Linear Feature Engineering

By MD Armanuzzaman, Xi Tan

Process Overview

1. Model selection
    - polynomial selection
    - Basis expansion
2. Dealing with overfitting
    - K-fold cross validation
    - Polynomial selection with k-fold cross validation
    - With the selected polynomial generate least square error on the whole data.
3. Feature selection

Polynomial selection

- Select a model to train our data
- For polynomial (1 to p) run k fold chunks and get the testing mean error.
- Keep track of test errors for every p.
- Compare the mean test error for every P and select the p with smallest test error.

Model selection

- Divided the training data into K chunks to select the desired P.
  - Run p from 1 to p and observed that least testing Error.

Therefore, the selected model for our data:

$$Y = lX^{p}+hX^{p-1}+...+cX^{2}+bX^{1}+ax^{0}$$

Basis expansion

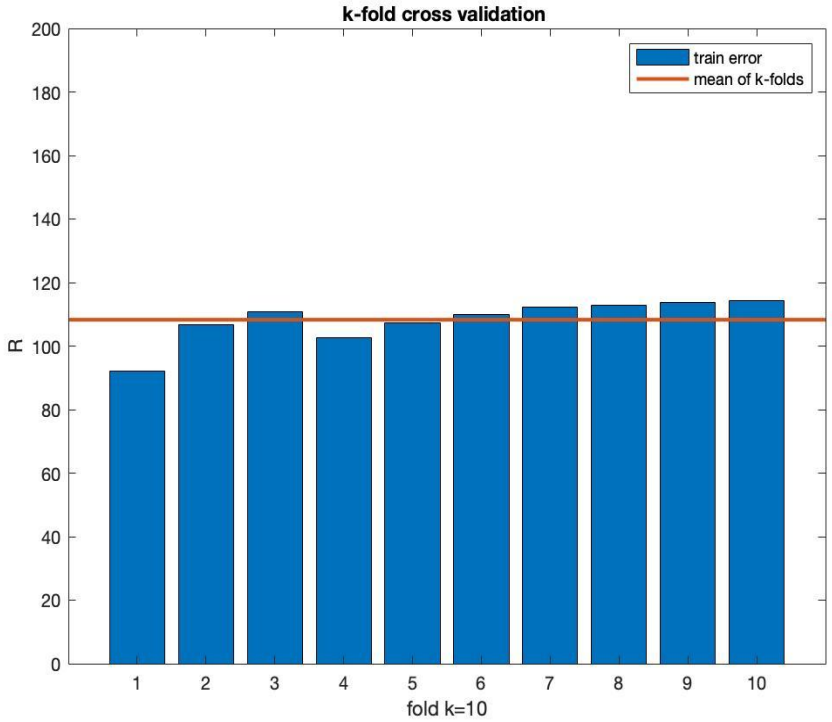
- After selecting the model we implemented the basis expansion to fit this model.

