v.Excercise 6.8.10

Vidhi Shah, Sahil Shah, Kaarthik Sundaramoorthy

6/27/2020

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 β + ϵ, where β has some elements that are exactly equal to zero.

```
library(leaps)
set.seed(1)
n = 1000
p = 20
x <- matrix(rnorm(n * p), n, p)
b <- rnorm(p)
b[2]<-0
b[5]<-0
b[18]<-0
b[12]<-0
eps <- rnorm(n)
y <- x %*% b + eps
```

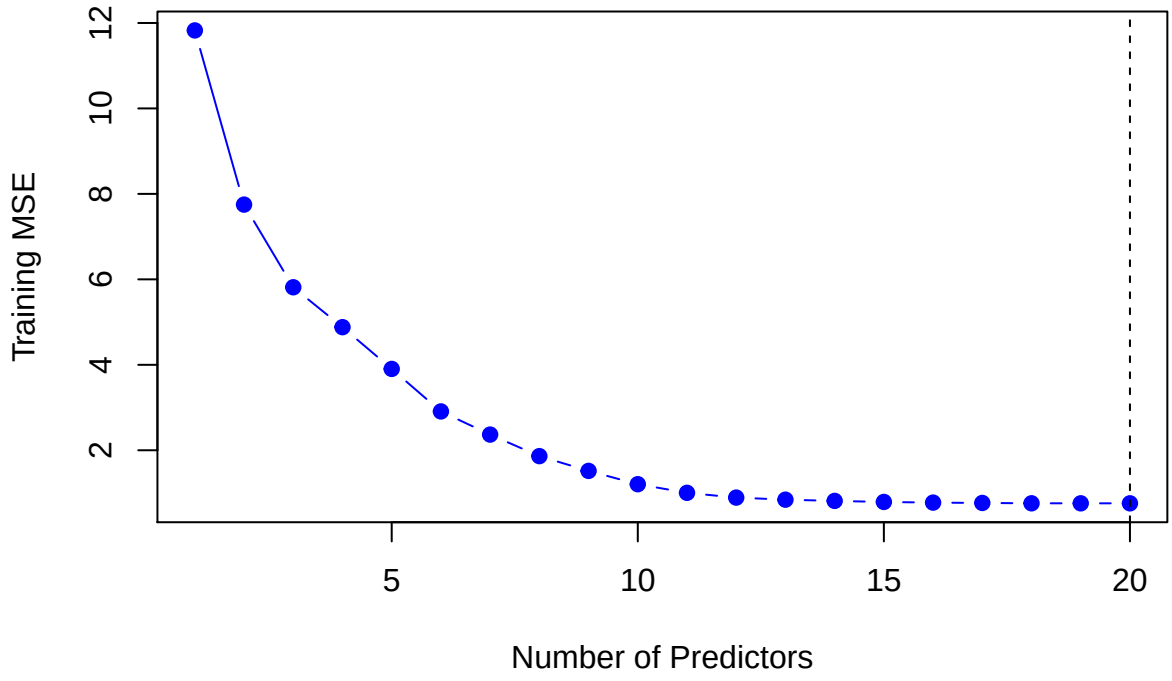
(b) Split your data set into a training set containing 100 observations and a test set containing 900 observations.

```
train <- sample(seq(1000), 100, replace = FALSE)
test <- (!train)
y.train = y[train, ]
y.test = y[-train, ]
x.train = x[train, ]
x.test = x[-train, ]
```

