BIAS VARIANCE TRADEOFF IN R - Assignment 12

Sharon Morris

4/17/2017

Using the stats and boot libraries in R perform a cross-validation experiment to observe the bias variance tradeoff. You'll use the auto data set from previous assignments. This dataset has 392 observations across 5 variables.

```
library(stats)
library(boot)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

Load Data

```
autoData <- read.table(
    "https://raw.githubusercontent.com/indianspice/IS605/master/Hw/auto-mpg.data",
    header = FALSE, as.is = TRUE)

colnames(autoData) <- c("displacement", "horsepower","weight","acceleration","mpg")

head(autoData)
```

```
##   displacement horsepower weight acceleration mpg
## 1          307        130   3504         12.0  18
## 2          350        165   3693         11.5  15
## 3          318        150   3436         11.0  18
## 4          304        150   3433         12.0  16
## 5          302        140   3449         10.5  17
## 6          429        198   4341         10.0  15
```

Explore the Data

```
sum(is.na(autoData))
```

```
## [1] 0
```

```
str(autoData)
```

```
## 'data.frame':    392 obs. of  5 variables:
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
```

Polynomial Fit Model Fit a polynomial model of various degrees using the glm function in R and then measure the cross validation error using cv.glm function.

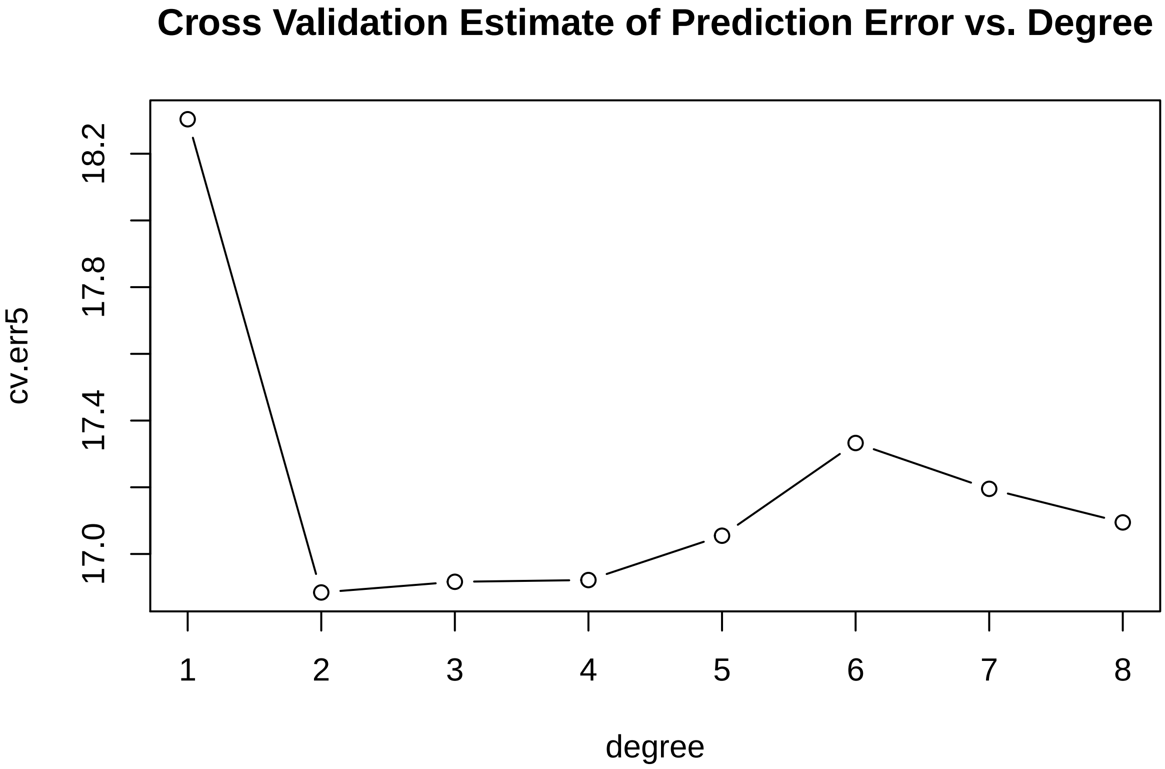
```
n <- 8
degree <- 1:n
cv.err5 <- numeric()
for (i in degree) {
  glm.fit <- glm(mpg ~ poly(displacement + horsepower + weight + acceleration,
                      mpg = i), data = autoData)
  cv.err5[i - min(degree) + 1] <- cv.glm(autoData, glm.fit, K = 5)$delta[1]
}
summary(glm.fit)
```

