Curve fitting using Least squares Approximation

Mathematics for IT, C2 Assignment

Rituvendra Singh (MIT2020079)*

* *Semester I, MTECH(IT)*

Department of Information Technology

Indian Institute of Information Technology (IIIT), Allahabad

Uttar Pradesh (INDIA)

***Abstract*—The method of Fitting Lines, curves and surfaces to data points is crucial in various scientific areas. In this paper, we have discussed and analyse the method of solving a curve fitting problem using Method of Least squares approximation and revisited the concepts of orthogonality, Least squares and it's application in solving a system of linear equations that do not have a solutions. Sometimes, even though a solution does not exist, often we are satisfied with just getting a approximate solution. Determining this approximate solution will act as basis of finding a best fit curve problem. Here we have demonstrated a method of Least square approximation for curve fitting is proposed in this study. The key methods / mechanisms used are orthogonal projection and Least Square approximations for curve fitting.**

***Index Terms*—Orthogonality, Least squares, Curve fitting, Orthogonal projection.**

I. INTRODUCTION

Efficient and precise curve/line fitting plays an important role and serves in large areas as an important module in multiple applications. We have considered a common problem here that often comes in practical applications: which is to fit a curve on the data points given. This problem is commonly stated as a typical problem of nonlinear least squares.

In this study we will need to find the nearest point on a subspace to a given data point. The nearest point has the characteristic that the difference is orthogonal, or perpendicular, to the subspace between the two points. For this reason, we will first review some concepts of orthogonality in Linear algebra.

A. Orthogonal vectors

The dot product of two vectors x,y in $\mathbb{R}^n$ is given by [6],

$$\langle u \cdot v \rangle = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$$

Definition : Two vectors u and v in $\mathbb{R}^n$ are perpendicular or orthogonal if $\langle u, v \rangle = 0$.

B. Orthogonal complements

- *Definition :* We say that the set is a orthogonal complement of W if The set of all vectors u in this set are orthogonal or perpendicular to all vector w in W, and denote by $W^{\perp}$. [1]
- A vector x is in the orthogonal complement set $W^{\perp}$ if and only if x is orthogonal to all vectors in a set that spans subspace W.
- $W^{\perp}$ is a subspace of $\mathbb{R}^n$ for any subset W of $\mathbb{R}^n$.
- Any vector is always orthogonal to Zero vector. [1]

C. Orthogonal projections

What vector within a linear subspace of $\mathbb{R}^n$ best approximates a given vector in $\mathbb{R}^n$? The next section provides answer to this question. This is a way of determining the nearest vector on a subspace to a given vector. The vector $\hat{y}$ is called the orthogonal projection of vector y onto W and is often denoted as $proj_w y$. The orthogonal projection of vector y onto W is illustrated in given below figure.

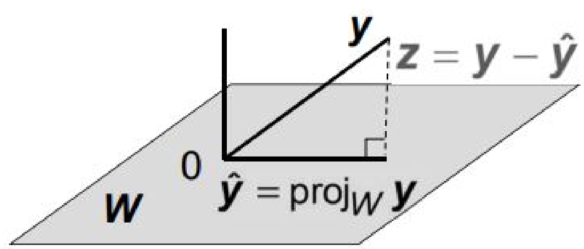

Fig. 1: orthogonal projection of y onto W [2]

D. Approaches for Curve fitting

A method for determining the best-fitting curve for a given set of data points is called as curve fitting. Two popular methods of Curve fitting are :-

1) Interolation: In this method we construct a parametric curve that passes through (interpolates) a set of given data points. The given Figure. illustrates a Linear Interpolation for some data points.

2) Least Squares method: By "Least squares" we mean to say that the we minimizes the sum of the squares of the errors generated in the solution of each single equation by overall solution. [3] We will discuss about it in detail in later sections.

E. Least Squares problem

- Generally, finding approximate solutions of equation allows the closest vector on a subspace to a given vector to

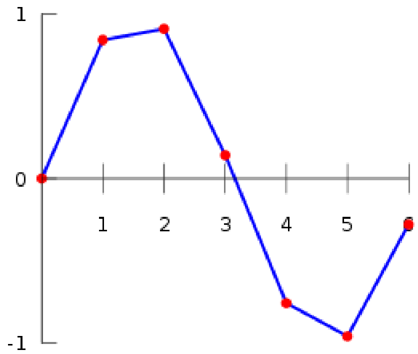

Fig. 2: Method of Interpolation
Source: wikipedia.org

be computed. This becomes a problem of orthogonality: one needs to consider which vectors are orthogonal to the subspace.

- A solution is needed for a linear system $Ax = b$, but none exists. Finding an x that makes Ax as close as possible to b is the best thing one can do. The general least-squares problem is simply to find an x that makes $\|b - Ax\|$ as small as much possible. The term least-squares derives from the fact that $\|b - Ax\|$ is the square root of a sum of squares.
- Definition : If A is an mxn matrix and b is in $\mathbb{R}^m$, a least-squares solution of $Ax = b$ is an in $\mathbb{R}^n$ such that [8]

$$\|b - A\hat{x}\| < \|b - Ax\| \quad \text{for all x in } \mathbb{R}^n.$$

- The vector Ax will necessarily be in the column space Col A no matter what x is chosen. So we look for an x that makes Ax the closest point in Col A to b.