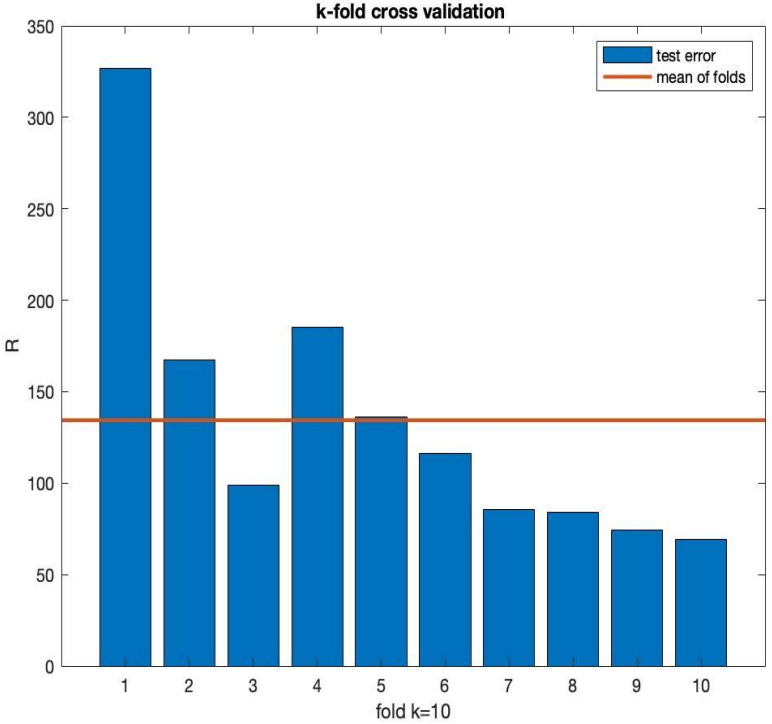
After basis expansion

Z = [1, x^1, x^2, x^3,..., x^p]

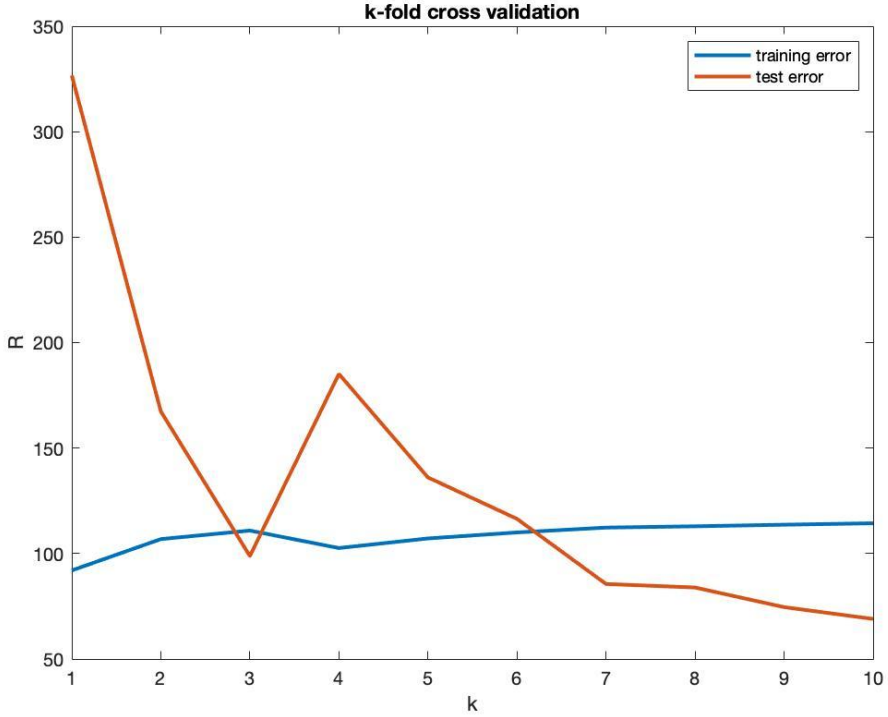
1	x^1	x^2	...	x^p
1	x^1	x^2	...	x^p
...	...	...	...	...
...	...	...	...	...
1	x^1	x^2	...	x^p

