MATH 304 - Numerical Analysis and Optimization

Project ---Least Squares Regression

Xi Chen

xc166

Video Link

Outline

- Introduction
- Methodology
- Result
- Discussion

Introduction

In this project, we will try to use least square regression to find out the curve that fit the data.

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

Methodology

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x^1 + a_0$$

We need to find out the best set of a_n to fit the curve. The process is basically solving the overdetermined function:

$$Ax = B$$

Methodology

$$\left[\begin{array}{ccc} & A & \\ \hline \sqrt{\lambda} & & \\ & \sqrt{\lambda} & \\ & & \sqrt{\lambda} \end{array}\right] \cdot \alpha = \left[\begin{array}{c} B \\ \hline 0 \\ 0 \\ 0 \end{array}\right]$$

Figure 1. Ax=B with regularization

Methodology

Regularization is used to minimize the overfitting influence which is caused by too inappropriate coefficient for the least square regression. Sometimes, the model is trained too well to fit some train data. In this case, this model will lose some capacity to be generalized to deal with other data like the test data. A common feature of overfitting is that there are many obvious distortions and the coefficients are very large in the fitting curve. To solve this problem, we introduce the concept of regularization which means adding a penalty for this kind of situation to make the coefficient not outstanding in the training. The general format for the regularization is:

$$\|A \cdot \alpha - B\|_2^2 + \lambda \cdot \|\alpha\|_2^2$$

Methodology

We set up one array to store the x value for the data which is called a. Based on this array, we create a new matrix b:

$$b = [a^1 \quad \cdots \quad a^{degree}]$$

Then we will splice a column matrix which stand for the coefficient for a^0 that is all 1 with b adduction from the left.

$$b = [ones \quad b]$$

Methodology

Then we create a new matrix $c\ degree \times degree$ whose the main diagonal is $\sqrt{\lambda}$:

$$c = \begin{bmatrix} \sqrt{\lambda} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda} \end{bmatrix}$$

Matrix A is the combination of Matrix c and Matrix b from vertical direction:

$$A = \begin{bmatrix} b \\ c \end{bmatrix}$$

Methodology

$$A = \begin{bmatrix} a^0 & a^1 & a^2 & \cdots & a^{degree} \\ \sqrt{\lambda} & 0 & \cdots & & 0 \\ 0 & \sqrt{\lambda} & \cdots & & 0 \\ 0 & 0 & \sqrt{\lambda} & \cdots & \\ & \cdots & & & \end{bmatrix}$$

Picture 1. Composition of A matrix

Methodology

Meanwhile, we need to deal with the y values of the data points as well. We set up one array to store the y value for the data which is called y. To carry out the matrix calculation, y should be an $\# \; of \; data \, points \; \times degree$ matrix. So we need to provide enough zeros for it:

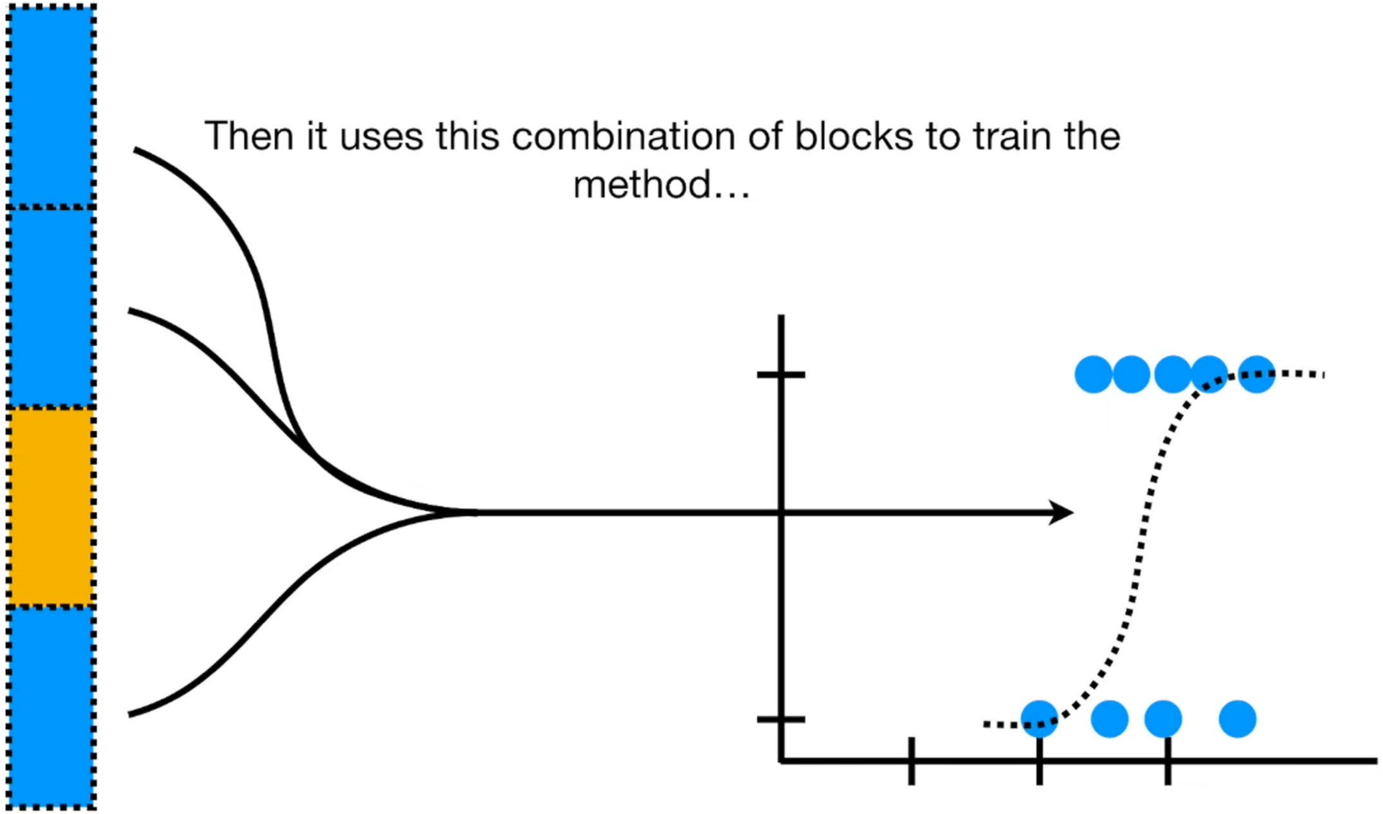
$$B = \begin{matrix} y \\ \vdots \\ 0 \end{matrix}$$

So by calculation:

$$X = A \backslash B$$

We can find out the solution for this least square problem.

Methodology



Picture 2. Example of 4-fold cross validation

Result-Task1

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Training Error	0.855951	0.853836	0.27023	0.243957	0.168454	0.153101	0.110869	0.074539	1.18E-20
Test Error	0.156486	0.146281	0.148816	0.161313	0.25901	0.302324	0.398575	0.413903	4.979396

