Introduction to Statistical Learning 6/19/2017

Chapter 3: Linear Regression

Name: roninlaw

Email: roninlaw

1

Based on the p-values we know that there is strong evidence that TV and Radio play an important tole in the prediction of sales, while Newspapers do not.

2

kNN Regression: For a given point x_0 it is the average of the k closest points
kNN Classification: x_0 is assigned to the class that the majority of the k closest points fall in

3

(a) *(iii)*: If IQ/GPA are kept constant then we simply have to examine $(\hat{\beta}_3 + \hat{\beta}_5 \cdot \text{GPA}) \cdot \text{Gender}$. If GPA > 3.5, then the term will be negative for females. The term is always 0 for males, so we know that as long as GPA is sufficiently high it will result in males making a higher salary.

(b) $\hat{y} = 50 + 20(4.0) + .07(110) + 35(1) + .01(4.0)(110) - 10(4.0)(1) = 137.1$

(c) False. To make this claim we would have to see the corresponding p-value to the coefficient

4

(a) We would expect the RSS of the cubic regression to be lower because the model is more flexible, and more flexible models capture more of the error in training data.

(b) We would expect the RSS of the linear model to be lower because the true data is closer to linear than it is to cubic.

(c) We would expect the RSS of the cubic regression to be lower for the same reason as (a)

(d) Because we do not know what the true distribution is like we do not have enough information to answer the questiont

5

$$
\begin{aligned}
\hat{y} &= x_i\hat{\beta} \\
&= x_i\frac{\sum_{j=1}^{n} x_j y_j}{\sum_{k=1}^{n} x_k^2} \\
&= \frac{x_i(x_1y_1 + \cdots + x_ny_n)}{x_1^2 + \ldots x_n^2} \\
&= \frac{(x_ix_1)y_1}{x_1^2 + \cdots x_n^2} + \frac{(x_ix_2)y_2}{x_1^2 + \cdots x_n^2} + \cdots + \frac{(x_ix_n)y_n}{x_1^2 + \cdots x_n^2} \\
&= \sum_{j=1}^{n} a_j y_j \text{ where } a_j = \frac{x_ix_j}{\sum_{j=1}^{n} x_j^2}
\end{aligned} \tag{1}
$$

6

$$
\begin{aligned}
\hat{y} &= \hat{\beta}_0 + \hat{\beta}_1 \bar{x} \\
&= (\bar{y} - \hat{\beta}_1 \bar{x}) + \beta_1 \bar{x} \\
&= \bar{y}
\end{aligned}
$$

8

```r
Auto = read.csv("Auto.csv", header=T, na.strings="?")
summary(Auto)
lm.fit = lm(mpg~horsepower,data=Auto)
predict(lm.fit,data.frame(horsepower=c(98)),interval="prediction")
predict(lm.fit,data.frame(horsepower=c(98)),interval="confidence")
```

(a) (i) There is a relationship between the predictor and the response.

(ii) The R-square value tells us that the relationship is not that strong.

(iii) The relationship between the predictor and the response is negative.

(iv)
- $\hat{f}(98) = 39.9359 - .1578(98) = 24.4715$
- Prediction: [14.81, 34.12]
- Confidence: [23.97, 24.96]

(b)

```r
attach(Auto)
plot(horsepower,mpg)
abline(lm.fit,col="red")
```

(c)

```r
par(mfrow=c(2,2))
plot(lm.fit)
```

One problem with the plot is the lack of linearity in the true data, which implies there might be a better way to model the data.

9

(a)

```r
pairs(Auto)
```

(b)

```r
cor(subset(Auto,select=-name))
```

(c)

```r
summary(lm(mpg~.-name,data=Auto))
```

(i) There is a relationship between the predictors, and the response. Because we are regressing on a lot of variables we examine the F-score, and the corresponding p-value. Because the F-statistic is significantly greater than 1, and the p-value is very small we conclude that there is a relationship.

(ii) Weight, year, and origin seem to have a statistically significant relationship to the response.

(iii) The coefficient for the year variable tells us that there is a positive relationship between an increase in year and number of miles per gallon.

(d)

```
par(mfrow=c(2,2))
plot(lm(mpg~.-name,data=Auto))
```

The residual plots do indicate that there are a fairly large amount of points with unusually large outliers. The leverage plot does indicate that there are points with high leverage. Specifically, the right most point in the bottom right plot.

(e)

```
lm.i1 = lm(formula = mpg ~ cylinders*horsepower) # * adds terms for the individual
    components
lm.i2 = lm(formula = mpg ~ weight + year + cylinders:weight) # : adds interaction, but
    does not add individual terms
lm.i3 = lm(formula = mpg ~ cylinders*weight + displacement*cylinders +
    displacement*weight)
```

The first two interaction models were done using random interaction terms, while the third interaction model was made using terms that were highly correlated based on the correlation matrix from part (c). The second model had the lowest R^2, and all three models had high F-statistics and low corresponding p-values.

(f)

```
lm.i3 = lm(formula = mpg ~ I(weight^2) + weight + log(weight))
lm.i4 = lm(formula = mpg ~ weight + I(weight^2))
```

The first interaction model was made by randomly transforming the weight variable. I chose this variable because it was the variable most highly correlated with mpg. This model ended up having a reasonably high F-statistic value with a low corresponding p-value, but the p-values for the individual coefficients was quite low. The second model also ended up having a high F-statistic value with a low corresponding p-value, but the individual p-values for the coefficients were much lower.

