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

k-fold cross-validation follows the following steps: 1. divide the set into k parts 2. remove the first part 3. fit the model on the remaining k - 1 parts 4. compute the MSE on the first part 5. Repeat k times, removing a different part each time 6. average the k different MSEs

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

i. The validation set approach?

The validation set approach is conceptually simple and easier to implement than k-fold CV. However, because its estimate is based on a random division of the sample, its estimate can be highly variable. k-fold CV provides a non-random result. The validation set approach also is based upon only a subset of observations, while k-fold CV gets the full scope of the data provided. This causes the validation set approach to tend to overestimate the test error rate moreso than k-fold CV would in general.

ii. LOOCV?

The Leave-One-Out Cross-Validation approach is really just a special case of k-fold CV where k = n. This approach is less computationally intensive than general k-fold CV, and the two approaches tend to yield similar results. Compared to k-fold CV, LOOCV has less bias and more variance when k < n, but k-fold CV has a computational advantage under this condition. more variance when k > n. Neither suffers excessively from high bias or variance.

Applied Questions

Exercise 5: 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) Fit a logistic regression model that uses income and balance to predict default.

```
library(ISLR)
glm.fit = glm(Default$default ~ Default$income + Default$balance, family=binomial(link
="logit"), data=Default)
glm.fit
```

```
##
## Call:  glm(formula = Default$default ~ Default$income + Default$balance,
##     family = binomial(link = "logit"), data = Default)
##
## Coefficients:
##     (Intercept)   Default$income  Default$balance
##      -1.154e+01        2.081e-05        5.647e-03
##
## Degrees of Freedom: 9999 Total (i.e. Null);  9997 Residual
## Null Deviance:       2921
## Residual Deviance: 1579  AIC: 1585
```

(b) 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.

```
set.seed(1)
# Default has 10,000 observations
# Let's use 7500 for training
set.train=sample(nrow(Default), nrow(Default)*0.5)
```

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

```
glm.trainFit = glm(Default$default ~ Default$income + Default$balance,
family=binomial(link=logit), subset=set.train)
glm.trainFit
```

```
##
## Call:  glm(formula = Default$default ~ Default$income + Default$balance,
##     family = binomial(link = logit), subset = set.train)
##
## Coefficients:
##     (Intercept)   Default$income  Default$balance
##      -1.208e+01        1.858e-05        6.053e-03
##
## Degrees of Freedom: 4999 Total (i.e. Null);  4997 Residual
## Null Deviance:       1457
## Residual Deviance: 734.4     AIC: 740.4
```

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.

```
posteriorProb = predict(glm.trainFit, data=Default[-set.trian,], type="response")
```

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

```
pred2 = ifelse(posteriorProb > 0.5, "Yes", "No")
table(pred2, data=Default[-set.train,]$default)
```

```
##      data
## pred2   No  Yes
##   No  4756  166
##   Yes   77    1
```

(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.

(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.

Exercise 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 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) 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.

```
glm.fit = glm(Default$default ~ Default$income + Default$balance,
              family=binomial(link='logit'))
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Default$default ~ Default$income + Default$balance,
##     family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## Default$income   2.081e-05  4.985e-06   4.174 2.99e-05 ***
## Default$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
```

income standard error: 4.985e-06

balance standard error: 2.274e-07

(b) 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.

```
boot.fn <- function(df, trainid) {
  return(coef(glm(default ~ income + balance, data=df, family=binomial, subset=traini
d)))
}
```

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

```
boot::boot(Default, boot.fn, R=100)
```

```
##
## CASE RESAMPLING BOOTSTRAP FOR CENSORED DATA
##
##
## Call:
## boot::boot(data = Default, statistic = boot.fn, R = 100)
##
##
## Bootstrap Statistics :
##          original        bias     std. error
## t1* -1.154047e+01  8.977336e-02 4.135699e-01
## t2*  2.080898e-05  6.334127e-08 4.216745e-06
## t3*  5.647103e-03 -5.306251e-05 2.117636e-04
```

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

glm standard errors

income: 4.985e-06 balance: 2.274e-04

bootstrap standard errors

income: 5.607415e-06 balance: 2.413098e-04

The bootstrap errors are higher than the glm standard errors, particularly for income. I suspect this may be a flaw in my boot.fn() function.

Exercise 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).

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

```
glm.fit = glm(Weekly$Direction ~ Weekly$Lag1 + Weekly$Lag2, family=binomial(link='logi
t'))
```

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

```
glm.fitFirstObsRemoved = glm(Weekly$Direction[2:1089] ~ Weekly$Lag1[2:1089] + Weekly$L
ag2[2:1089], family=binomial(link='logit'))
```

(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?

(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.

Teamwork report

Team member	Conceptual	Applied	Contribution %
Nehemya	Yes	Yes	100%
Total			100%