Prepared by Richard Rennehan

NSCC Institute of Technology

DBAS3075 Introduction to Statistical Learning

Cross-validation assignment

April 4, 2018

# Questions and Answers

1. (1 pt) What are the three methods of holding out a subset of observations from the model fitting process in order to estimate a test error rate?
   1. **The three methods are the Validation Set Approach, Leave One Out Cross Validation (LOOCV), and K-Folds Cross-Validation.**

2. (1 pt) Match the three methods listed in question 1 with the appropriate definition from the following definitions:

(a) K-Folds cross-validation: randomly divides the set of observations into a specified number of groups of approximately equal sizes (greater than 1), using each group as a validation set for testing the model accuracy, while the remaining groups are used to fit the model.

(b) Validation set approach: Randomly divides the set of observations into two parts; one part for fitting models, and another part used as a validations set for testing model accuracy.

(c) Leave one out cross validation: Given a total of N observations, hold out one observation that will be used for testing model accuracy, while the N-1 remaining observations are used for model fitting. Repeat this process N times with each observation being left out once (and only once), then average the residuals from each observation.

1. (1 pt) Which of the 3 methods of holding out a subset of observations should take the longest time to compute?
   1. **The leave one out cross validation is the most thorough, taking the longest time to compute.**
2. (1 pt) Create an R session, and before starting to write any scripts, be sure to include the boot, gam, leaps, and ISLR packages. Load the Hitters data set and then look at the data. Which of the variables are categorical variables (list these)? Remove these variables (columns) from your data.

*hitters = Hitters*

*names(hitters)*

*summary(hitters)*

**Columns 14 (league), 15 (division), and 20 (newLeague) are categorical.**

*hitters = hitters[,-c(14,15,20)]*

1. (1 pt) When inspecting the data, you should notice that some of the fields don't have apropriate values stored in them. What function do we use to remove these values? Be sure to remove these values before moving on to question 6. How many observations are remaining once this has been done?

**We use the na.omit() function to remove empty NA (not available) values.**

*Hitters = na.omit(hitters)*

**After we run this line, we go from 322 observations to 263 observations.**

1. (1 pt) Perform Best Subset Regression on the remaining data, using Salary as the response variable. Look at the plot of adjusted R-squared values and you should notice a significant improvement in the value of adjusted R-squared when the number of variables is increased from 1 to 2. Which two variables are included in the best two variable subset discovered by Best Subset Regression?

*best\_subset=regsubsets(Salary~.,data=hitters, nvmax=16, method="exhaustive")*

*best\_summary = summary(best\_subset)*

*best\_summary*

*plot(best\_summary$adjr2, xlab="Number of variables", ylab="Adjusted R-squared", main="Adj-R^2 values of models of size 1-16 found by best subset selection")*

**The two variable model includes CRBI and Hits.**

7. (1 pt) When we want to use the Validation Set approach for building and testing models, we typically divide our data into two roughly equal groups. This is easy when we have an even number of observations, but when we have an odd number of observations, we want to put the extra observation in the training set. We can randomly divide our data by using the sample() function. For the remaining data from the Hitters data set, what values should we put in the arguments in the following line of code:

train=sample(263, 132) #Fill in the blanks or report the values, but order matters for points.

**The first value is the number of observations we have. The second value is typically half of that. Because half of 263 is 131.5, we round up to 132 to create the extra observation.**

1. (1 pt) Before creating the train vector (using the code in question 7), what function can you use in order to create randomized results that are able to be reproduced?

**We use set.seed() to create randomized results that stay consistent which each run-through**

9. (1 pt) Suppose I created the vector called "train" that was discussed in question 7, and suppose my data set is called "data", and I built a model called "m1". What is the name of the statistic calculated in the following line of code:

statistic=mean((Salary-predict(m1,data))[train]^2)

**This is an example of validation set approach. The statistic would be the mean squared average (MSE).**

10. (1 pt) Using beta terms (or just the letter b will do, such as b0, b1,...) for coefficients, write out the generalized model that will be produced by the following code:

lm(Salary~poly(Hits,3), data=Hitters, subset=train)

The general model syntax is:

**y = b0 + b1X + b2X­2 + b3X3 …**

Fill in each X with Hits variable:

**y = b + b1(Hits) + b2(Hits)2 + b3(Hits)3**

11. (1 pt) The following two lines of code each produce exactly the same thing:

lm1=lm(Salary~CRBI, data=Hitters)

glm1=glm(Salary~CRBI, data=Hitters)

If I wanted to perform a method of cross-validation (not validation set approach), why would I want to choose to use the second line of code instead of the first line?

