Bruce Campbell ST-617 Homework 2

Wed Jul 06 09:25:39 2016

Chapter 6

Problem 10

We have seen that as the number of features used in a model increases, the training error will necessarily decrease, but the test error may not. We will now explore this in a simulated data set. ### a) Generate a data set with p = 20 features, n = 1,000 observations, and an associated quantitative response vector generated according to the model $Y = X\beta + \epsilon$, where β has some elements that are exactly equal to zero.

```
rm(list = ls())
set.seed(7)
beta_v <- rnorm(20, 0, 1)
clamped_coeff = (beta_v < 0.5 & beta_v > -0.5)
beta_v[clamped_coeff] <- 0

X <- matrix(NA, nrow = 1000, ncol = 20)
Y <- matrix(NA, nrow = 1000, ncol = 1)
for (i in 1:1000) {
    x_i = rnorm(20, 0, 1)
    err = 0.1 * rnorm(1, 0, 1)
    Y_i = beta_v %*% x_i + err

    X[i, ] <- x_i

    Y[i] <- Y_i

}

DF <- as.data.frame(X)
DF <- cbind(DF, Y)
names(DF) = c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10", "X11",
    "X12", "X13", "X14", "X15", "X16", "X17", "X18", "X19", "X20", "Y")
```

b)

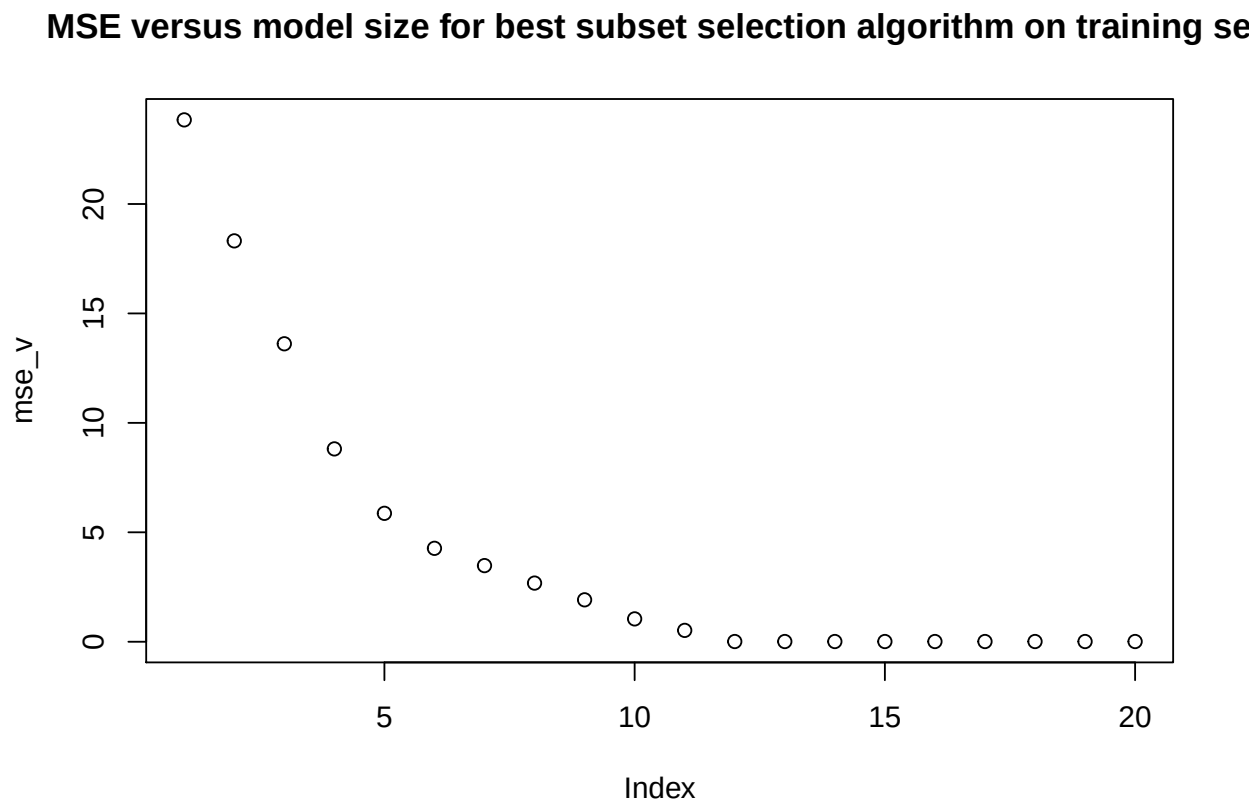
Split your data set into a training set containing 100 observations and a test set containing 900 observations.

```
train = sample(nrow(DF), 900)
DFTrain <- DF[train, ]
DFTest <- DF[-train, ]
```

