Homework 8

Andrew Boschee

3/30/2020

Question 6.8.4: Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2$$

for a particular value of

$$\lambda$$

. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

i. Increase initially, and then eventually start decreasing in an inverted U shape. ii. Decrease initially, and then eventually start increasing in a U shape. iii. Steadily increase. iV. Steadily decrease. v. Remain constant.

Part A: As we increase lambda from 0, the training RSS will:

Model will be less flexibile and will result in steady increase of training error. (iii)

Part B: As we increase lambda from 0, the test RSS will:

RSS will initially decrease and increase in U shape as the model becomes less flexible. (ii)

Part C: As we increase lambda from 0, the variance will:

Steady decrease in variance from additional constraints. (iv)

Part D: As we increase lambda from 0, the (sqaured) bias will:

Decrease in flexibility for model will steadily increase bias. (iii).

Part E: As we increase **lambda** from 0, the ireeducible error will:

Irreducible error is constant regardless of lambda. (v)

College Applications Predictions

Question 6.8.9: In this exercise, we will predict the number of applications received using the other variables in the College data set.

Part A: Split the data into a training and test set.

Using a 75/25 ratio, the training and test sets are split with a summary in the table below.

Table 1: Data Split Summary (75/25)

Total	Train	Test
777	582	195

MLR

Part B: Fit a linear model using least squares on the training set, and report the test error obtained.

Starting with a linear model, predictions are made on the test set giving an initial error rate to compare with other models.

Table 2: MSE - Linear Model - College

1559649

Ridge Regression

Part C: Fit a ridge regression model on the training set, with lambda chosen by cross-validation. Report the test error obtained.

After building the model with cv.glmnet() function, the optimal lambda value was chosen and used to make predictions on the test dataset. Using ridge regression with ten folds we see a a higher MSE value in comparison to MLR from part A.

Table 3: MSE - Ridge Regression - College Data

1797867

Lasso

Part D: Fit a lasso model on the training set, with lambda chosen by cross-validation. Report the test error obtained, as well as the number of non-zero coefficient estimates.

While Lasso performed better than ridge regression, it still had a slightly higher MSE in comparison to MLR in part B. From the table, we can also see the value of 16 for the count of non-zero coefficient estimates. I didn't expect to see quite so many to be used in the model. That seems like quite a few factors to take into account.

Table 4: MSE - Lasso - College Data

1625391

Table 5: Count of Non-Zero Coefficient Estimates

16

Principal Component Regression

Part E: Fit a PCR model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation.

PCR gave identical result to MLR and better than both ridge and lasso models.

Table 6: MSE - PCR - College Data

1559649

Partial Least Squares

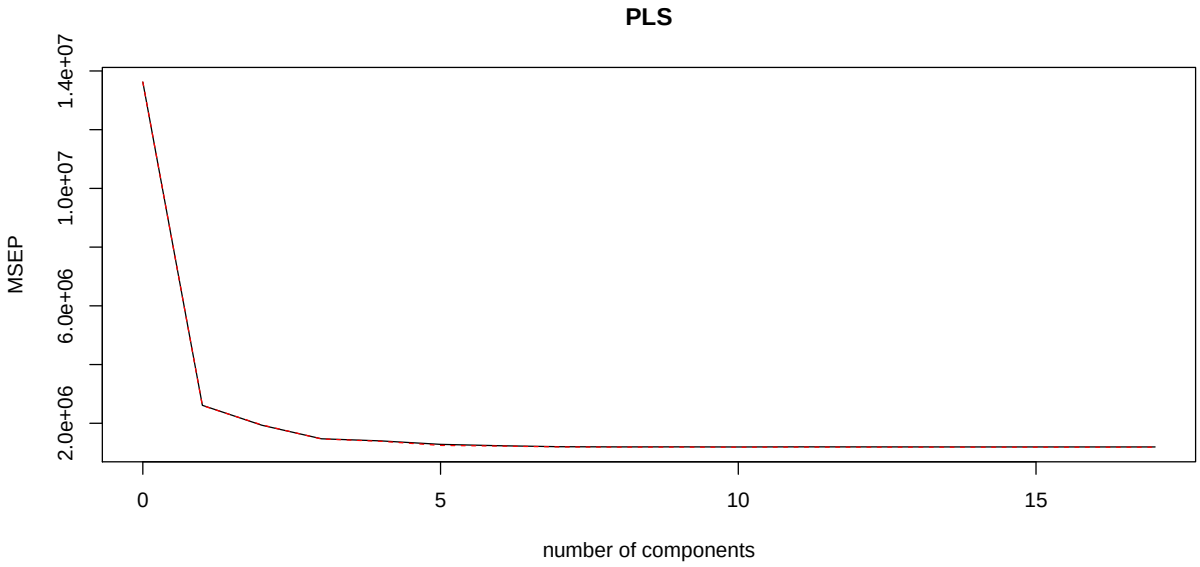
Part F: Fit a PLS model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation.