**Compared to lm, the glm function provides additional functionality that makes cross- validation possible down the line. An example is the cv.glm() function, which lets us create new models to determine error values.**

# References

## R Code:

**library**("ISLR", lib.loc="~/R/win-library/3.4")  
**library**("leaps", lib.loc="~/R/win-library/3.4")  
**library**("boot", lib.loc="~/R/win-library/3.4")

## Warning: package 'boot' was built under R version 3.4.4

**library**("gam", lib.loc="~/R/win-library/3.4")

## Warning: package 'gam' was built under R version 3.4.4

## Loading required package: splines

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.4

## Loaded gam 1.15

hitters = Hitters  
**names**(hitters)

## [1] "AtBat" "Hits" "HmRun" "Runs" "RBI"   
## [6] "Walks" "Years" "CAtBat" "CHits" "CHmRun"   
## [11] "CRuns" "CRBI" "CWalks" "League" "Division"   
## [16] "PutOuts" "Assists" "Errors" "Salary" "NewLeague"

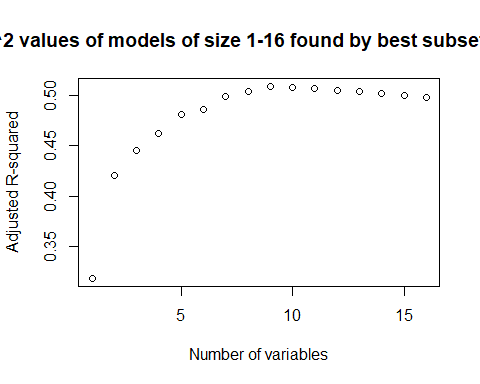
**summary**(hitters)

## AtBat Hits HmRun Runs   
## Min. : 16.0 Min. : 1 Min. : 0.00 Min. : 0.00   
## 1st Qu.:255.2 1st Qu.: 64 1st Qu.: 4.00 1st Qu.: 30.25   
## Median :379.5 Median : 96 Median : 8.00 Median : 48.00   
## Mean :380.9 Mean :101 Mean :10.77 Mean : 50.91   
## 3rd Qu.:512.0 3rd Qu.:137 3rd Qu.:16.00 3rd Qu.: 69.00   
## Max. :687.0 Max. :238 Max. :40.00 Max. :130.00   
##   
## RBI Walks Years CAtBat   
## Min. : 0.00 Min. : 0.00 Min. : 1.000 Min. : 19.0   
## 1st Qu.: 28.00 1st Qu.: 22.00 1st Qu.: 4.000 1st Qu.: 816.8   
## Median : 44.00 Median : 35.00 Median : 6.000 Median : 1928.0   
## Mean : 48.03 Mean : 38.74 Mean : 7.444 Mean : 2648.7   
## 3rd Qu.: 64.75 3rd Qu.: 53.00 3rd Qu.:11.000 3rd Qu.: 3924.2   
## Max. :121.00 Max. :105.00 Max. :24.000 Max. :14053.0   
##   
## CHits CHmRun CRuns CRBI   
## Min. : 4.0 Min. : 0.00 Min. : 1.0 Min. : 0.00   
## 1st Qu.: 209.0 1st Qu.: 14.00 1st Qu.: 100.2 1st Qu.: 88.75   
## Median : 508.0 Median : 37.50 Median : 247.0 Median : 220.50   
## Mean : 717.6 Mean : 69.49 Mean : 358.8 Mean : 330.12   
## 3rd Qu.:1059.2 3rd Qu.: 90.00 3rd Qu.: 526.2 3rd Qu.: 426.25   
## Max. :4256.0 Max. :548.00 Max. :2165.0 Max. :1659.00   
##   
## CWalks League Division PutOuts Assists   
## Min. : 0.00 A:175 E:157 Min. : 0.0 Min. : 0.0   
## 1st Qu.: 67.25 N:147 W:165 1st Qu.: 109.2 1st Qu.: 7.0   
## Median : 170.50 Median : 212.0 Median : 39.5   
## Mean : 260.24 Mean : 288.9 Mean :106.9   
## 3rd Qu.: 339.25 3rd Qu.: 325.0 3rd Qu.:166.0   
## Max. :1566.00 Max. :1378.0 Max. :492.0   
##   
## Errors Salary NewLeague  
## Min. : 0.00 Min. : 67.5 A:176   
## 1st Qu.: 3.00 1st Qu.: 190.0 N:146   
## Median : 6.00 Median : 425.0   
## Mean : 8.04 Mean : 535.9   
## 3rd Qu.:11.00 3rd Qu.: 750.0   
## Max. :32.00 Max. :2460.0   
## NA's :59