c)

Perform best subset selection on the training set, and plot the training set MSE associated with the best model of each size.

```r
library(leaps)
attach(DFTrain)
regfit.full <- regsubsets(Y ~ ., data = DFTrain, nvmax = 20)
reg.summary <- summary(regfit.full)

mse_v <- reg.summary$rss/nrow(DFTrain)

plot(mse_v)
title("MSE versus model size for best subset selection algorithm on training set.")
```

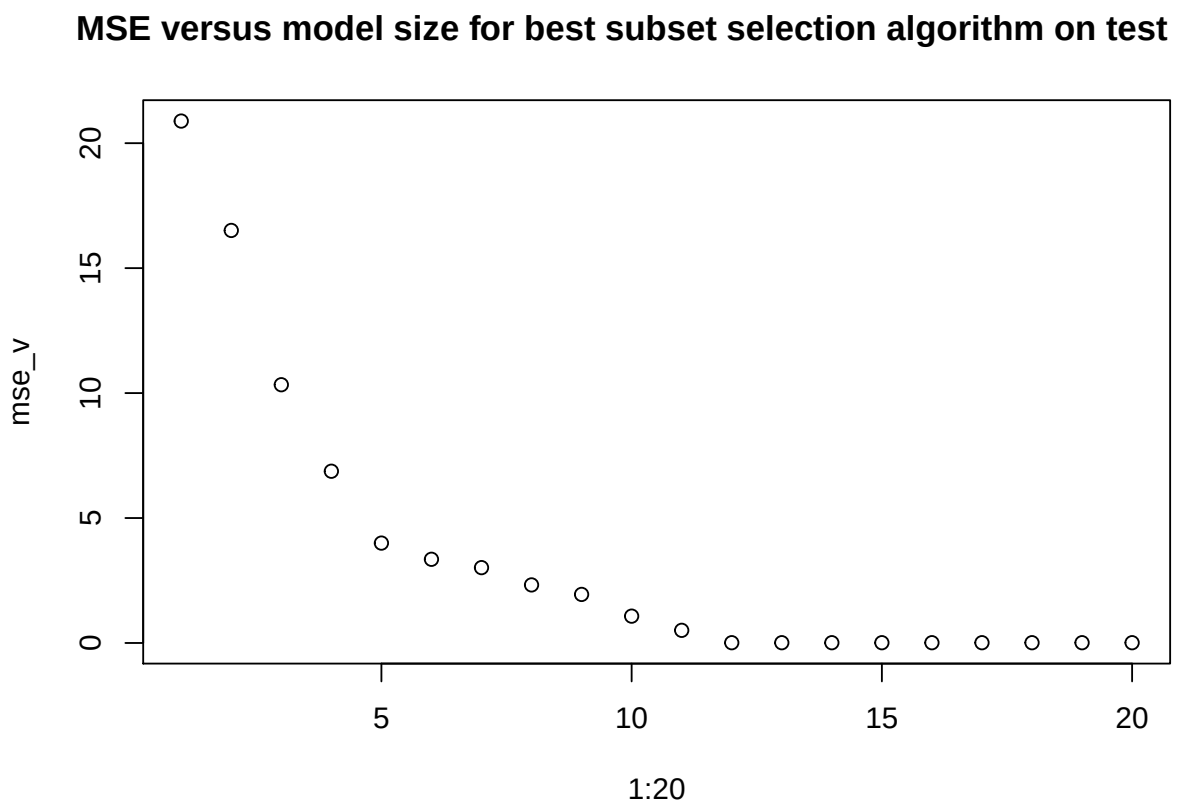


d)

