Symbols, Patterns and Signals Coursework 1: Unknown Signal

Tan Han Yue Kenneth
WB18936

Introduction

The main objective of this coursework is to develop a program that estimates a model based on the given training data using the least squared regression method. Using the least squared regression method, a squared error is calculated for all the data points and they would be summed together. The SSE (sum of squared errors) is then used to indicate how well the line fits. The equation for calculating the SSE is as below:

$$R = \sum_i (y_i - \hat{y}_i)^2, \qquad \hat{y}_i = (a_1 + \ldots + a_N X_i)$$

Where $\hat{y}$ is the y hat value and the remaining would be explained further below.

Task 1: Degree of Polynomial

In this coursework, we will be working on three different least square regression function namely linear, polynomial, unknown. The first task of this coursework is to determine the degree of the polynomial function. This was done by inspecting the basic_3.csv file from the training data given. Basic_3.csv file was chosen because it does not contain any noise, and by carefully inspecting the scatter plot of it, a polynomial trend can be seen.

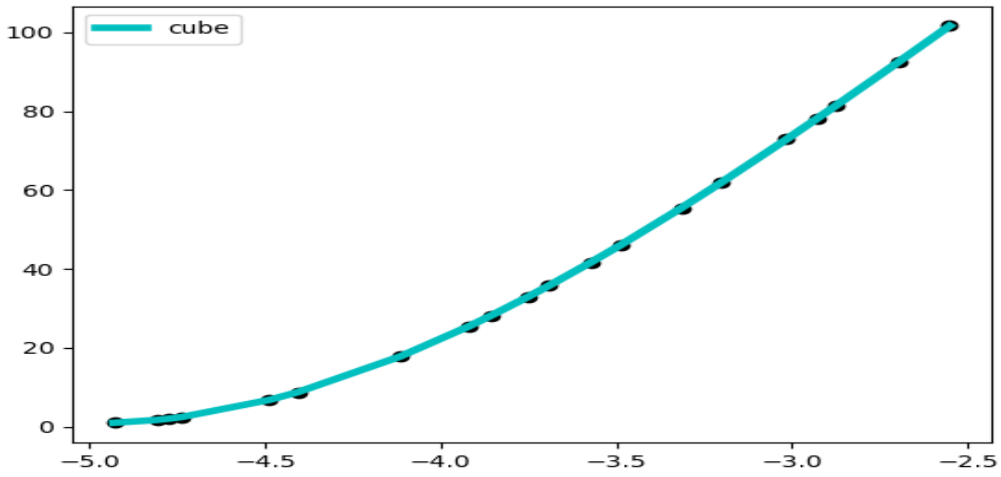


Figure 1: Plot of X^3 for Basic_3.csv

After comparing the plots of X^2 to X^5, it was visually observed that all plots looked like figure 1. Hence the only way to pick is to compare the SSE.

As seen in the table 1, the SSE of X^3 is the lowest and the degree of the polynomial function was chosen to be 3. The matrix form of the cubic least square regression is hence:

Degree	Sum of Squared Errors
X^2	15.743318683748834
X^3	**1.4384831613188792e-18**
X^4	2.6012776884995257e-12
X^5	4.505371714272158e-08

Table 1: Sum of squared errors of X^2 to X^5

$$\mathbf{a_{LS} = (X^T X)^{-1} X^T y}$$

where

$$y_i = \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix}, \qquad X_i = \begin{bmatrix} 1 & x_1 & {x_1}^2 & {x_1}^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_i & {x_i}^2 & {x_i}^3 \end{bmatrix}, \qquad a_{LS} = \begin{bmatrix} a_0 \\ \vdots \\ a_3 \end{bmatrix}$$

Task 2: Unknown function

The second task is to determine the unknown function. The task begins with inspecting the scatter plots of Basic.csv files. After a careful observation, it was concluded that neither linear nor exponential function could plot Basic_5.csv well. This indicates that the scatter plot of the Basic_5.csv could help decide what the unknown function is. Based on the plot, the unknown function was determined to be either cubic or sinusoidal.

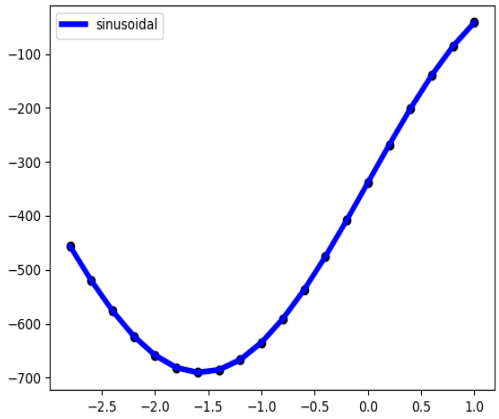


Figure 2: Basic_5.csv with Sinusoidal function

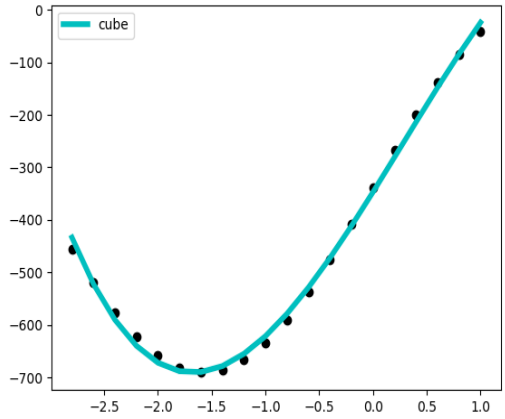


Figure 3: Basic_5.csv with cubic function

Next, both sinusoidal and cubic SSE are then compared to pick the unknown function.