```
## 
## Call:
## glm(formula = mpg ~ poly(displacement + horsepower + weight + 
##     acceleration, mpg = i), data = autoData)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -13.0921   -2.5604   -0.5072    1.7661   16.2801  
## 
## Coefficients:
##                                                                    Estimate
## (Intercept)                                                         23.4459
## poly(displacement + horsepower + weight + acceleration, mpg = i)1 -129.0809
## poly(displacement + horsepower + weight + acceleration, mpg = i)2   24.5509
## poly(displacement + horsepower + weight + acceleration, mpg = i)3   -0.6179
## poly(displacement + horsepower + weight + acceleration, mpg = i)4   -2.3957
## poly(displacement + horsepower + weight + acceleration, mpg = i)5    2.8573
## poly(displacement + horsepower + weight + acceleration, mpg = i)6   -3.7582
## poly(displacement + horsepower + weight + acceleration, mpg = i)7    6.9607
## poly(displacement + horsepower + weight + acceleration, mpg = i)8   -1.3778
##                                                                   Std. Error
## (Intercept)                                                           0.2077
## poly(displacement + horsepower + weight + acceleration, mpg = i)1     4.1119
## poly(displacement + horsepower + weight + acceleration, mpg = i)2     4.1119
## poly(displacement + horsepower + weight + acceleration, mpg = i)3     4.1119
## poly(displacement + horsepower + weight + acceleration, mpg = i)4     4.1119
## poly(displacement + horsepower + weight + acceleration, mpg = i)5     4.1119
## poly(displacement + horsepower + weight + acceleration, mpg = i)6     4.1119
## poly(displacement + horsepower + weight + acceleration, mpg = i)7     4.1119
## poly(displacement + horsepower + weight + acceleration, mpg = i)8     4.1119
##                                                                   t value
## (Intercept)                                                       112.894
## poly(displacement + horsepower + weight + acceleration, mpg = i)1 -31.392
## poly(displacement + horsepower + weight + acceleration, mpg = i)2   5.971
## poly(displacement + horsepower + weight + acceleration, mpg = i)3  -0.150
## poly(displacement + horsepower + weight + acceleration, mpg = i)4  -0.583
## poly(displacement + horsepower + weight + acceleration, mpg = i)5   0.695
## poly(displacement + horsepower + weight + acceleration, mpg = i)6  -0.914
## poly(displacement + horsepower + weight + acceleration, mpg = i)7   1.693
## poly(displacement + horsepower + weight + acceleration, mpg = i)8  -0.335
##                                                                   Pr(>|t|)
## (Intercept)                                                        < 2e-16
## poly(displacement + horsepower + weight + acceleration, mpg = i)1  < 2e-16
## poly(displacement + horsepower + weight + acceleration, mpg = i)2 5.39e-09
## poly(displacement + horsepower + weight + acceleration, mpg = i)3   0.8806
## poly(displacement + horsepower + weight + acceleration, mpg = i)4   0.5605
## poly(displacement + horsepower + weight + acceleration, mpg = i)5   0.4875
## poly(displacement + horsepower + weight + acceleration, mpg = i)6   0.3613
## poly(displacement + horsepower + weight + acceleration, mpg = i)7   0.0913
## poly(displacement + horsepower + weight + acceleration, mpg = i)8   0.7378
```