K-fold cross validation



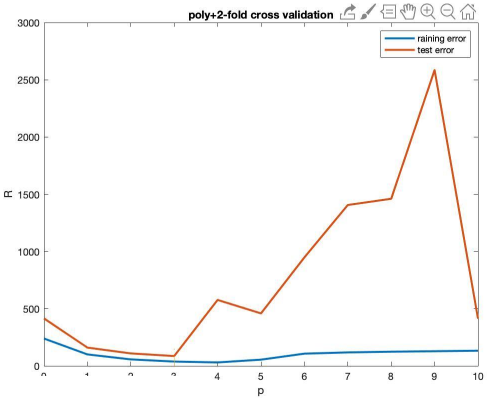


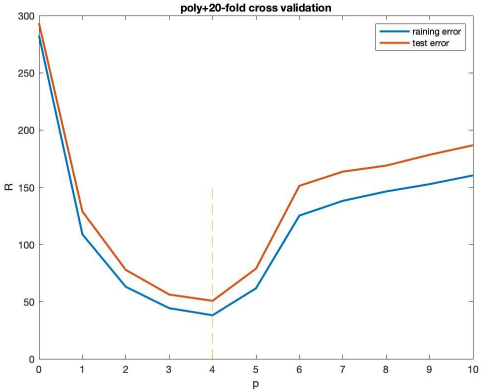
K-fold cross validation

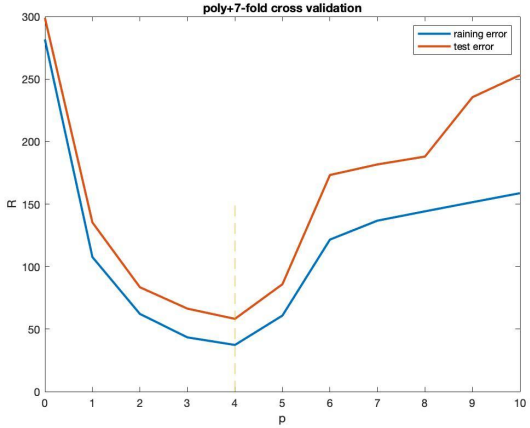


If we only use k-fold cross validation to train our data, the test errors are not so good.

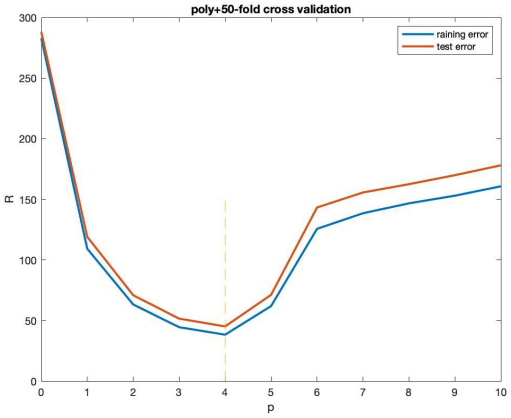
Polynomial selection using K-fold cross validation