hitters = hitters[,**-c**(14,15,20)] *#Remove categorical variables)*  
  
hitters = **na.omit**(hitters) *#Remove NAs*  
  
best\_subset=**regsubsets**(Salary**~**.,data=hitters, nvmax=16, method="exhaustive")  
best\_summary = **summary**(best\_subset)  
best\_summary

## Subset selection object  
## Call: regsubsets.formula(Salary ~ ., data = hitters, nvmax = 16, method = "exhaustive")  
## 16 Variables (and intercept)  
## Forced in Forced out  
## AtBat FALSE FALSE  
## Hits FALSE FALSE  
## HmRun FALSE FALSE  
## Runs FALSE FALSE  
## RBI FALSE FALSE  
## Walks FALSE FALSE  
## Years FALSE FALSE  
## CAtBat FALSE FALSE  
## CHits FALSE FALSE  
## CHmRun FALSE FALSE  
## CRuns FALSE FALSE  
## CRBI FALSE FALSE  
## CWalks FALSE FALSE  
## PutOuts FALSE FALSE  
## Assists FALSE FALSE  
## Errors FALSE FALSE  
## 1 subsets of each size up to 16  
## Selection Algorithm: exhaustive  
## AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns  
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " "   
## 3 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " "   
## 4 ( 1 ) "\*" "\*" " " " " " " " " " " " " " " " " " "   
## 5 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " " "   
## 6 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " "\*"   
## 7 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*"   
## 9 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*"   
## 10 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*"   
## 11 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*"   
## 12 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" " " " " "\*"   
## 13 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" "\*" " " "\*"   
## 14 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" "\*" "\*" " " "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## CRBI CWalks PutOuts Assists Errors  
## 1 ( 1 ) "\*" " " " " " " " "   
## 2 ( 1 ) "\*" " " " " " " " "   
## 3 ( 1 ) "\*" " " "\*" " " " "   
## 4 ( 1 ) "\*" " " "\*" " " " "   
## 5 ( 1 ) "\*" " " "\*" " " " "   
## 6 ( 1 ) " " "\*" "\*" " " " "   
## 7 ( 1 ) " " "\*" "\*" " " " "   
## 8 ( 1 ) "\*" "\*" "\*" " " " "   
## 9 ( 1 ) "\*" "\*" "\*" "\*" " "   
## 10 ( 1 ) "\*" "\*" "\*" "\*" " "   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 12 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 14 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*"

**plot**(best\_summary**$**adjr2, xlab="Number of variables", ylab="Adjusted R-squared", main="Adj-R^2 values of models of size 1-16 found by best subset selection")



**set.seed**(1)  
train = **sample**(263,132)  
  
  
**lm**(Salary**~poly**(Hits,3), data=Hitters, subset=train)

##   
## Call:  
## lm(formula = Salary ~ poly(Hits, 3), data = Hitters, subset = train)  
##   
## Coefficients:  
## (Intercept) poly(Hits, 3)1 poly(Hits, 3)2 poly(Hits, 3)3   
## 509.1 4456.7 393.3 -729.1