Function	Sum of Squared Errors
Sinusoidal	**1.050131637030211e-25**
Cubic	2612.9757784503813

Table 2: Sum of squared errors for sinusoidal function and Cubic function

With the sinusoidal function having the lower SSE, it was chosen to be the unknown function. The matrix form of the sinusoidal least squared regression is shown below:

$$\mathbf{a_{LS} = (X^T X)^{-1} X^T y}$$

where

$$y_i = \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix} \qquad X_i = \begin{bmatrix} 1 & sin(X_1) \\ \vdots & \vdots \\ 1 & sin(X_i) \end{bmatrix} \qquad a_{LS} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

Task 3: Overfitting/Generalisation

The last task is to prevent the program to be overfitting. This is crucial as the purpose of the program is not to overfit but to generalise the data. Having a program that overfits means that the program would not be able to fit new data with the correct function as noise/outliers from the training data are picked up and taken in as a factor in choosing the best fit function. As this program favours polynomial over linear when there is a small difference in SSE among the both, the K-fold cross validation was implemented as a measure to prevent overfitting. The process of K-fold cross validation is as below.

1) In a segment of 20 data points, the data points are split into K groups. K in this case is 4. This would give us 5 data points per group.
2) 3 groups of data points would be used as train data, and 1 group of data point would be used as test data.
3) The least square regression would be fitted on the train data and the SSE would be calculated for the test data using the matrix from least square method on the train data.
4) Step 2 and 3 would be repeated for 4 times, with each group being the test data once. The mean of the 4 SSE would be taken to decide which function fits best.

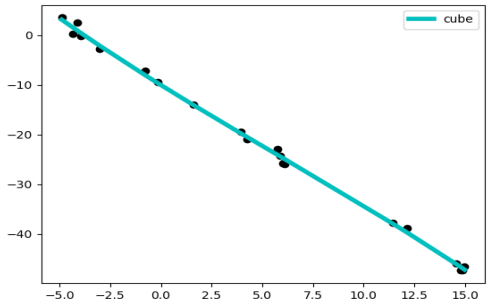
Group	Test data / Train data				SSE
1	**Test data**	Train data	Train data	Train data	Error 1
2	Train data	**Test data**	Train data	Train data	Error 2
3	Train data	Train data	**Test data**	Train data	Error 3
4	Train data	Train data	Train data	**Test data**	Error 4

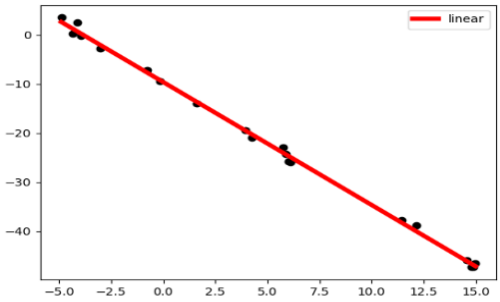
Table 3: Process 2 – 4

After implementing the K-fold cross validation, several improvements can be seen below.

Noise_1.csv:

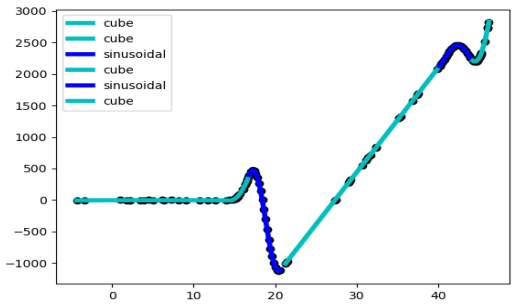
Before cross validation ***After cross validation***

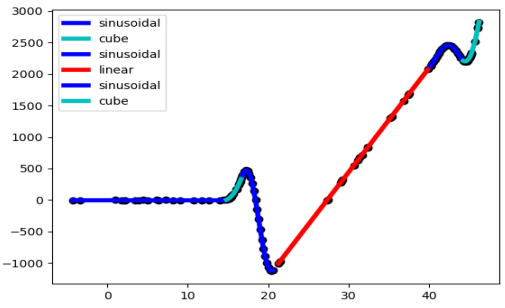




SSE: ***10.985055464509532*** ***12.207460140137117***

Adv_3.csv:





SSE: ***979.5922076723731*** ***1017.6979899741876***

Conclusion

As shown in the figures above, even though SSE have increased, the k-fold cross validation has successfully prevented overfitting. In Noise_1.csv, the function switched from Cubic to linear. In Adv_3.csv, the function of the first segment and fourth segment switched from cubic to sinusoidal and linear, respectively.