Q21

John Bute

2024-11-01

R Markdown

1. Generate a data set with p = 20 features, n = 1,000 observations, and an associated quantitative response vector generated according to the model Y = Xβ + ε, where β has some elements that are exactly equal to zero

```
set.seed(1)
n <- 1000
p <- 20

X <- matrix(rnorm(n*p), nrow = n, ncol = p)

B <- rnorm(p)

B[5] = 0
B[9] = 0
B[14] = 0
B[19] = 0

epsilon <- rnorm(n)

Y <- X %*% B + epsilon
```

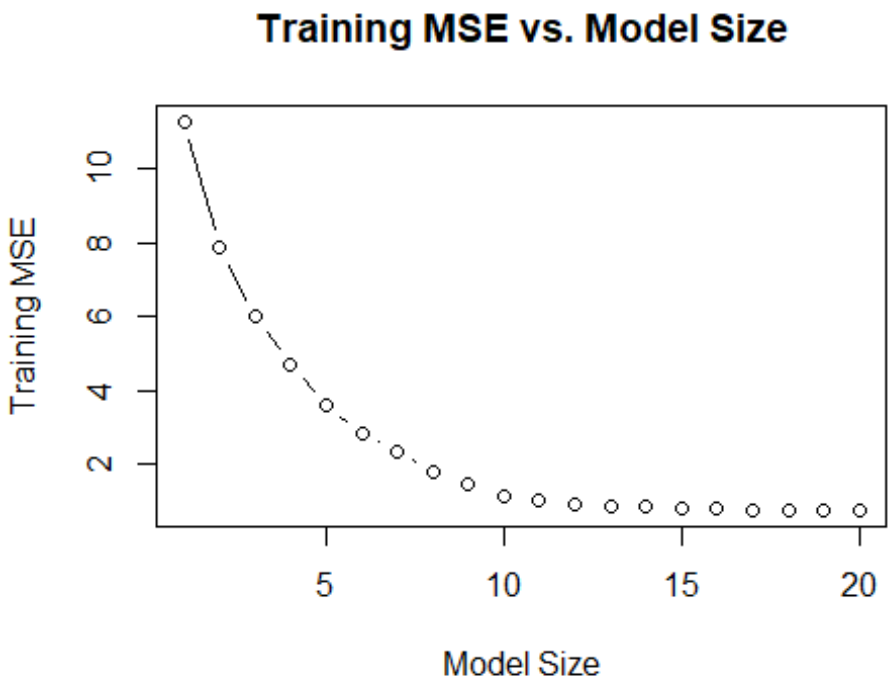
2. Split your dataset into a training set containing 100 observations and a test set containing 900 observations

```
colnames(X) <- paste0("X", 1:p)

train_indices <- sample(1:n, 100)

test_indices <- setdiff(1:n, train_indices)

X_train <- X[train_indices, ]
Y_train <- Y[train_indices]

X_test <- X[test_indices, ]
Y_test <- Y[test_indices]
```

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

```
library(leaps)
subset_selection <- regsubsets(X_train, Y_train, nvmax = p)
train_mse <- rep(NA, p)
```

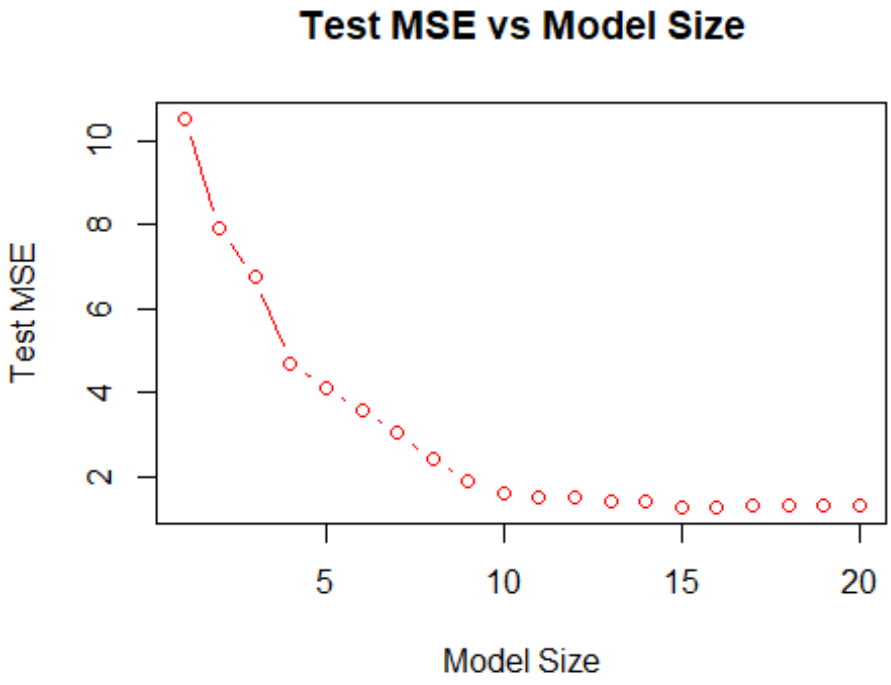
```r
errors = rep(NA, p)
x_cols = colnames(X_train, do.NULL = FALSE, prefix = "x.")

for (i in 1:p){
  coefficients <- coef(subset_selection, id = i)
  selected_vars <- names(coefficients)[-1]
  X_train_subset <- X_train[, selected_vars, drop = FALSE]

  pred <- X_train_subset %*% coefficients[selected_vars]
  train_mse[i] <- mean((Y_train - pred)^2)
}


plot(1:p, train_mse, ylab = "Training MSE", xlab = "Model Size", type = "b",
main = "Training MSE vs. Model Size")
```



