Bruce Campbell ST-617 Homework 2

Tue Jul 05 18:47:16 2016

Chapter 5

Problem 8

We will now perform cross-validation on a simulated data set. ### a) Generate a simulated data set as follows:

```r
set.seed(1)
# y=rnorm (100) #<------------- Not sure why this is necessary
x = rnorm(100)
y = x - 2 * x^2 + rnorm(100)
```

In this data set, what is n and what is p?

$n = 100$ and $p = 2$

Write out the model used to generate the data in equation form.

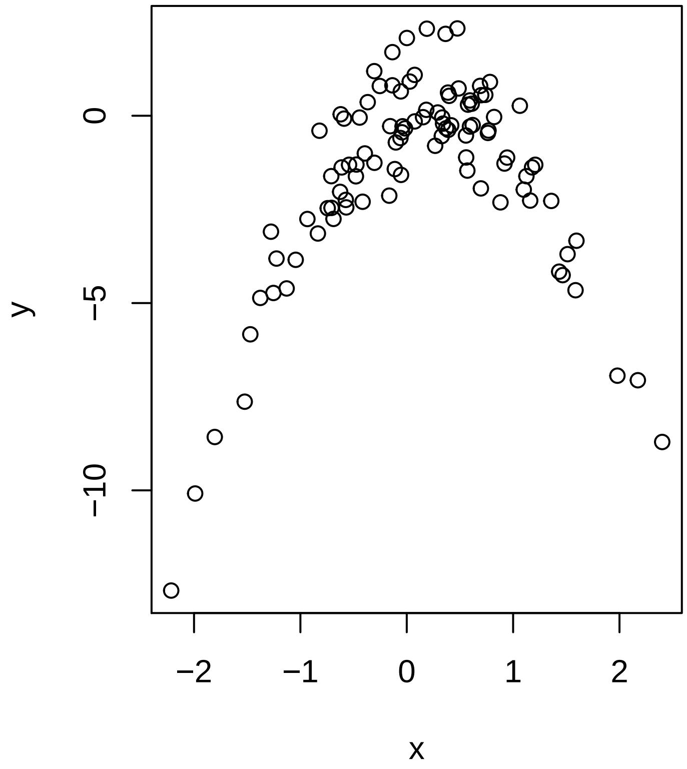
$$Y = \beta_1 X + \beta_2 X^2 + \epsilon$$

Where $\beta_1 = 1$, $\beta_2 = -2$, and $\epsilon = N(0, 1)$

b)

Create a scatterplot of X against Y . Comment on what you find.

```r
plot(x, y)
```



We see the quadratic realtionship described in the model corrupted by the noise.

c)

Set a random seed, and then compute the LOOCV errors that result from fitting the following four models using least squares:

i.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

ii.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$

iii.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

iv.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \epsilon$$

```
set.seed(17)
library(boot)

loocv_rates <- data.frame(model = character(), LOOCV_ERROR_delta1 = numeric(),
    LOOCV_ERROR_delta2 = numeric())
```

```r
DF <- data.frame(X = x, Y = y)

glm.fit_1 <- glm(Y ~ X, data = DF)
coef(glm.fit_1)
```

```
## (Intercept)           X 
##   -1.625427    0.692497
```

```r
cv.err_1 <- cv.glm(DF, glm.fit_1)
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X", LOOCV_ERROR_delta1 = cv.err_1$delta[1],
    LOOCV_ERROR_delta2 = cv.err_1$delta[2]))

glm.fit_2 <- glm(Y ~ X + I(X^2), data = DF)
coef(glm.fit_2)
```

```
## (Intercept)           X      I(X^2) 
##  0.05671501  1.01716087 -2.11892120
```

```r
cv.err_2 <- cv.glm(DF, glm.fit_2)
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X+X^2", LOOCV_ERROR_delta1 = cv.err_2$delta[1],
    LOOCV_ERROR_delta2 = cv.err_2$delta[2]))


glm.fit_3 = glm(Y ~ X + I(X^2) + I(X^3), data = DF)
cv.err_3 = cv.glm(DF, glm.fit_3)
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X+X^2+X^3", LOOCV_ERROR_delta1 = cv.err_3$delta
    LOOCV_ERROR_delta2 = cv.err_3$delta[2]))


glm.fit_4 = glm(Y ~ X + I(X^2) + I(X^3) + I(X^4), data = DF)
cv.err_4 = cv.glm(DF, glm.fit_4)
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X+X^2+X^3+X^4", LOOCV_ERROR_delta1 = cv.err_4$de
    LOOCV_ERROR_delta2 = cv.err_4$delta[2]))

library(pander)
pander(loocv_rates)
```