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

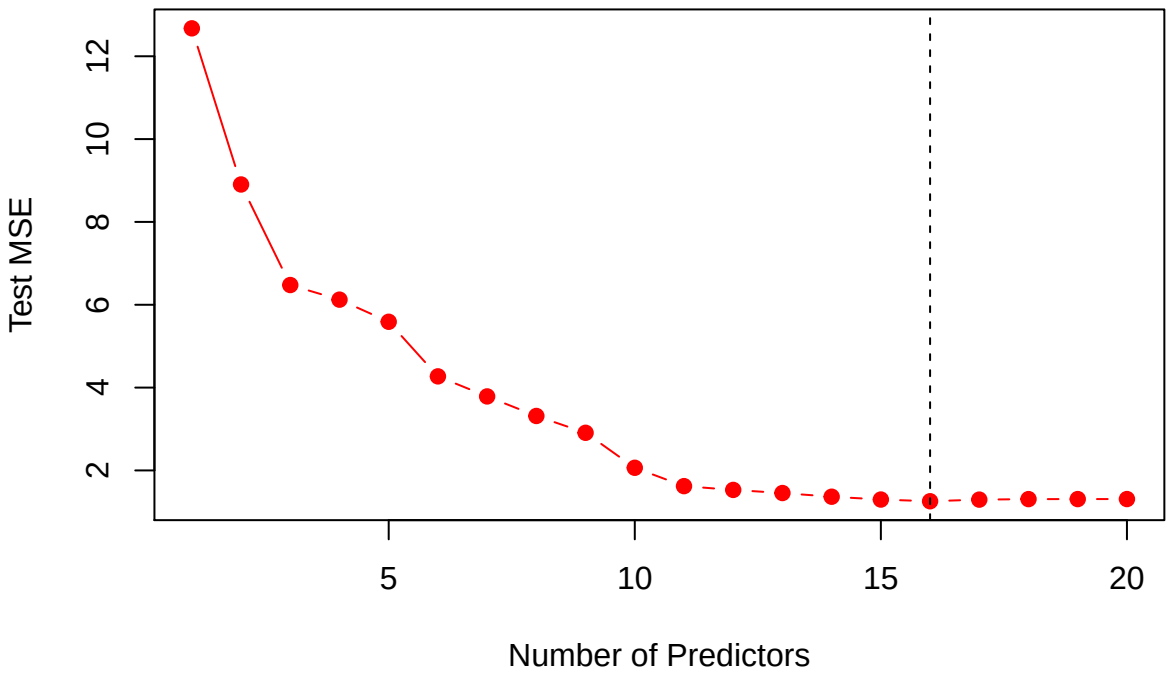
```
library(leaps)
regfit.full = regsubsets(y ~ ., data = data.frame(x = x.train, y = y.train),
    nvmax = p)
val.errors = rep(NA, p)
x_cols = colnames(x, do.NULL = FALSE, prefix = "x.")
for (i in 1:p) {
```

```
    coefi = coef(regfit.full, id = i)
    pred = as.matrix(x.train[, x_cols %in% names(coefi)]) %*% coefi[names(coefi) %in%
        x_cols]
    val.errors[i] = mean((y.train - pred)^2)
}
plot(val.errors, xlab = "Number of Predictors" ,ylab = "Training MSE",
     col = "blue" , pch = 19, type = "b")
abline(v = which.min(val.errors), lty=2)
```



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

```
test.errors = rep(NA, p)
for (i in 1:p) {
    coefi = coef(regfit.full, id = i)
    pred = as.matrix(x.test[, x_cols %in% names(coefi)]) %*% coefi[names(coefi) %in%
        x_cols]
    test.errors[i] = mean((y.test - pred)^2)
}
plot(test.errors, xlab = "Number of Predictors", ylab = "Test MSE",
     col = "red", pch = 19, type = "b")
abline(v = which.min(test.errors), lty = 2)
```



(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.

```
which.min(test.errors)
```

```
## [1] 16
```

16 parameter model has the smallest test 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.

```
#coef(regfit.full, which.min(val.errors))[-1]
names(b) <- colnames(x)[2:20]
True.b = b[b!=0]
True.b = c(`(Intercept)` = 0, True.b)
Fitted = coef(regfit.full, id = 16)
rbind(True.b, Fitted, Err = (True.b-Fitted))
```

