Model Selection Framework

Matt Clark

March 22, 2018

The following code is for running a negbinomial glm on the park visitation data with a various combinations of variables and using leave one out cross validation (LOOCV) for model selection AND to assess model fit

This is for JOTR only and looks at total yearly visitation. This is intended to be used as a framework for later analysis of DEVA and JOTR visitation stratified by season and with social media data added

# Let’s do it

### Step 1

Import the simplified dataset and clean it up

library(readr)  
scaff <- read\_csv("~/ParkBreak/Scaffold data - Sheet1.csv")

## Parsed with column specification:  
## cols(  
## Year = col\_integer(),  
## Month = col\_integer(),  
## JOTR\_all = col\_double(),  
## JOTR\_overnight = col\_double(),  
## CCI\_USA = col\_double(),  
## LA\_pop = col\_integer(),  
## GoogleSearch\_JOTR = col\_integer(),  
## GoogleSearch\_DEVA = col\_integer(),  
## GoogleSearch\_Coachella = col\_integer(),  
## GasPrices = col\_double(),  
## NPS\_noJOTR = col\_number(),  
## BurningMan = col\_integer()  
## )

View(scaff)

There are some NA’s in there, lets remove those

scaff<-na.omit(scaff)

### Step 2

For this framework, lets look at 5 variables CCI, LA POP, Cochella Google searches, gas prices, and burning man attendance.

We need to standardize these variables so they are all on the same scale and the slopes are comparable

Create a function for standardizing variables and use it (or lose it)

stdize<-function(x) {return((x-mean(x)/(2\*sd(x))))}  
  
scaff$stdcci<-stdize(scaff$CCI\_USA)  
scaff$stdlapop<-stdize(scaff$LA\_pop)  
scaff$stdcochella<-stdize(scaff$GoogleSearch\_Coachella)  
scaff$stdgas<-stdize(scaff$GasPrices)  
scaff$stdburningman<-stdize(scaff$BurningMan)

Now all our parameter estimetes will be comparable to each other ###Step 3 Lets make our model using the different paramater combinations

library(MASS)  
## M1 we will use all the variables  
M1<-glm.nb(JOTR\_all~stdcci+stdlapop+stdcochella+stdgas+stdburningman, data=scaff)  
## M2 let's remove the google search variables  
M2<-glm.nb(JOTR\_all~stdcci+stdlapop+stdgas, data=scaff)  
## M3 lets use ONLY the google search terms  
M3<-glm.nb(JOTR\_all~stdcochella+stdburningman, data=scaff)  
## M4 lets use ONLY the economic variables  
M4<-glm.nb(JOTR\_all~stdcci+stdgas, data=scaff)

### Step 4

Let’s use LOOCV to evaluate each of the models

First create an empty vector to fill with using a loop

leftout<-rep(NA,times=length(scaff$JOTR\_all))

Now let’s fill it by removing one row at a time and running our model on the other datapoints, trying to predct the row we removed

for(i in 1:length(scaff$JOTR\_all)){  
 sub\_dat<-scaff[-i,]  
 m\_sub<-glm.nb(JOTR\_all~stdcci+stdlapop+stdcochella+stdgas+stdburningman, data=sub\_dat)  
 leftout[i]=predict(m\_sub,newdat=scaff[i,])   
}

### Step 5

Let’s recore our test statistics The test statistics for LOOCV are RMSE and R^2

library('Metrics')

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

## Make an R^2 function  
R2 <- function (x, y) cor(x, y) ^ 2  
R2M1<-R2(leftout,scaff$JOTR\_all)  
rmseM1<-rmse(leftout,scaff$JOTR\_all)

### Step 6

Now that we have successfully done this, let’s do it for the rest of the variables and compare them

## M2   
leftout2<-rep(NA,times=length(scaff$JOTR\_all))  
  
for(i in 1:length(scaff$JOTR\_all)){  
 sub\_dat2<-scaff[-i,]  
 m\_sub2<-glm.nb(JOTR\_all~stdcci+stdlapop+stdgas, data=sub\_dat2)  
 leftout2[i]=predict(m\_sub2,newdat=scaff[i,])   
}  
  
R2M2<-R2(leftout2,scaff$JOTR\_all)  
rmseM2<-rmse(leftout2,scaff$JOTR\_all)  
  
## M3  
leftout3<-rep(NA,times=length(scaff$JOTR\_all))  
  
for(i in 1:length(scaff$JOTR\_all)){  
 sub\_dat3<-scaff[-i,]  
 m\_sub3<-glm.nb(JOTR\_all~stdcochella+stdburningman, data=sub\_dat3)  
 leftout3[i]=predict(m\_sub3,newdat=scaff[i,])   
}  
  
R2M3<-R2(leftout3,scaff$JOTR\_all)  
rmseM3<-rmse(leftout3,scaff$JOTR\_all)  
  
## M4  
leftout4<-rep(NA,times=length(scaff$JOTR\_all))  
  
for(i in 1:length(scaff$JOTR\_all)){  
 sub\_dat4<-scaff[-i,]  
 m\_sub4<-glm.nb(JOTR\_all~stdcci+stdgas, data=sub\_dat4)  
 leftout4[i]=predict(m\_sub4,newdat=scaff[i,])   
}  
  
R2M4<-R2(leftout4,scaff$JOTR\_all)  
rmseM4<-rmse(leftout4,scaff$JOTR\_all)

### Step 7

Create a table to compare values

modcomp <- matrix(c(rmseM1,R2M1,rmseM2,R2M2,rmseM3,R2M3,rmseM4,R2M4),ncol=2,byrow=TRUE)  
 colnames(modcomp) <- c("Root Mean Square Error","R^2")  
 rownames(modcomp) <- c("M1","M2","M3","M4")  
 modcomp <- as.table(modcomp)  
 modcomp

## Root Mean Square Error R^2  
## M1 1.444810e+05 3.620844e-01  
## M2 1.444810e+05 2.718040e-01  
## M3 1.444810e+05 1.883733e-01  
## M4 1.444810e+05 2.852881e-01

Lets do one more looking at themost significant variables

M5<-glm.nb(JOTR\_all~stdcci+stdcochella+stdgas, data=scaff)  
   
 leftout5<-rep(NA,times=length(scaff$JOTR\_all))  
   
 for(i in 1:length(scaff$JOTR\_all)){  
 sub\_dat5<-scaff[-i,]  
 m\_sub5<-glm.nb(JOTR\_all~stdcci+stdcochella+stdgas, data=sub\_dat5)  
 leftout5[i]=predict(m\_sub5,newdat=scaff[i,])   
 }  
   
 R2M5<-R2(leftout5,scaff$JOTR\_all)  
 rmseM5<-rmse(leftout5,scaff$JOTR\_all)  
  
 modcomp <- matrix(c(rmseM1,R2M1,rmseM2,R2M2,rmseM3,R2M3,rmseM4,R2M4,rmseM5,R2M5),ncol=2,byrow=TRUE)  
 colnames(modcomp) <- c("Root Mean Square Error","R^2")  
 rownames(modcomp) <- c("M1","M2","M3","M4","M5")  
 modcomp <- as.table(modcomp)  
 modcomp

## Root Mean Square Error R^2  
## M1 1.444810e+05 3.620844e-01  
## M2 1.444810e+05 2.718040e-01  
## M3 1.444810e+05 1.883733e-01  
## M4 1.444810e+05 2.852881e-01  
## M5 1.444810e+05 3.846582e-01