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

```r
test_mse <- rep(NA, p)

for (i in 1:p) {
  coefficients <- coef(subset_selection, id = i)
  selected_vars <- names(coefficients)[-1]
  X_test_subset <- X_test[, selected_vars, drop = FALSE]
  pred <- X_test_subset %*% coefficients[selected_vars]
```

```
  test_mse[i] <- mean((Y_test - pred)^2)
}

plot(1:p, test_mse, ylab = "Test MSE", xlab = "Model Size", type = "b", col =
"red", main = "Test MSE vs Model Size")
```




```
min_test_mse_size <- which.min(test_mse)
cat("Model size with minimum test MSE:", min_test_mse_size, "\n")

## Model size with minimum test MSE: 15

cat("Minimum Test MSE:", test_mse[min_test_mse_size], "\n")

## Minimum Test MSE: 1.281235
```

We found the minimum test MSE at an intermediate model size (15), rather than a model with only an intercept or with all features, thus showing that this model strikes a good balance between variance nad bias, suggesting that the model does not overfit.

```
best_model_coefficients <- coef(subset_selection, id = min_test_mse_size)
cat("Best model coefficients (model size =", min_test_mse_size, "):\n")

## Best model coefficients (model size = 15 ):

print(best_model_coefficients)

## (Intercept)          X2          X3          X4          X6          X7
##  0.03519428  0.35871039 -0.64548835 -1.74550310 -0.30274108 -1.27061990
```

```
##          X8         X10         X11         X12         X13         X15
##  0.68116585  0.57392707  0.90392727  0.56869742 -0.22612139 -0.91633913
##         X16         X17         X18         X20
## -0.34672823  0.18484563  1.61223854 -1.08100610
```

```
cat("True coefficients (B):\n")
```

```
## True coefficients (B):
```

```
print(B)
```

```
##  [1]  0.2353485  0.2448250 -0.6421869 -1.9348085  0.0000000 -0.2835501
##  [7] -1.4097291  0.7231804  0.0000000  0.7304903  0.8791534  0.5545564
## [13] -0.2845811  0.0000000 -0.7154889 -0.2705279  0.3129646  1.6698068
## [19]  0.0000000 -1.0154889
```

Firstly, our best model aligns its non zero coefficients well with the true coefficients as we set X5, X9, X14, and X19 to zero, thus identifying the important predictors Furthermore, the magnitudes of our coefficients in our model are somewhat close to the true values, but not quite exact due to the noise in the data, and the bias-variance tradeoff (although they all retain a general direction)

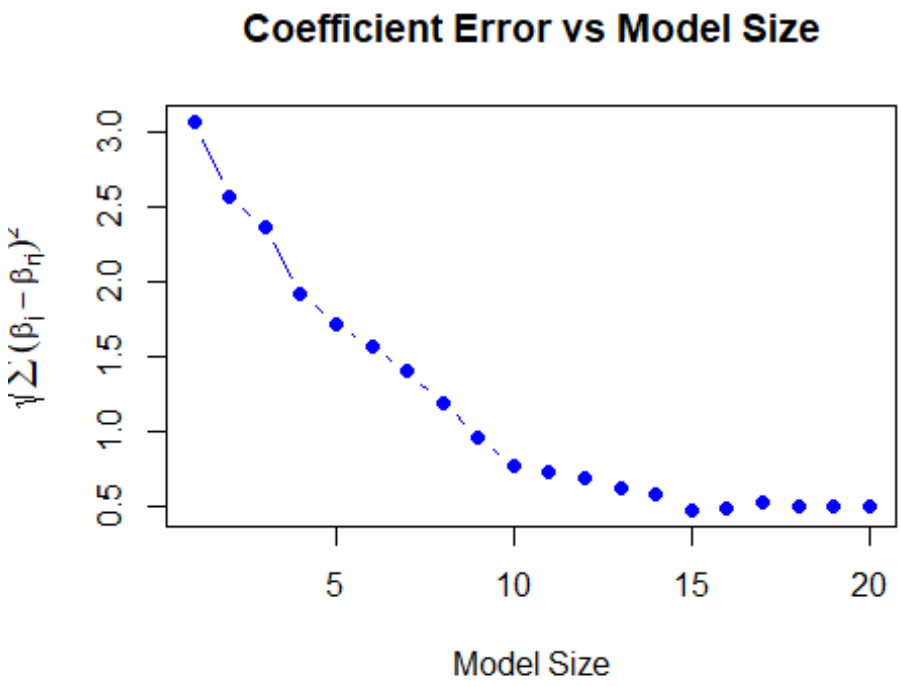
7. Create a plot displaying

$$\sqrt{\sum_{j=1}^{p}\left(\beta_j - \hat{\beta}_{rj}\right)^2}$$

for a range of values of r, where ˆβrj 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 4.?

```
coefficients_error <- rep(NA, p)

for (i in 1:p){
  coefficients <- coef(subset_selection, id = i)
  estimated_beta <- rep(0, p)

  selected_vars <- names(coefficients)[-1]
  selected_indices <- as.numeric(gsub("X", "", selected_vars))

  estimated_beta[selected_indices] <- coefficients[selected_vars]

  coefficients_error[i] <- sqrt(sum((B - estimated_beta)^2))
}


plot(1:p, coefficients_error, type = "b", col = "blue", pch = 19,
     xlab = "Model Size", ylab = expression(sqrt(sum((beta[j] - hat(beta)[r *
```

```
j])^2))),
     main = "Coefficient Error vs Model Size")
```



The coefficient error will decrease as we introduce more features, as larger models have more predictors that can capture true coefficients. However, after 15, we see we start to increase coefficient error as due to overfitting. We see something similar with the test MSE, as after the 15 features introduced, we see our test MSE suffer as a result.