model	LOOCV_ERROR_delta1	LOOCV_ERROR_delta2
Y~X	7.288	7.285
Y~X+X^2	0.9374	0.9372
Y~X+X^{2+X}3	0.9566	0.9563
Y~X+X^{2+X}3+X^4	0.9539	0.9534

d)

Repeat (c) using another random seed, and report your results. Are your results the same as what you got in (c)? Why?

```r
set.seed(173)
library(boot)

loocv_rates <- data.frame(model = character(), LOOCV_ERROR_delta1 = numeric(),
    LOOCV_ERROR_delta2 = numeric())
DF <- data.frame(X = x, Y = y)

glm.fit_1 <- glm(Y ~ X, data = DF)
coef(glm.fit_1)
```

```
## (Intercept)           X
##   -1.625427    0.692497
```

```r
cv.err_1 <- cv.glm(DF, glm.fit_1)
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X", LOOCV_ERROR_delta1 = cv.err_1$delta[1],
    LOOCV_ERROR_delta2 = cv.err_1$delta[2]))

glm.fit_2 <- glm(Y ~ X + I(X^2), data = DF)
coef(glm.fit_2)
```

```
## (Intercept)           X      I(X^2)
##  0.05671501  1.01716087 -2.11892120
```

```r
cv.err_2 <- cv.glm(DF, glm.fit_2)
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X+X^2", LOOCV_ERROR_delta1 = cv.err_2$delta[1],
    LOOCV_ERROR_delta2 = cv.err_2$delta[2]))


glm.fit_3 = glm(Y ~ X + I(X^2) + I(X^3), data = DF)
cv.err_3 = cv.glm(DF, glm.fit_3)
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X+X^2+X^3", LOOCV_ERROR_delta1 = cv.err_3$delta
    LOOCV_ERROR_delta2 = cv.err_3$delta[2]))


glm.fit_4 = glm(Y ~ X + I(X^2) + I(X^3) + I(X^4), data = DF)
cv.err_4 = cv.glm(DF, glm.fit_4)
summary(glm.fit_4)
```

```
##
## Call:
## glm(formula = Y ~ X + I(X^2) + I(X^3) + I(X^4), data = DF)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0550  -0.6212  -0.1567   0.5952   2.2267
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.156703   0.139462   1.124    0.264
## X            1.030826   0.191337   5.387 5.17e-07 ***
## I(X^2)      -2.409898   0.234855 -10.261  < 2e-16 ***
## I(X^3)      -0.009133   0.067229  -0.136    0.892
## I(X^4)       0.069785   0.053240   1.311    0.193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9197797)
##
##     Null deviance: 700.852  on 99  degrees of freedom
## Residual deviance:  87.379  on 95  degrees of freedom
## AIC: 282.3
##
## Number of Fisher Scoring iterations: 2
```

```
loocv_rates <- rbind(loocv_rates, data.frame(model = "Y~X+X^2+X^3+X^4", LOOCV_ERROR_delta1 = cv.err_4$de
    LOOCV_ERROR_delta2 = cv.err_4$delta[2]))

library(pander)
pander(loocv_rates)
```

model	LOOCV_ERROR_delta1	LOOCV_ERROR_delta2
Y~X	7.288	7.285
Y~X+X^2	0.9374	0.9372
Y~X+X^{2+X}3	0.9566	0.9563
Y~X+X^{2+X}3+X^4	0.9539	0.9534

These are the same results. The reason for this is that the LOOCV algorithm is deterministic. It trains n models with n-1 training points reserving the nth point as a test point. There is no random splitting of the training and test data.

e)

Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

The model with the lowest LOOCV error rate is the quadratic model. This is as expected since the data was generated via a quadratic relationship.

f)

Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in (c) using least squares. Do these results agree with the conclusions drawn based on the cross-validation results?

The p-values for the third and fourth coefficient are not significant. This is consistent with the cross validation results where the quadratic model had the lowest error, and