SPS CW1 Report

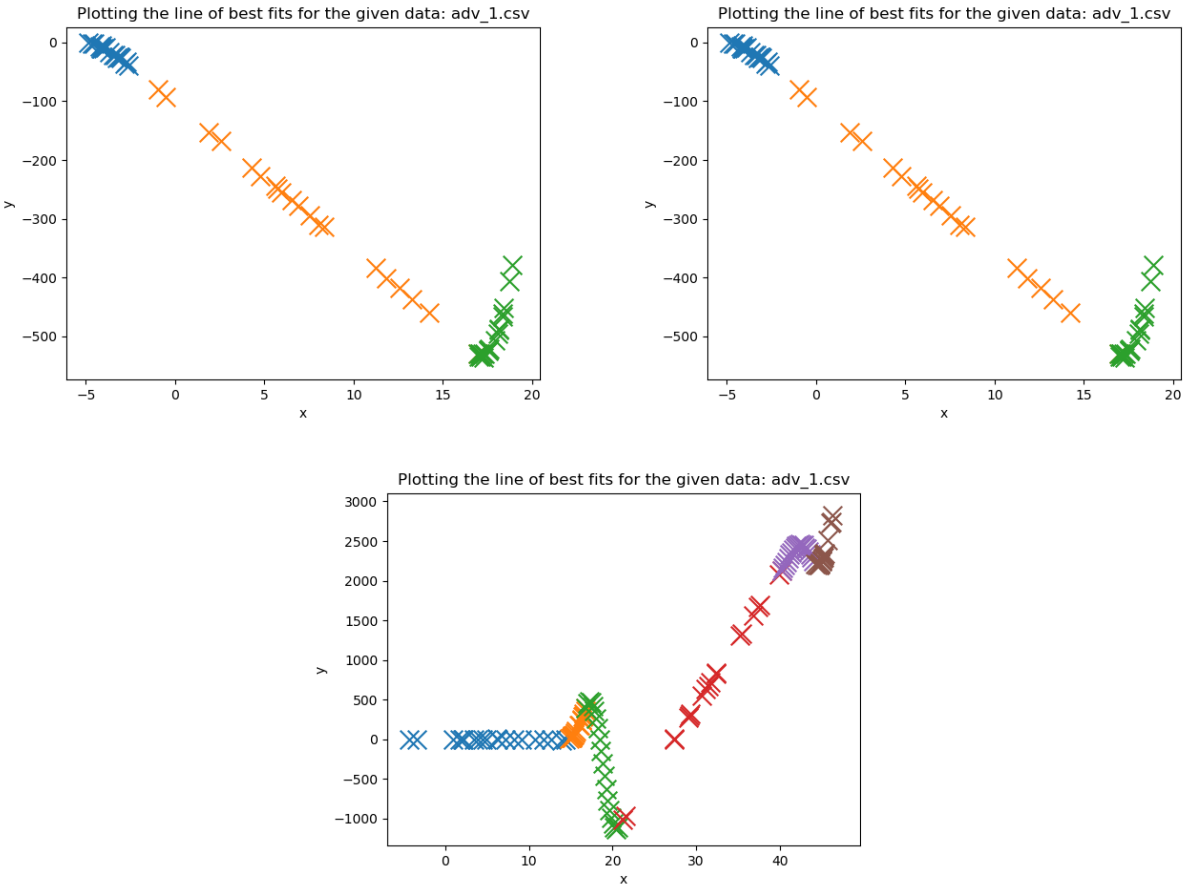
USERID: oa18502

March 2020

1 Introduction

An unknown signal has been provided and we are informed that we want to perform least squares regression on the signal data to find the 'lines of best fit'. We are told that the curves are either linear, polynomial or of a specific unknown type. By initially plotting the data, using the given "utilities.py" python file, the following graphs are produced for the important "adv_1.csv", "adv_2.csv" and "adv_3.csv" files respectively in table 1.

Table 1: Plotting each set of data.



It is evident from the third graph that the unknown function is a sine or a cosine function. Since it makes no difference which one is used for least squares regression, the sine function has been selected. A successful signal reconstruction program will visibly fit the data with minimal error, but additionally the solutions should not overfit the data. All the code is contained within the attached python file submission and it can also be found on my following github: https://github.com/MikeMNelhams/Basic-Least-Squares-Regression/tree/master. Furthermore, from a practical perspective, the code should run smoothly, reliably and quickly, whilst also being easy to read.

2 Regression methods

The three given categories for lines of best fit are: linear, polynomial and sinusoidal. For simplicity, linear regression is a sub-classification of polynomial regression, therefore there are now only two categories of polynomial and sinusoidal. All of the regression equations have been adapted directly from the lecture notes, which can be found via https://uob-coms21202.github.io/COMS21202.github.io/RuiLectures/Lec3-handout.pdf and https://github.com/UoB-COMS21202/lab_sheets_public/blob/master/lab_3/labsheet3_answers.ipynb. The given data should be given as a $20k$x2 matrix of coordinates, where $k \epsilon R$, with each different curve divided into 20-point segments. It is useful to split the columns of the matrix into two different vectors:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

which are going to be used in the least squares regression algorithm. For polynomial least squares regression of degree N, the equation for the line of best fit for any general coordinates (x, y), is assumed to be:

$$y = a_1 + a_2 x + a_3 x^2 + \cdots + a_N x^N \tag{1}$$

where a_1, a_2, a_3 and a_N are all coefficients, which are represented in the following column vector $\mathbf{a}$:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}.$$