Table 1. Small Data Error without regularization

Result-Task1

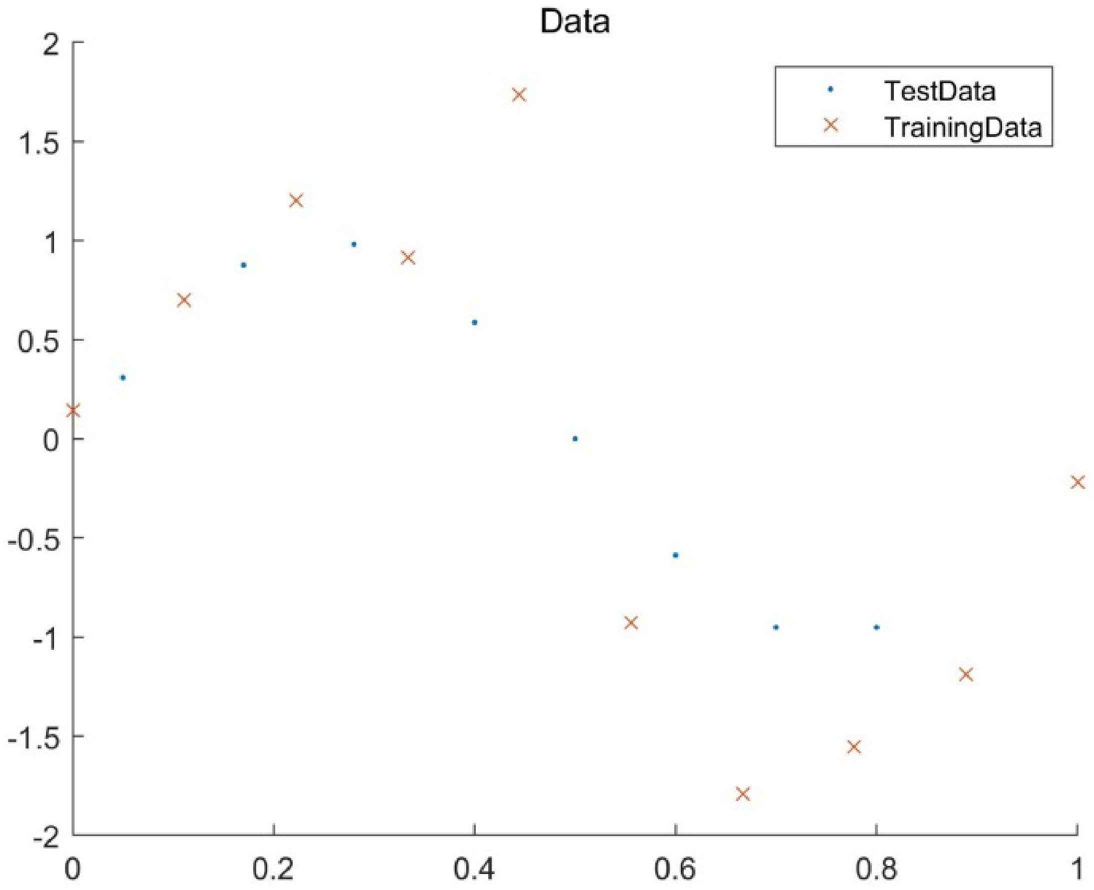


Figure 2. Training Data and Test Data for Task1

Result-Task1

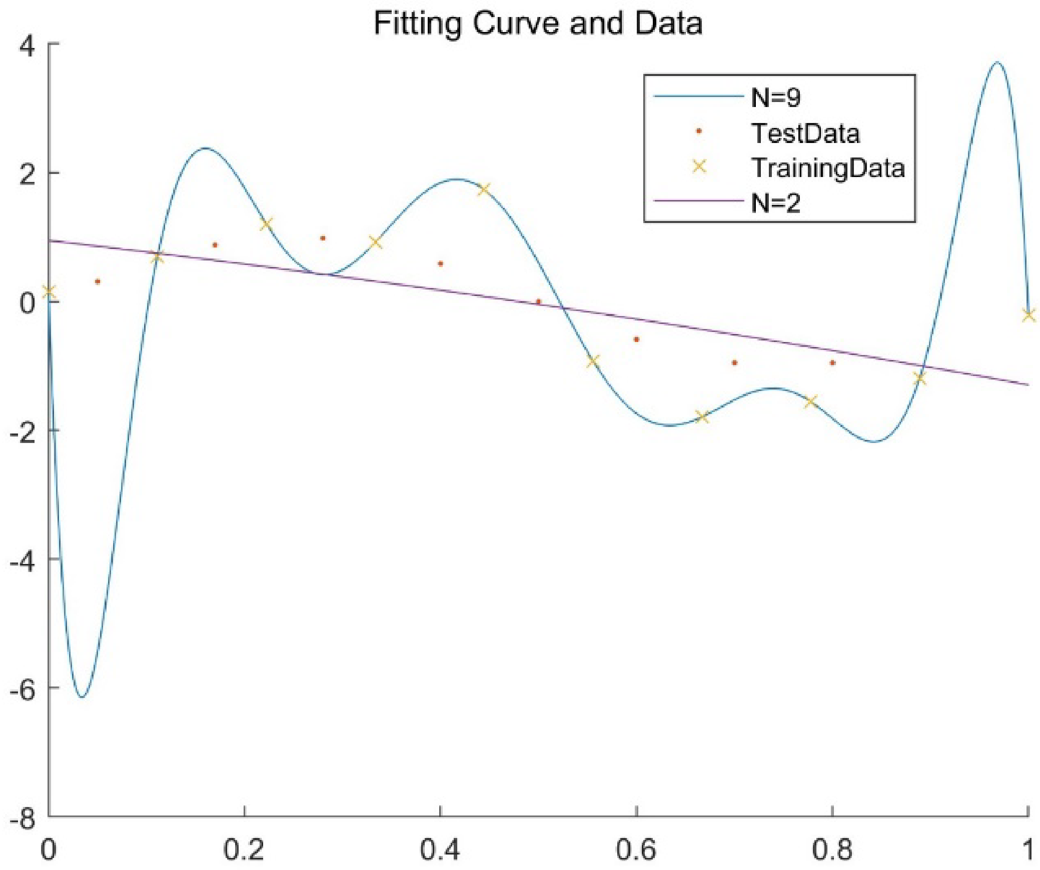


Figure 3. Data and Fitting Curve with smallest error

Result-Task1

From a., we can see that when $n = 2$, we have the smallest Test Error:

$$a_0 = 0.9464$$
$$a_1 = -1.7276$$
$$a_2 = -0.5127$$

When $n = 9$, we have the smallest Training Error:

$$a_0 = 0.0000$$
$$a_1 = -0.0004$$
$$a_2 = 0.0104$$
$$a_3 = -0.0935$$
$$a_4 = 0.4348$$
$$a_5 = -1.1689$$
$$a_6 = 1.8835$$
$$a_7 = -1.7944$$
$$a_8 = 0.9318$$
$$a_9 = -0.2032$$

Result-Task2

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Training Error	0.386 532	0.386 018	0.215 725	0.215 499	0.209 562	0.206 802	0.205 401	0.202 191	0.200 51
Test Error	0.181 946	0.178 101	0.004 612	0.004 618	0.000 477	0.002 946	0.005 07	0.009 12	0.007 493