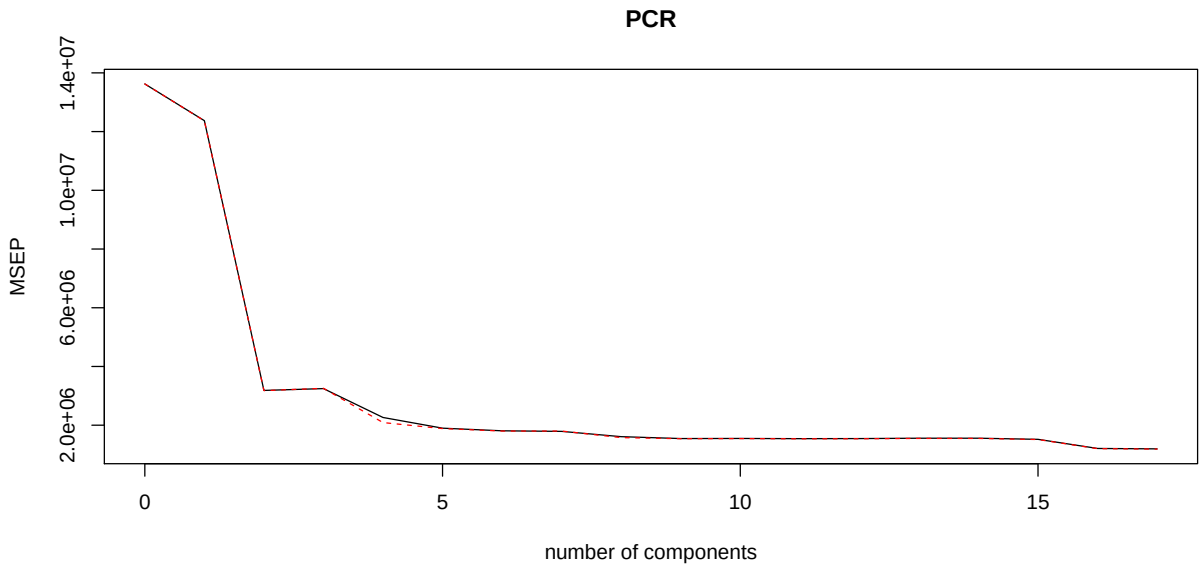
PLS gives identical results as PCR and on the lower end of the error rates when comparing models

Table 7: MSE - PLS - College Data

1559649

PLS vs PCR





Results

Part G: Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these 5 approaches?

From the results, Ridge and Lasso models did not perform as well as other methods and none of them were very impressive overall. Ridge definitely performed the worst while there was not much separtation regarding MSE overall with the other methods

With PCR and PLS providing identical results, plots above look at the comparison of error regarding the number of components. While they both flatten out around four or five components, PLS has a quicker drop in MSE, while PCR also seems to go up a tad bit between two and three.

Boston Crime Rate

Question 6.8.11: We will now try to predict per capita crime rate in the **Boston** data set.

Part A: Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss resultls for the approaches that you consider.

Similar to prior exercise, we will be using 75/25 split on the training and test set. Each of the listed methods are used in this step with output comparison at the end. It will be interesting to see the variables that are ssen as important when comparing forward and backward stepwise methods.

Table 8: Data Split Summary (75/25)

Total	Train	Test
506	379	127

Model Comparison

Part B: Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross-validation, or some other reasonable alternative (not training error).

Models Used: Backward Stepwise Selection, Forward Stepwise Selection, Principal Component Regression, Lasso, and Ridge Regression.

All of these models where farly close in the end with backward stepwise regression taking a slight edge over the lasso model.

Table 9: Boston Model MSE Comparison

Backward	Forward	PCR	Lasso	Ridge
38.08648	38.08648	38.0854	38.49346	38.34539

Best Model - Backward Stepwise Selection (Number Features = 4)

Part C: Does your chosen model involve all of the features in the data set? Why or why not?

Backward selection gave optimal results with only 4 of the features used in the model. When going through the loop, the mean squared error value began to go back up once more features were removed. The model saw that when there were more than four features being used they were not useful and dropping them resulted in less noise with better predictions.

	coef(bwdStep, id = 4)
(Intercept)	7.3546484
dis	-0.2636015
rad	0.4762822
black	-0.0129239
medv	-0.1250073