Similarly, the simple sine least squares regression for any general coordinates (x, y) is assumed to be:

$$y = a_1 + a_2 \sin x \tag{2}$$

For the purpose of least squares regression, the matrix $\mathbf{X}$ is defined as:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^N \\ 1 & x_2 & x_2^2 & \cdots & x_2^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^N \end{bmatrix}.$$

The residual error between our approximation and the given data will therefore be the sum of the squared difference between the two, iterated over every data point. If the residual error is defined as a vector function $R(\mathbf{a})$, then:

$$\begin{aligned} R(\mathbf{a}) &= \sum_i (y_i - y)^2 \\ &= \sum_i (y_i - (a_1 + a_2 x + a_3 x^2 + \cdots + a_N x^N))^2 \\ &= ||\mathbf{y} - \mathbf{X}\mathbf{a}||^2. \end{aligned} \tag{3}$$

Solving the equation (3) for when $R(\mathbf{a}) = 0$ produces the following matrix equation:

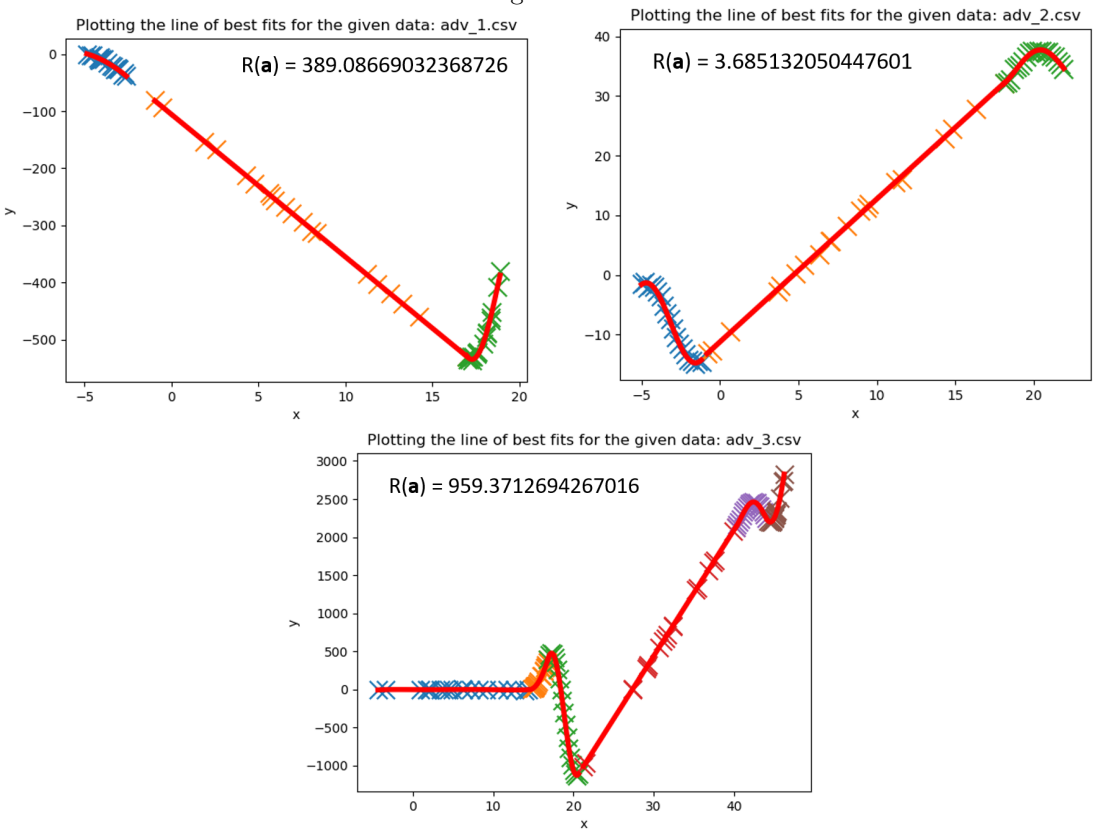
$$\mathbf{a} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \tag{4}$$

Equation (4) can quickly generate the coefficients to the least squares solution which can be substituted back into equations (1) and (2) to generate the lines of best fit. The residual error for the sum over all the segments in the signal is $R_t(\mathbf{a})$.

3 Results

The code executes the least squares regression algorithm accurately and follows the given formatting criteria provided. The results for the given files "adv_1.csv", "adv_2.csv" and "adv_3.csv" are shown in table 2.

Table 2: Plotting each line of best fit.



Firstly, to avoid overfitting the highest allowable polynomial degree for the algorithm is 5. The reason for this is because with only 20 data-points it is very difficult to know whether or not a curve is of a high degree. This is due to the fact that higher degree polynomials may exhibit more rapid changes in gradient and therefore it is difficult to determine if the polynomial is noisy or of a high degree, with such few data points. Subsequently, after some trial and error, 5 seems highly reasonable.

To choose an appropriate curve, the algorithm first sets the error to infinite. The residual error for every type of curve is calculated, then the system goes through the sorted list of errors and selects the least-error curve if it reduces the error more than one standard deviation of the previous method. This is a simple method which is not perfect, but it carefully avoids the majority of common overfitting errors and the curve will usually match the data without overfitting if the data does not exceed a 5th degree polynomial. It is more important to have a good error, which visually matches the curve without overfitting, than have no error and the curve overfits, because it therefore holds very little useful information.