QF-P1 Linear Feature Engineering

Xi Tan, MD Armanuzzaman

September 2019

1 Prediction Result

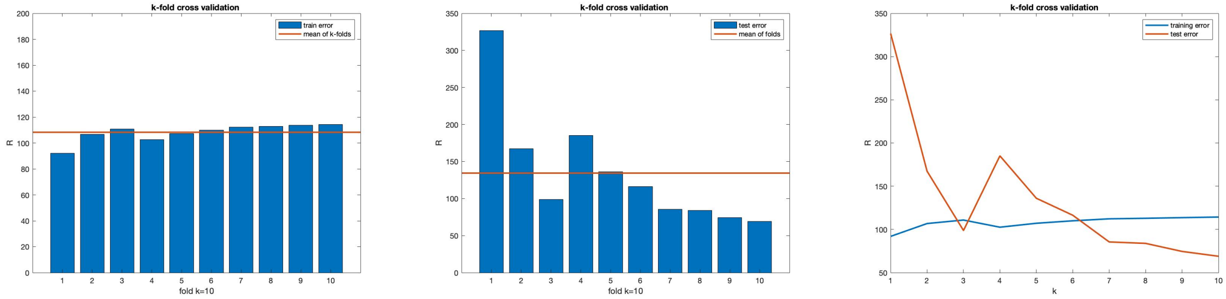
The result of our project is (choose p=4, then y = a*x^0 + b*x^1 + c*x^2 + d*x^3 + e*x^4):

Train Error	Test Error
38.3947	bigger than 38.3947

2 Dealing with overfitting

2.1 K-fold Cross Validation

We tried k-fold cross validation first, set the k as 10, and got the two error-sets:



(a) k-fold-cross-validation-train-error

(b) k-fold-cross-validation-test-error

(c) k-fold-cross-validation

Figure 1: k-fold cross validation

From those figures, we can know if we only use k-fold to divide our data into k parts, the test error is very different from the train error. From Figure1-c we know, the prediction is very worse. So we need more method to make our prediction more valid.

2.2 Polynomial selection using K-fold cross validation

With the dataset we all have, for p = 0,1,...,10 estimate the test error using 10-fold cross validation. Verify that we get estimated test error as in the following table:

Order	Train Error	Test Error
0	281.8702	302.7441
1	108.0804	136.3692
2	62.4948	82.8214
3	43.7644	64.6859
4	37.6895	56.8482
5	61.1295	85.8305
6	122.7573	200.6629
7	136.8197	215.4716
8	145.0155	229.5739
9	151.4901	282.6136
10	158.9121	263.9399

Table 1: errors

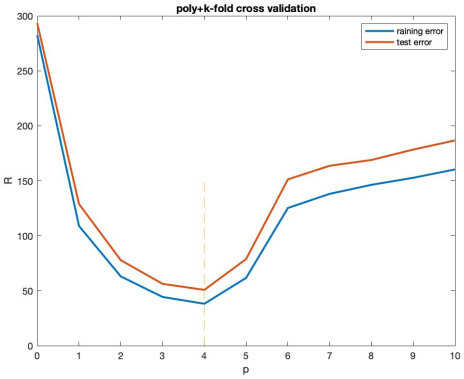
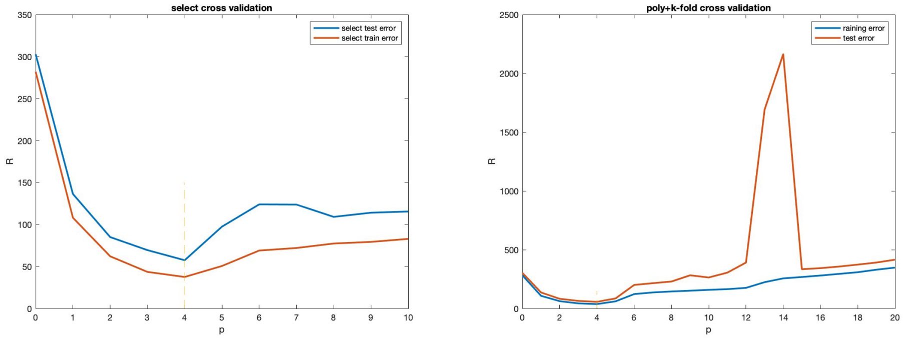


Figure 2: cross validation error

From figure2, the smallest test error is 56.8482 when p is 4 which means we expand the x to fourth degree polynomial.

3 Feature Selection

We don't know which column of x make the biggest effort to our prediction, so we choose the average of the 8 column as x9, and add the new x9 into the original x dataset. Expand our x data sets into 9 columns. The do polynomial selection using K-fold cross validation, just the same as part two. Then we got the result as figure3-a.



(a) select-poly-with-k-fold-cv

(b) poly+k-fold-cv-p20

Figure 3: linear feature engineering

Add average of x into the old x makes no difference to get the best p, but the test error became smaller.

4 Conclusion

From the very beginning, we don't know how large the p should be tested and how to set the k. So we tried to set k from 1 to 100 and found out that, no matter how large the k set, it always return the best p is 4. Then we tried p from 0 to 20, this is something interesting, like the figure3-b shows, the best p is 4. And when p came to 14, the test error is very large. Finally, we just set p from 0 to 10.