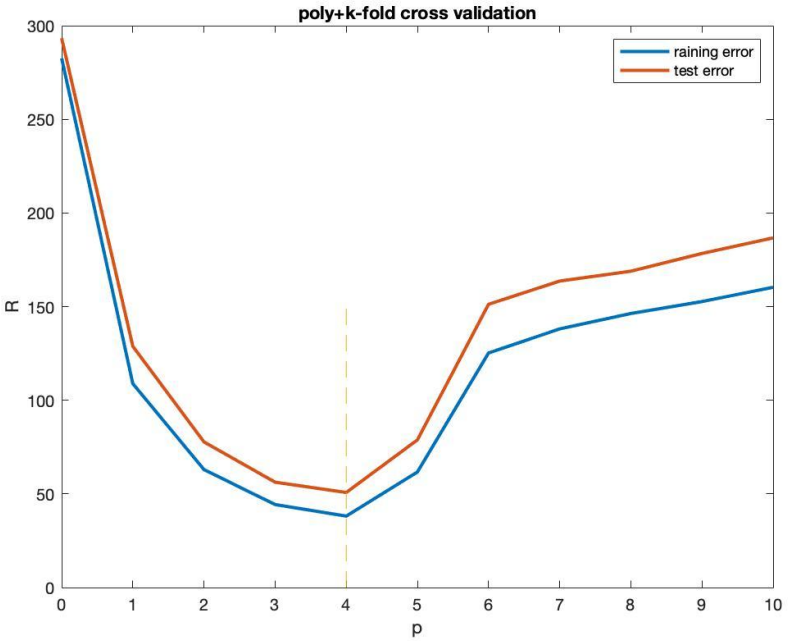


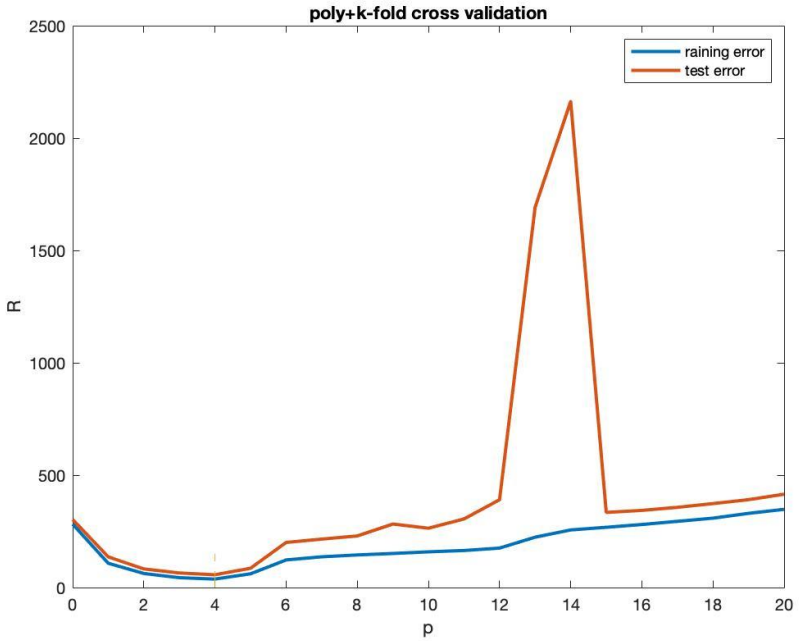


k?



Polynomial selection using K-fold cross validation





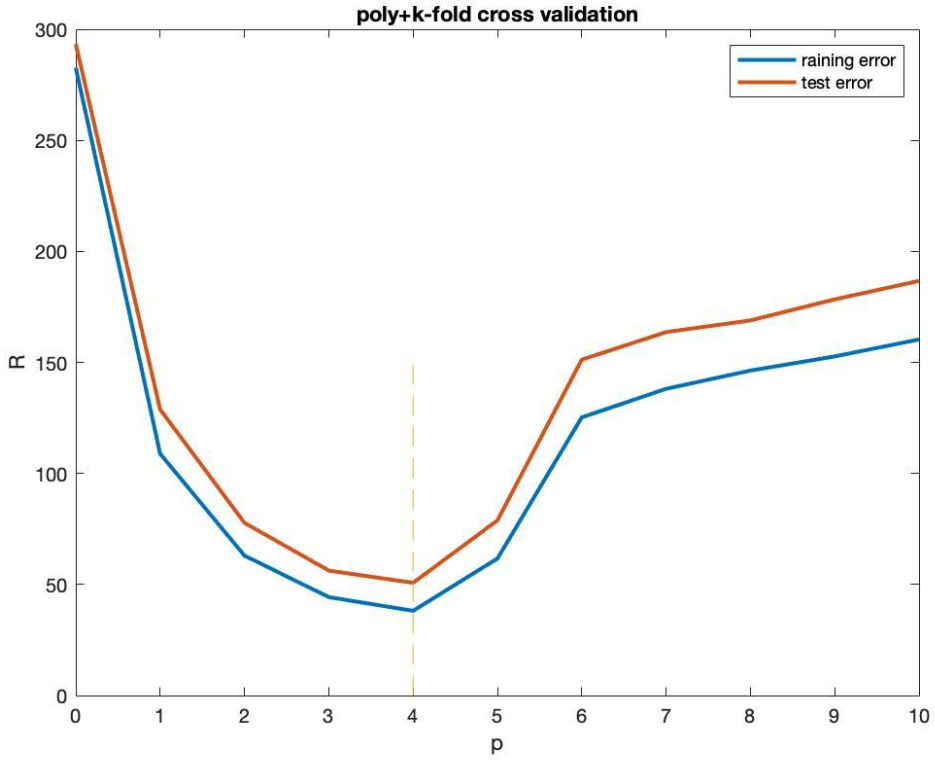
p?

