DA 605 - Assignment 12

Dan Fanelli

BIAS VARIANCE TRADEOFF IN R

- Use the stats and boot libraries
- 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.
- We want to:
  - fit a polynomial model of various degrees using the glm function in R
  - measure the cross validation error using cv.glm function.

```
library(boot)
library(stats)
library(knitr)

mpg_df <- read.table("auto-mpg.data", sep="")
# displacement, horsepower, weight, acceleration, mpg
names(mpg_df) <- c("disp", "hp", "wt", "acc", "mpg")
kable(head(mpg_df, n=10), align = "l")
```

disp	hp	wt	acc	mpg
307	130	3504	12.0	18
350	165	3693	11.5	15
318	150	3436	11.0	18
304	150	3433	12.0	16
302	140	3449	10.5	17
429	198	4341	10.0	15
454	220	4354	9.0	14
440	215	4312	8.5	14
455	225	4425	10.0	14
390	190	3850	8.5	15

- Fit various polynomial models to compute mpg as a function of the other four variables using glm function:
  - acceleration
  - weight
  - horsepower
  - displacement
- For example, the following will fit a 2nd degree polynomial function between mpg and the remaining 4 variables and perform 5 iterations of cross-validations.
  - glm.fit=glm(mpg~poly(disp+hp+wt+acc,2), data=auto)
  - cv.err5[2] = cv.glm(auto,glm.fit,K=5)$delta[1]
- This result will be stored in a cv.err5 array.

- cv.glm returns the estimated cross validation error and its adjusted value in a variable called delta. * Please see the help on cv.glm to see more information.
- Once you have fit the various polynomials from degree 1 to 8, you can plot the crossvalidation error function as degree=1:8, plot(degree,cv.err5,type='b')
- For you assignment, please create an R-markdown document where you load the auto data set, perform the polynomial fit and then plot the resulting 5 fold cross validation curve. Your output should show the characteristic U-shape illustrating the tradeoff between bias and variance.

```
degree_errors = c()
degree_values = c(1:8)
for(degree in degree_values)
{
  glm.fit=glm(mpg~poly(disp+hp+wt+acc,2), data=mpg_df)
  deg_err <- cv.glm(mpg_df, glm.fit, K=5)$delta[1]
  degree_errors[degree] <- deg_err
}

plot(degree_values, degree_errors, type='b')
```