Assignment12

John Bute

2024-10-14

Assignment 12

We collect a set of data (n = 100 observations) containing a single predictor X and a quantitative response Y . We then fit a linear regression model to the data, as well as a separate cubic regression, i.e. $Y = \beta0 + \beta1X + \beta2X2 + \beta3X3 + \varepsilon$.

1. Suppose that the true (statistical) relationship between X and Y is linear. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

We would expect the RSS for cubic regression to be equal or lower than the RSS for linear regression. Even though the true relationship between X and Y is linear, cubic regression, at its worse case, can reduce to a linear model by having its higher coefficients (B2, B3) closer to 0. Otherwise, the cubic regression model is more flexible and fit a wider range of patterns in the data, as it has more parameters and can bend more, potentially overfitting the training data.

2. Answer question 1., using test RSS rather than training RSS.

On the contrary however, I believe that the linear regression model will have a lower RSS than the cubic regression model, as the cubic regression model will overfit data of a linear nature, and not generalize well to new data. The higher-order terms (X2 and X3) will hurt the model's performance as it tries to fit patterns that would not exist in the test set, leading to a higher test RSS. However, a linear regression model will capture the true nature of the data, and will perform well with new data, especially if that data's X and Y is of linear nature. Finally, the linear regression model will have lower bias and lower virance, due to this linear nature, giving the model a smaller RSS.

3. Suppose that the true relationship between X and Y is not linear, but we don't know "how far" it is from being linear. Consider the training RSS for the linear regression, andfor the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

For training RSS, the cubic regression will do better than the linear regression, as its extra degrees of freedom will allow the cubic regression model to see patterns within the data and be more flexible, thus capturing patterns that may be non linear. Meanwhile, the linear model offers limitations, as its straight line relationship between X and Y causes the model

to lack flexibility to account for non-linear patterns, thus not fitting the data closely, giving it a high bias, low variance (due to its simplicty of having one term), therefore higher training RSS

4. Answer question 3., using test RSS rather than training RSS.

There is not enough information to determine which test RSS will be lower between the cubic regression and linear regresion mdoel, as we do not know the extent to which the data is non-linear. If the data is slightly non-linear, then the linear regression test RSS might be lower than the cubic's test RSS as the linear regression model will still generalize well. However, if the relationship between the data is strongly non-linear, then the cubic regression's test RSS will be lower, as the cubic regression's extra parameters will allow it to capture underlying patterns far better than the linear regression model. This ambiguity is caused by not knowing the true nature of the data, and thus not knowing what is the best level of flexibility necessary to fit the model.

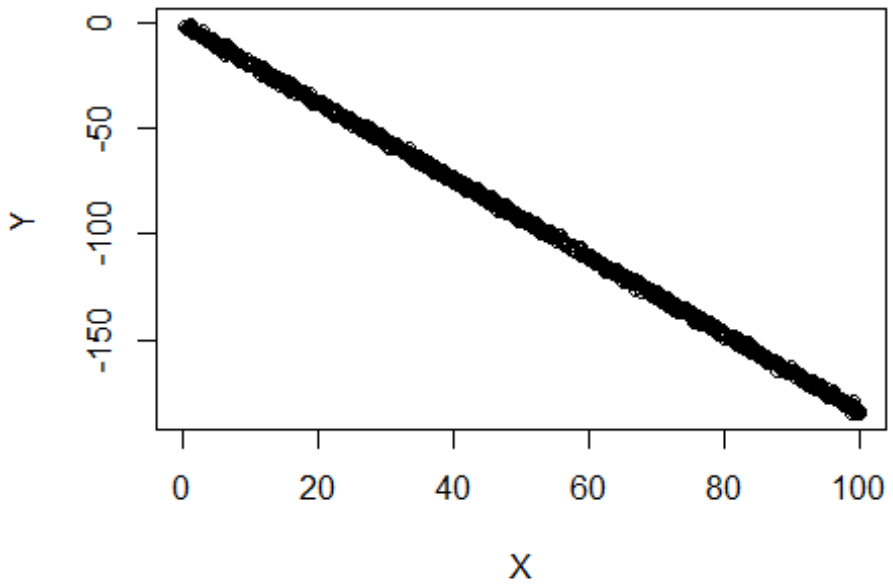
5. Generate some data to illustrate the situation of questions 1. and 2

```r
set.seed(0)
N <- 1000
X <- runif(N, 0, 100)
epsilon <-rnorm(N, 0, 1)

beta_0 = 5 - 2*5*runif(1)
beta_1 = beta_0 = 3 - 2*3*runif(1)

Y <- beta_0 + beta_1 * X + epsilon

train_idx <- sample(1:N, 70)

X_train <- X[train_idx]
Y_train <- Y[train_idx]
X_test <- X[-train_idx]
Y_test <- Y[-train_idx]

linear_model <- lm(Y_train ~ X_train)
Y_pred_linear_train <- predict(linear_model, data.frame(X_train = X_train))
Y_pred_linear_test <- predict(linear_model, data.frame(X_train = X_test))

cubic_model <- lm(Y_train ~ poly(X_train, 3, raw = TRUE))

Y_pred_cubic_train <- predict(cubic_model, data.frame(X_train = X_train))
Y_pred_cubic_test <- predict(cubic_model, data.frame(X_train = X_test))

RSS_linear_train <- sum((Y_train - Y_pred_linear_train)^2)
RSS_cubic_train <- sum((Y_train - Y_pred_cubic_train)^2)

RSS_linear_test <- sum((Y_test - Y_pred_linear_test)^2)
RSS_cubic_test <- sum((Y_test - Y_pred_cubic_test)^2)
```

```r
results <- data.frame(
  Model = c("Linear", "Cubic"),
  Training_RSS = c(RSS_linear_train, RSS_cubic_train),
  Test_RSS = c(RSS_linear_test, RSS_cubic_test)
)

print(results)
```

```
##    Model Training_RSS Test_RSS
## 1 Linear     74.70173 939.2703
## 2  Cubic     73.14707 962.3796
```

```r
plot(data.frame(X, Y))
```



As shown, the training RSS for cubic slightly trumps the RSS for linear. However, in the test RSS, the linear model generalizes the data extremely well.