Polynomial selection using K-fold cross validation

Set k = 10

P = 0:10

We get the best p = 4

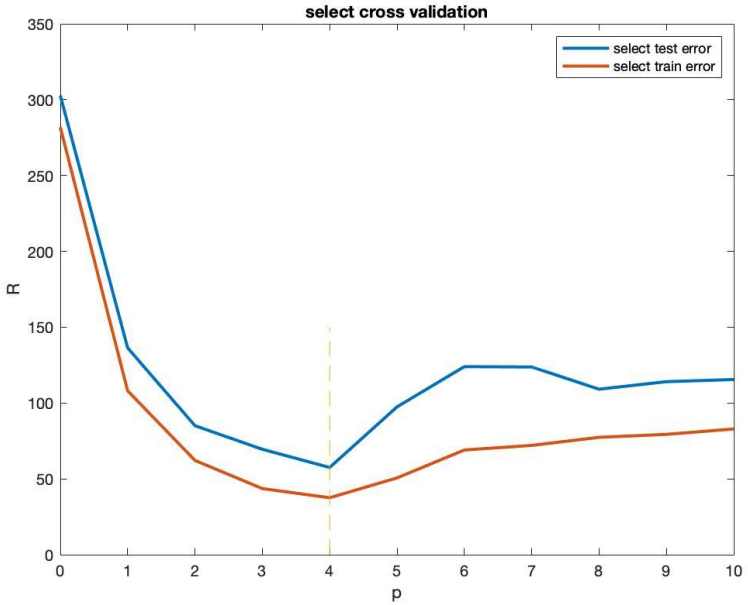


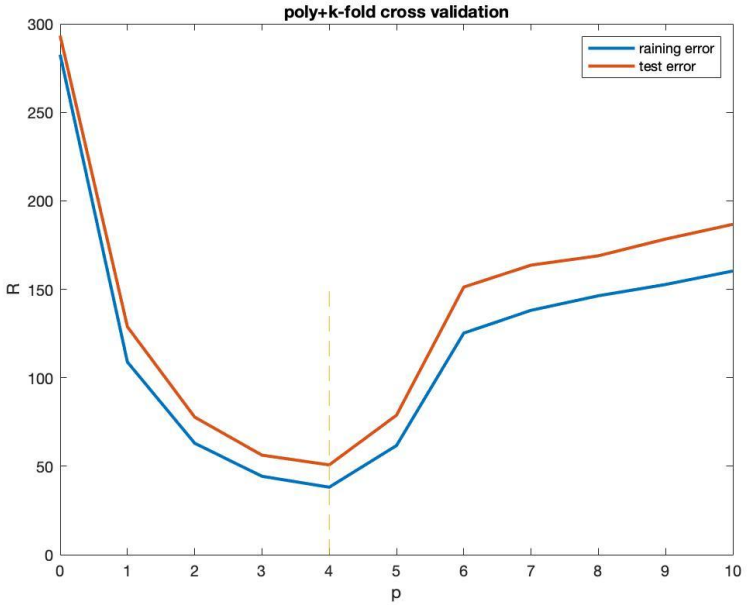
Feature selection

Try to define which feature of xi(0~8) is the most effective one to do the prediction for y.

Set new x9 as the average the the x, add the new x9 in x features. Train those new features.

Feature selection





Result

Prediction for the test error	Training error (R-train)	Poly+k-fold test error (R-smallest)
>38.3957	38.3947	56.8482

Conclusion

Train model(p = 4):

Y = a*x^4 + b*x^3 + c*x^2 + d*x^1 + 1

Expansion:

Z = [1, x^1,...,x^(p-1), x^p]

K-fold cross validation:

K = 10

Constants are very important, we tried to drop all constants, and when we run our codes, different k returns different best p.

Thank You