Table 2. Large Data Error without regularization

Result-Task2

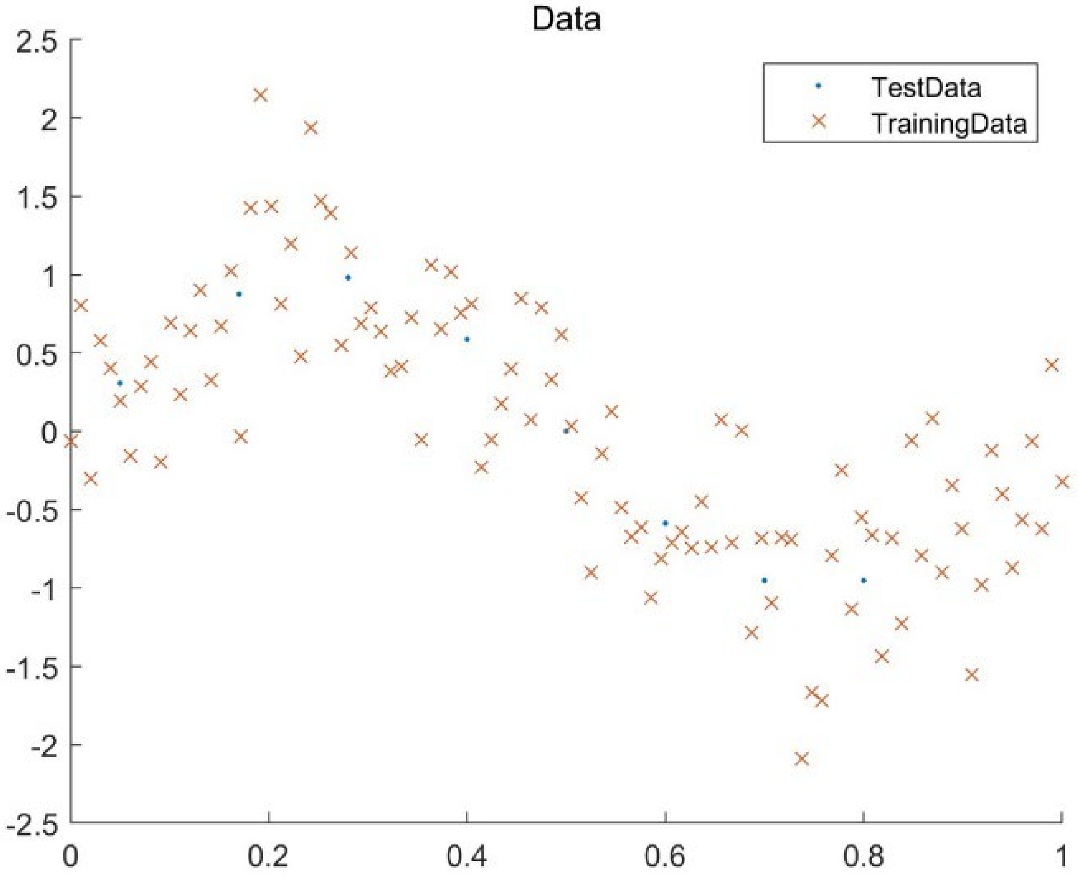


Figure 4. Training Data and Test Data for Task2

Result-Task2

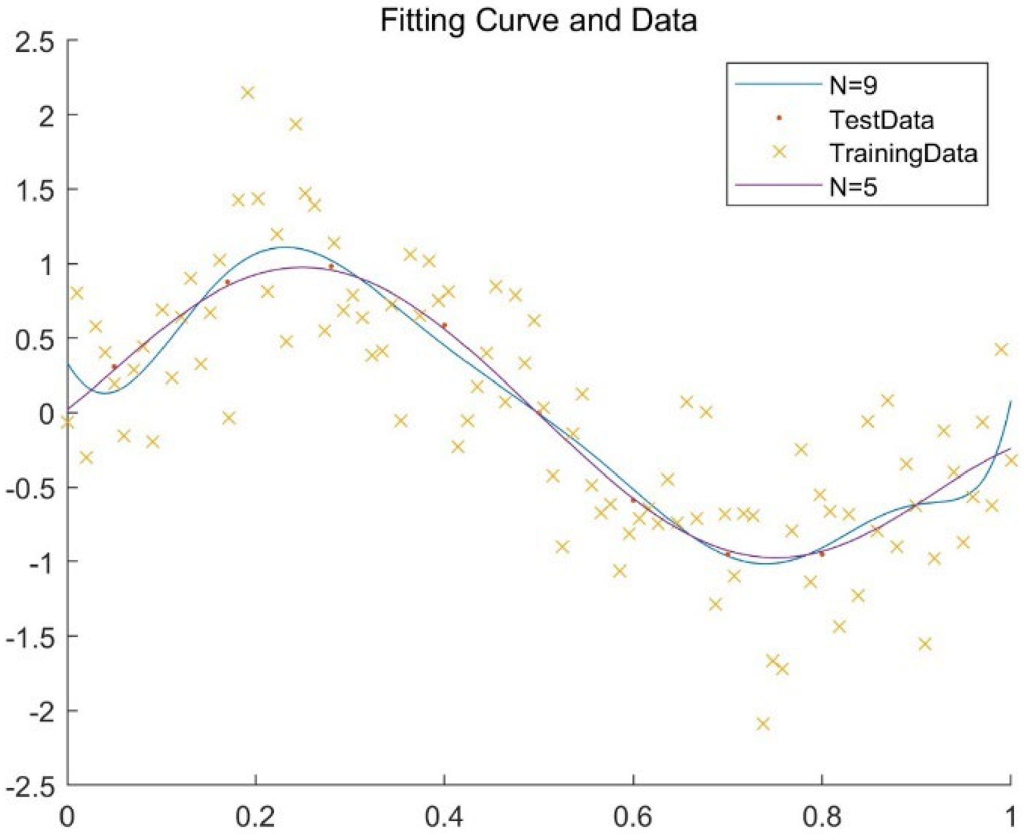


Figure 5. Large Data and Fitting Curve with smallest error

Result-Task2

From a., we can see that when $n = 5$, we have the smallest Test Error:

$$a_0 = 0.0212$$
$$a_1 = 4.8355$$
$$a_2 = 14.6874$$
$$a_3 = -108.8702$$
$$a_4 = 150.4993$$
$$a_5 = -61.4130$$

When $n = 9$, we have the smallest Train Error:

$$a_0 = 0.0000$$
$$a_1 = -0.0011$$
$$a_2 = 0.0142$$
$$a_3 = 0.0022$$
$$a_4 = -0.4917$$
$$a_5 = 2.2966$$
$$a_6 = -4.9279$$
$$a_7 = 5.6457$$
$$a_8 = -3.3434$$
$$a_9 = 0.8053$$

Result-Task3

Model	10^-6	10^-3	1	10^3	10^6
Training Error	0.156036	0.229233	0.839931	1.370526	1.376684
Test Error	0.280026	0.139054	0.192366	0.538351	0.541034