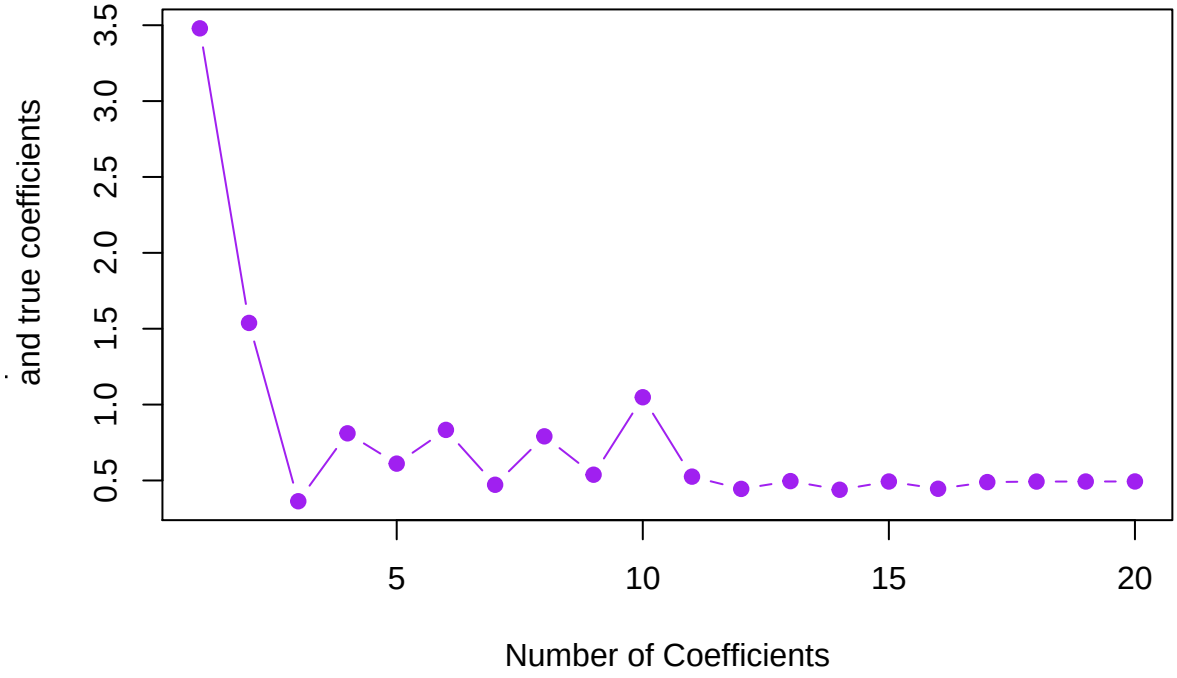
```
##         (Intercept)
## True.b  0.000000000 0.23534852 -0.64218689 -1.9348085 -0.283550120 -1.40972909
```

```
## Fitted  0.007064972 0.13800517 -0.67378904 -1.6871464 -0.279716500 -1.36865132
## Err    -0.007064972 0.09734335  0.03160215 -0.2476621 -0.003833621 -0.04107778
##
## True.b 0.72318044 2.03103552 0.7304903  0.87915338 -0.28458105 -0.674658
## Fitted 0.68628979 1.99834191 0.5664629  0.89338940 -0.21243303 -0.506367
## Err    0.03689065 0.03269362 0.1640274 -0.01423602 -0.07214803 -0.168291
##
## True.b -0.7154889 -0.27052793 0.3129646 0.8922593 -1.01548890
## Fitted -0.8375578 -0.35255542 0.2091002 0.7254166 -1.05248845
## Err     0.1220690  0.08202749 0.1038644 0.1668427  0.03699955
```

In the true model, a few parameteres were zeroed. The best subset model with MSE minimum built will be able to identify and remove the said parameters.

(g) Create a plot displaying p j=1(j − ˆ r j)2 for a range of values of r, whereˆ r j is the jth coeffcient estimate for the best model containing r coeffcients. Comment on what you observe. How does this compare to the test MSE plot from (d)?

```
val.errors <- rep(NA, 20)
x_cols = colnames(x, do.NULL = FALSE, prefix = "x.")
for (i in 1:20) {
    coefi <- coef(regfit.full, id = i)
    val.errors[i] <- sqrt(sum((b[x_cols
                                %in% names(coefi)] - coefi[names(coefi)
                                %in% x_cols])^2) +
                            sum(b[!(x_cols %in% names(coefi))])^2)
}
plot(val.errors, xlab = "Number of Coefficients", ylab = "Mean Squared Error for estimate
     and true coefficients",
     pch = 19, type = "b",col="purple")
```



Error is minimized for 3 variables. It can be noticed that for variables from 14-20, it decreases further. For a 14 variable model, test MSE is minimum, therefore, for whichever model produces the closest parameter estimates to the true parameter shall not give the least test MSE.