Plot the test set MSE associated with the best model of each size.

```r
test_set_size <- 100
mse_v = matrix(NA, 1, 20)
for (i in 1:20) {
    beta_subset <- data.frame(coef(regfit.full, i))

    mse_subset <- 0
    for (j in 1:test_set_size) {
        X_j = DFTest[, !(colnames(DFTest) %in% c("Y"))]
```

```
        X_j = X_j[j, ]

        Y_j = DFTest$Y[j]

        intercept = beta_subset["(Intercept)", ]

        coeff_names <- rownames(beta_subset)

        coeff_names <- coeff_names[-1]

        coeff_x <- as.matrix(beta_subset[-1, ])
        mse_v
        X_red <- X_j[, colnames(X_j) %in% coeff_names]

        X_red <- as.matrix(X_red)

        V1 = matrix(NA, 1, length(coeff_names))
        V2 = matrix(NA, length(coeff_names), 1)
        V1 = coeff_x
        V2 = X_red

        Yhat_j = intercept + V2 %*% V1

        mse_subset <- mse_subset + (Yhat_j - Y_j)^2
    }
    mse_subset <- mse_subset/test_set_size
    mse_v[i] = mse_subset
}
plot(1:20, mse_v)
title("MSE versus model size for best subset selection algorithm on test set.")
```



e)

For which model size does the test set MSE take on its minimum value? Comment on your results. If it takes on its minimum value for a model containing only an intercept or a model containing all of the features, then play around with the way that you are generating the data in (a) until you come up with a scenario in which the test set MSE is minimized for an intermediate model size.

```
min_mse_model_index <- which.min(mse_v)
library(pander)
coeff_min <- coef(regfit.full, min_mse_model_index)
pander(coeff_min)
```

Table 1: Table continues below

(Intercept)	X1	X2	X3	X5	X6	X7	X10	X11	X12
-0.0006906	2.289	-1.19	-0.6916	-0.9697	-0.9397	0.748	2.194	-0.001764	2.718

X13	X14	X15	X17	X18	X19	X20
2.28	-0.001799	1.894	-0.8938	-0.005165	0.006309	0.994

The 12th subset is the one with the minimum MSE.

f)

How does the model at which the test set MSE is minimized compare to the true model used to generate the data? Comment on the coefficient values.

```
V <- beta_v
V <- as.data.frame(V)
rownames(V) = c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10",
    "X11", "X12", "X13", "X14", "X15", "X16", "X17", "X18", "X19", "X20")
pander(t(V))
```

Table 3: Table continues below

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
V	2.287	-1.197	-0.6943	0	-0.9707	-0.9473	0.7481	0	0	2.19	0