Table 3. Regularization Error

Result-Task3

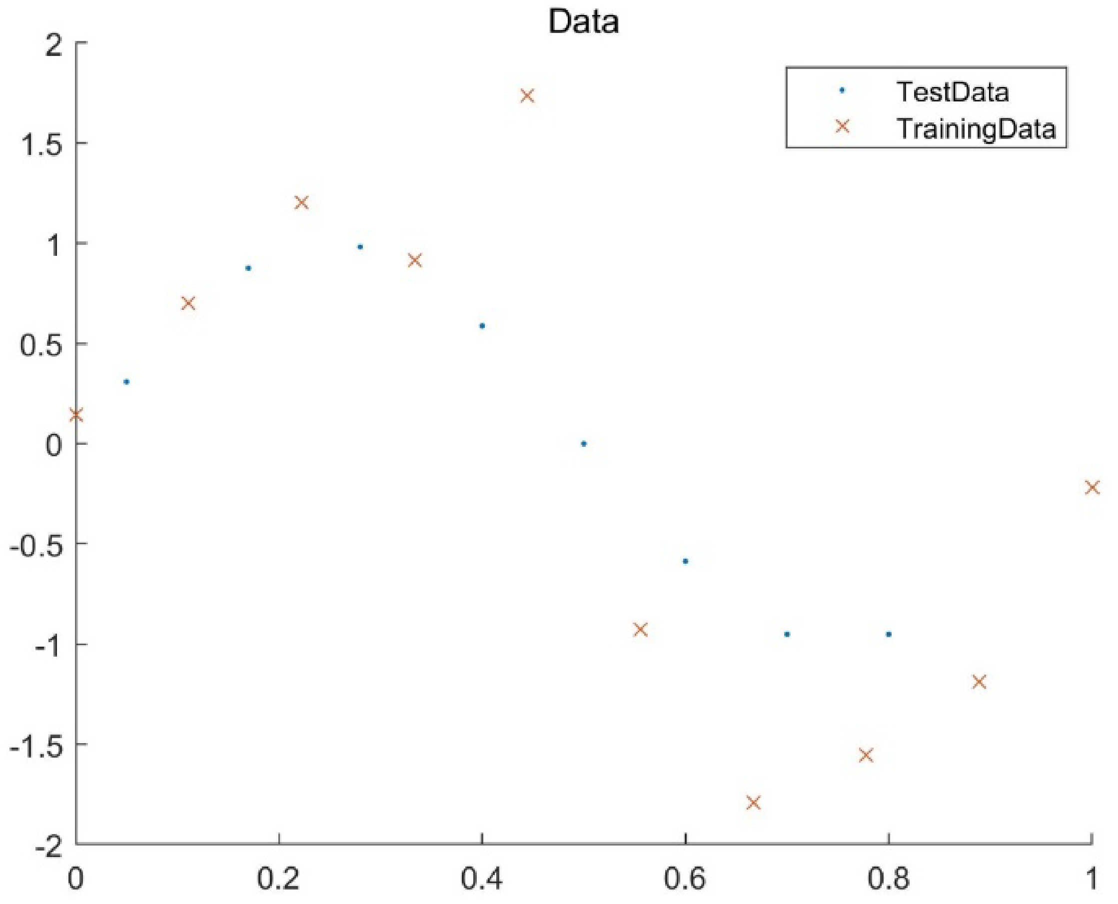


Figure 6. Training Data and Test Data for Task3

Result-Task3

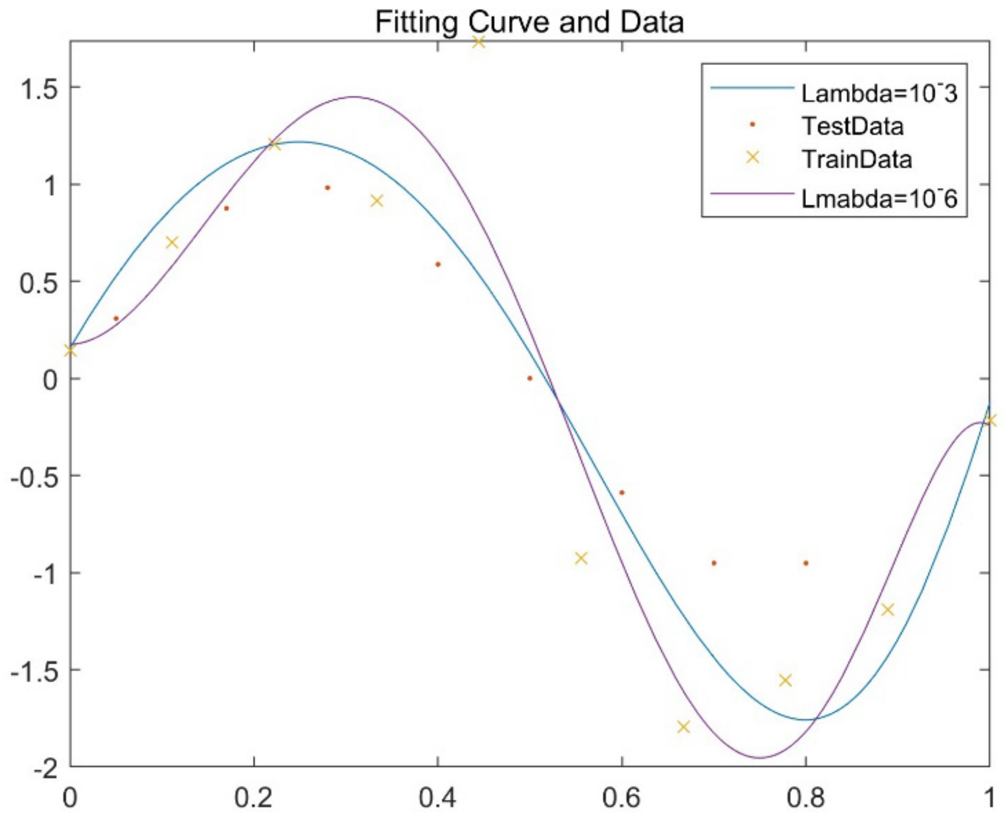


Figure 7. Fitting Curve and Data

Result-Task3

when $\lambda = 10^{-3}$, the coefficients are:

$$a_0 = 0.1566$$
$$a_1 = 8.0897$$
$$a_2 = -13.1951$$
$$a_3 = -9.4143$$
$$a_4 = 0.4092$$
$$a_5 = 6.9911$$
$$a_6 = 8.4356$$
$$a_7 = 5.6119$$
$$a_8 = -0.0217$$
$$a_9 = -7.1864$$

when $\lambda = 10^{-6}$, the coefficients are:

$$a_0 = 0.1754$$
$$a_1 = 0.0994$$
$$a_2 = 42.5976$$
$$a_3 = -93.6615$$
$$a_4 = -41.5986$$
$$a_5 = 92.0527$$
$$a_6 = 84.5541$$
$$a_7 = -29.4705$$
$$a_8 = -92.7029$$
$$a_9 = 37.7115$$

Result-Task4

Weight	10^-6	10^-3	10^-0	10^3	10^6
average validation error	0.061266	0.021855	0.064627	0.179386	0.136178

Table 4. Cross Validation

Result-Task4

$$Test\ Error = 0.27119115012813369557673835053065$$

Coefficient:

$$a_0 = -0.0185$$
$$a_1 = 7.8914$$
$$a_2 = -15.8554$$
$$a_3 = -4.7887$$
$$a_4 = 5.6510$$
$$a_5 = 7.6883$$
$$a_6 = 4.8984$$
$$a_7 = 0.8601$$
$$a_8 = -2.4331$$
$$a_9 = -4.1180$$

Best regularization weight:

$$\lambda = 10^{-3}$$

Discussion

Training errors decrease with the increase of degree

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Training Error	0.855951	0.853836	0.27023	0.243957	0.168454	0.153101	0.110869	0.074539	1.18E-20

Small Data

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Training Error	0.386532	0.386018	0.215725	0.215499	0.209562	0.206802	0.205401	0.202191	0.20051