```
##
## (Intercept)                                                       ***
## poly(displacement + horsepower + weight + acceleration, mpg = i)1 ***
## poly(displacement + horsepower + weight + acceleration, mpg = i)2 ***
## poly(displacement + horsepower + weight + acceleration, mpg = i)3
## poly(displacement + horsepower + weight + acceleration, mpg = i)4
## poly(displacement + horsepower + weight + acceleration, mpg = i)5
## poly(displacement + horsepower + weight + acceleration, mpg = i)6
## poly(displacement + horsepower + weight + acceleration, mpg = i)7 .
## poly(displacement + horsepower + weight + acceleration, mpg = i)8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 16.90757)
##
##     Null deviance: 23819.0  on 391  degrees of freedom
## Residual deviance:  6475.6  on 383  degrees of freedom
## AIC: 2231.8
##
## Number of Fisher Scoring iterations: 2
```

Once you have fit the various polynomials from degree 1 to 8, you can plot the cross- validation error function as

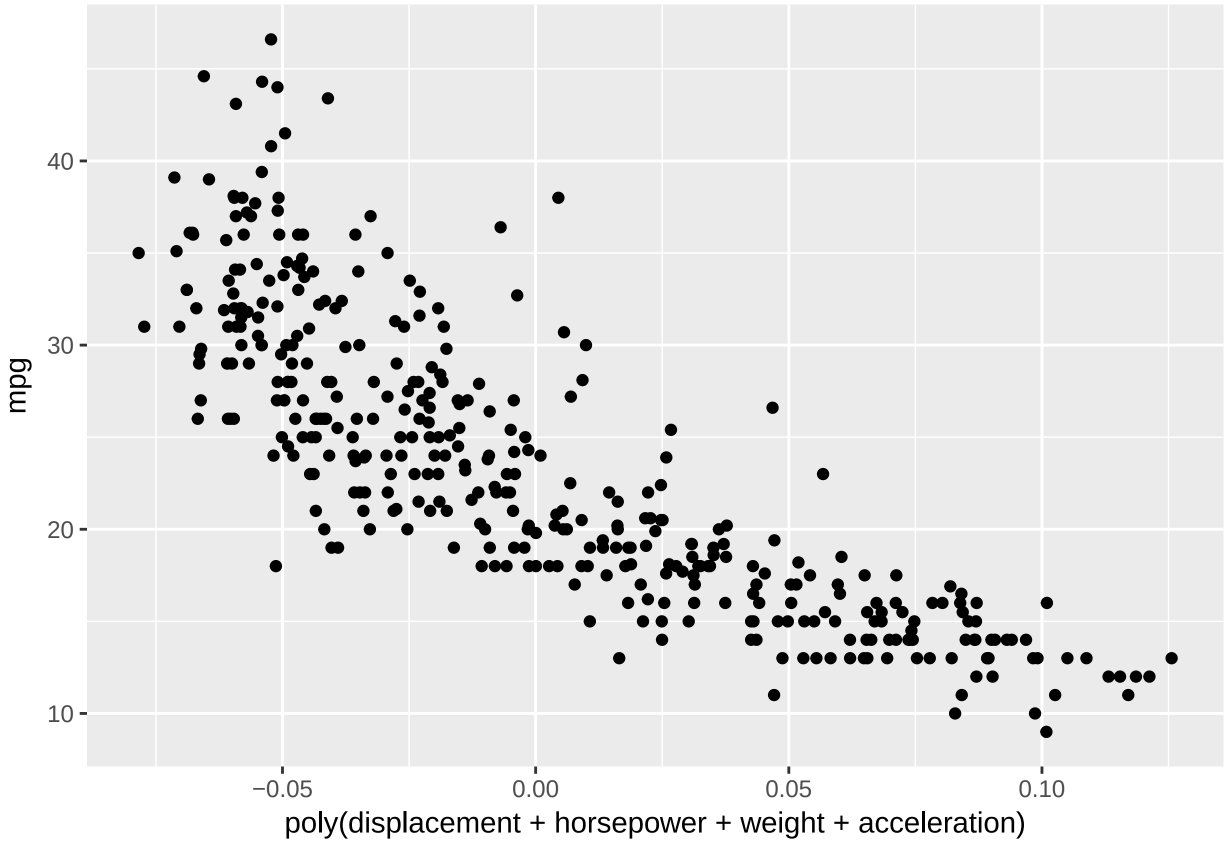
```
plot(degree,cv.err5,type='b', main = "Cross Validation Estimate of Prediction Error vs. Degree")
```



The graph below shows that a degree 2 or 3 polynomial seems to fit the model the closest.

```
ggplot(autoData, aes(x = poly(displacement + horsepower + weight + acceleration),
                     y = mpg)) + geom_point()
```

```
## Don't know how to automatically pick scale for object of type poly/matrix. Defaulting to continuous.
```



Reference https://www.r-bloggers.com/cross-validation-estimating-prediction-error/