	X12	X13	X14	X15	X16	X17	X18	X19	X20
V	2.717	2.281	0	1.896	0	-0.8938	0	0	0.9882

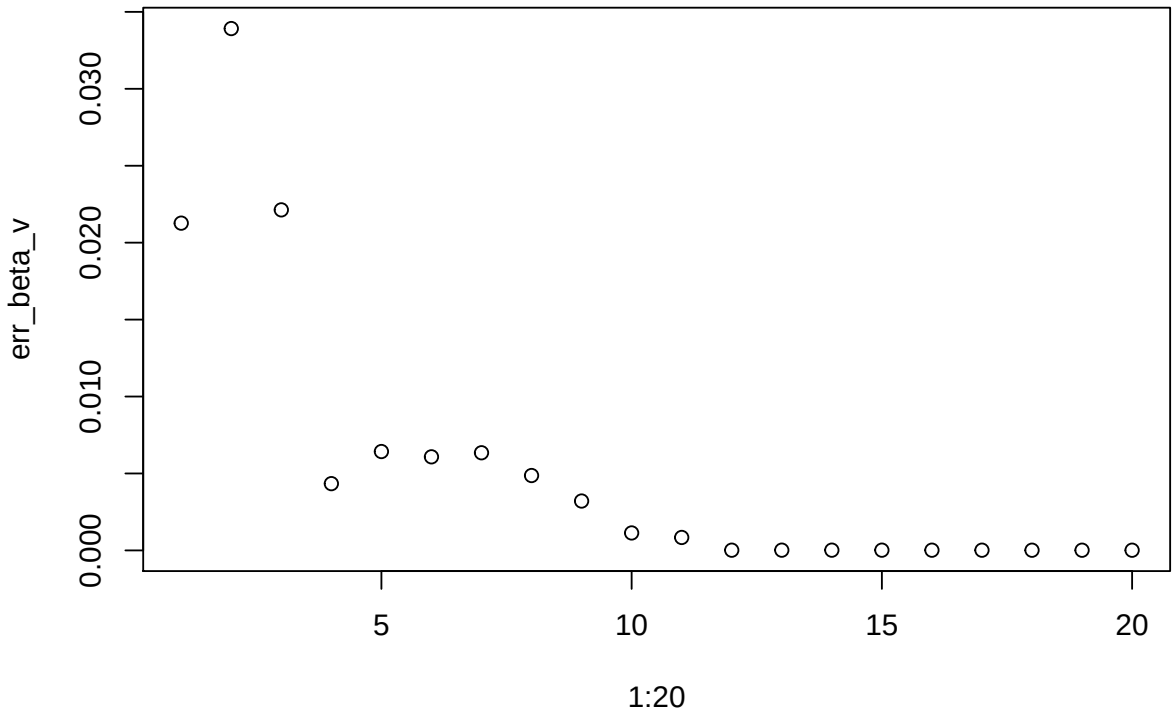
Interestingly, we see that the best subset with minimum MSE has exactly the same non-zero features we used to generate the data.

g)

Create a plot displaying $\sqrt{\sum(\beta_j - \hat{\beta}_j^r)^2}$ for a range of values of r, where $\hat{\beta}_j^r$ is the jth coefficient estimate for the best model containing r coefficients. Comment on what you observe. How does this compare to the test MSE plot from (d)?

```
err_beta_v = matrix(NA, 1, 20)
betaDF <- as.data.frame(V)
rownames(betaDF) = c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10",
    "X11", "X12", "X13", "X14", "X15", "X16", "X17", "X18", "X19", "X20")
betaDF <- t(betaDF)

for (i in 1:20) {
    beta_subset <- data.frame(coef(regfit.full, i))

    err_beta_subset <- 0

    intercept = beta_subset["(Intercept)", ]

    coeff_names <- rownames(beta_subset)

    coeff_names <- coeff_names[-1]
```

```
    coeff_x <- as.matrix(beta_subset[-1, ])

    beta_red <- betaDF[, colnames(betaDF) %in% coeff_names]

    for (j in 1:length(coeff_names)) {
        Beta_j = betaDF[, coeff_names[j]]

        BetaHat_j = beta_subset[coeff_names[j], ]
        err_beta_subset <- err_beta_subset + (Beta_j - BetaHat_j)^2
    }
    err_beta_subset <- err_beta_subset/length(coeff_names)
    err_beta_v[i] = err_beta_subset
}
plot(1:20, err_beta_v)
```



The error in the coefficients decreases as model size increases, but we see that it is not a monotonic decrease.

```
which.min(err_beta_v)
```

```
## [1] 20
```

The plots do look similar. The minimum error is achieved on the full model.