10

(a)

```
Carseats = read.csv("Carseats.csv", header=T, na.strings="?")
names(Carseats) # checking variable names
lm.fit = lm(Sales ~ Price + Urban + US, data = Carseats)
summary(lm.fit)
```

(b)

- Price: The low p-value, and sign of the coefficient suggests that there is a negative relationship between the price of an item and sales when UrbanYes and USYes factors are present.
- UrbanYes: The high p-value suggests that there is not a relationship between sales and where a store is located.
- USYes: The low p-value, and sign of the coefficient suggests that there is a positive relationship between whether or not a store is located in the United States and sales when Price and UrbanYes factors are present.

(c) $\text{Sales} = 13.04 - .05 \cdot \text{Price} - .02 \cdot \text{UrbanYes} + 1.20 \cdot \text{USYes}$

(d) Price and USYes

(e)

```
lm.fit2 = lm(Sales ~ Price + US, data = Carseats)
```

(f) The R^2 statistic tells us that the model in (e) is marginally better than the first model.

(g)

```
confint(lm.fit2)
```

(h)

```
par(mfrow=c(2,2))
plot(lm.fit2)
plot(predict(lm.fit2), rstudent(lm.fit2))
```

There is evidence of high leverage observations in the model. Specifically the rightmost point in the bottom right chart. There is not evidence of any outliers in the data

11

```
set.seed(1)
x = rnorm(100)
y = 2 * x + rnorm(100)
lm.fit = lm(y~x+0)
lm.fit2 = lm(x~y+0)
lm.fit3 = lm(y~x)
lm.fit4 = lm(x~y)
```

(a) The R^2 statistic tells us that the model is a good fit for the data.

- $\hat{\beta} = 1.99$
- SE($\hat{\beta}$) = .11
- t-statistic = 18.73
- p-value = 2e-16

(b) The R^2 statistic tells us that this model is also a good fit for the data.

- $\hat{\beta} = .39$
- SE($\hat{\beta}$) = .02
- t-statistic = 18.73
- p-value = 2e-16

(c) They are inverses of each other.

(d) Follows almost directly from rearranging $\frac{\hat{\beta}}{SE(\hat{\beta})}$. By using R's output for $\hat{\beta}$ and $SE(\hat{\beta})$ we also see that the equation holds.

(e) In the closed form of the t-statistic wherever we deal with some function of x_i we see it multiplied by a corresponding function of y_i which means the results will be the same.

(f)

```
lm.fit3 = lm(y~x)
lm.fit4 = lm(x~y) # resulting t-statistics are the same
```

12

(a) By looking at the denominator of the coefficient we see that the coefficients are equal when $x_i^2 = y_i^2$

(b)

```
x = rnorm(100)
y = x^2
lm.fit1 = lm(y~x+0)
lm.fit2 = lm(x~y+0)
```

(c)

```
x = rnorm(100)
y = x
lm.fit1 = lm(y~x+0)
lm.fit2 = lm(x~y+0)
```

13

(a)

```
x = rnorm(100)
```

(b)

```
eps = rnorm(n = 100, mean = 0, sd = .25^.5)
```

(c)

```
y = -1 + .5*x + eps
```

len$(y) = 100$, $\hat{\beta}_0 = -1$, and $\hat{\beta}_1 = .5$

(d)

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

The points are positively correlated.

(e)

```
lm.fit = lm(y~x)
```

$\hat{\beta}_0 = -1.02$, and $\hat{\beta}_1 = .50$. Both coefficients are very close to the true values, β_0 and β_1.

(f)

```
abline(lm.fit, col = "blue")
```

(g)

```
lm.fit2 = lm(y~ x + I(x^2))
summary(lm.fit2)
```

Based on the R^2 and RSE statistic there is evidence that the polynomial model is slightly better than the original model.

(h)

```
x1 = rnorm(100)
eps1 = rnorm(n = 100, mean = 0, sd = .05^.5)
y1 = -1 + .5*x1 + eps1
plot(x1,y1)
lm.fit1 = lm(y1~x1)
```

```
summary(lm.fit1)
abline(lm.fit1, col = "red")
```

$\beta_0 = -1, \beta_1 = .5, \hat{\beta_0} = -1.01, \hat{\beta_1} = .48$. The results are very similar to the original model that was built, but the points seem to hug the line of best fit more tightly. This makes sense since we decreased the variance from .25 to .05.

(i)

```
x2 = rnorm(100)
eps2 = rnorm(n = 100, mean = 0, sd = .95^.5)
y2 = -1 + .5*x2 + eps2
plot(x2,y2)
lm.fit2 = lm(y2~x2)
summary(lm.fit2)
abline(lm.fit2, col = "red")
```

$\beta_0 = -1, \beta_1 = .5, \hat{\beta_0} = -.98, \hat{\beta_1} = .59$. The results are similar to the original model that was built, but seem to be a lot more spread out. This makes sense since we increase the variance from .25 to .95

(j)

```
confint(lm.fit)
confint(lm.fit1)
confint(lm.fit2)
```

Examining the confidence intervals we see that as the variance increases the width of the intervals also increases. This makes intuitive sense since it will be harder to predict what the coefficients will be as we increase the noise.

14

(a)

```
set.seed(1)
x1 = runif(100)
x2 = .5*x1 + rnorm(100)/10
y = 2 + 2*x1 + .3*x2 + rnorm(100)
```

$Y = 2 + 2 \cdot X_1 + .3X_2 + \epsilon$. The regression coefficients are $\beta_0 = 2, \beta_{X_1} = 2, \beta_{X_2} = .3$

(b)

```
cor(x1,x2)
plot(x1,x2)
```

$\rho = .8351$

(c)

```
lm.fit = lm(y~x1+x2)
```