Large Data

Discussion

Test errors do not have this tendency

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Test Error	0.156486	0.146281	0.148816	0.161313	0.25901	0.302324	0.398575	0.413903	4.979396

Small Data

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Test Error	0.181946	0.178101	0.004612	0.004618	0.000477	0.002946	0.00507	0.00912	0.007493

Large Data

Discussion

Training errors decrease with the increase of dataset

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Training Error	0.855951	0.853836	0.27023	0.243957	0.168454	0.153101	0.110869	0.074539	1.18E-20

Small Data

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Training Error	0.386532	0.386018	0.215725	0.215499	0.209562	0.206802	0.205401	0.202191	0.20051

Large Data

Discussion

Test errors do have this tendency

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Test Error	0.156486	0.146281	0.148816	0.161313	0.25901	0.302324	0.398575	0.413903	4.979396

Small Data

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Test Error	0.181946	0.178101	0.004612	0.004618	0.000477	0.002946	0.00507	0.00912	0.007493

Large Data

Discussion

Value of N=9 is weird

Model	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
Training Error	0.855951	0.853836	0.27023	0.243957	0.168454	0.153101	0.110869	0.074539	1.18E-20
Test Error	0.156486	0.146281	0.148816	0.161313	0.25901	0.302324	0.398575	0.413903	4.979396

Table 1. Small Data Error without regularization

Thanks