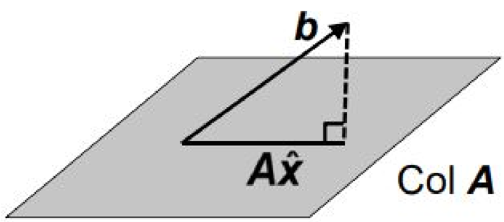

Fig. 3: orthogonal projection of y onto W [2]

F. Line/Curve fitting

One can asks in data modelling: ”On which line/curve/-surface is my data points supposed to lie on? ”.It is a best-fit problem and using a simple implementation of the least-squares approximation method, this can be solved. The mathematical method for finding the best-fitting curve for a given set of points by minimising the sum of the squares of the residuals of the curve points is the Least Squares Curve Fitting. Instead of the offset absolute values, the sum of the squares of the offsets is used since this allows the residuals to be viewed as a continuous differenceable quantity. We include an application of the method of least squares to curve fitting in data modelling in this subsection. Let us see a example regarding this,

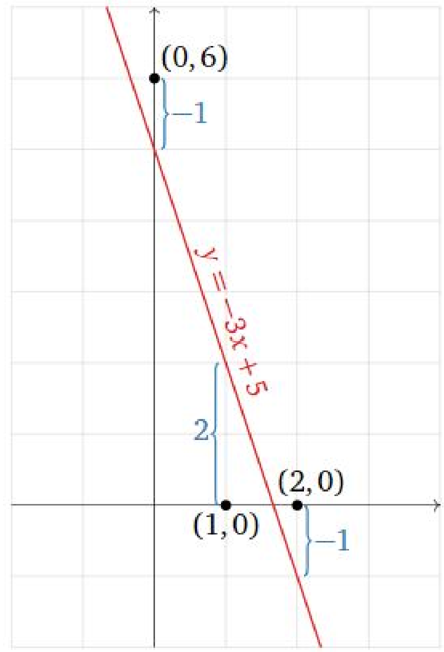

Fig. 4: Example of best-fit line that minimizes the sum of the squares of these vertical distances [2]

Suppose that we have measured three data points

$$(0, 5), \quad (1, 1), \quad (3, 0),$$

And here we want the points to be located on a line. These three points, of course, do not necessarily lie on a single line, but this may be due to our calculation mistakes. So here we want to predict the line on which they should lie? For a (non-vertical) line, the general equation is.

$$y = Mx + B$$

The following equations would be satisfied, Since we want our given data points to lie on this line, so, we have :

$$5 = M * 0 + B$$

$$1 = M * 1 + B$$

$$0 = M * 3 + B$$

We attempt to solve the above equations in the unknowns M and B in order to find the best-fit line. There is no real solution, since the three points do not necessarily lie on a graph, so we calculate a least-squares solution instead.

Putting our linear equations into matrix form, we have to solve $Ax = b$ for

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 3 & 1 \end{bmatrix} \quad x = \begin{bmatrix} M \\ B \end{bmatrix} \quad b = \begin{bmatrix} 5 \\ 1 \\ 0 \end{bmatrix}$$

We solved this problem and the least-squares solution to $Ax = b$ is given as,

$$\hat{x} = \begin{bmatrix} M \\ B \end{bmatrix} = \begin{bmatrix} -3 \\ 5 \end{bmatrix}$$

hence our line of best fit is given by,

$$y = -3x + 5.$$

The least-squares solution $\hat{x}$ minimizes the sum of the squares of the values of the vector $b - A\hat{x}$. $A\hat{x}$ is the vector whose entries are the y-coordinates of the graph of the line at the values of x we specified in our given data points, and b is the vector whose entries are the y-coordinates of those data points. The difference $b - A\hat{x}$ is the vertical distance of the graph from the data points.

II. PROPOSED METHODOLOGY

A general curve fitting problem in data modelling have the following form: we have a given number of data points (x,y) are, and we need to find a function that best approximates these points.

$$y = B_1 g_1(x) + B_2 g_2(x) + \cdots + B_n g_n(x)$$

here $g_1, g_2, \ldots, g_n$ are fixed functions of x. For different exmaples like best fit for a line, parabola , nonlinear curve etc we will have different functions $g_n(x)$. We evaluate the above equation on the given data points to obtain a system of linear equations in the unknowns $B_1, B_2, \ldots, B_n$. The $g_I(x)$ simply becomes numbers once they are evaluated, so they does not matter and we determine the least-squares solution. The resulting best-fit function that we obtained will minimise the summation of the squares of the vertical distances from the curve of y=f(x) to our original given data points.

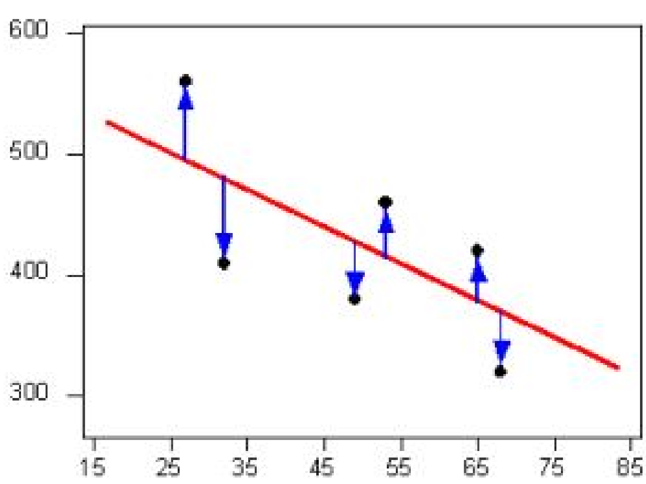

Fig. 5: Method of Least square approximation

In this study we are going to find a best fit curve for data points scattered as a sine function. We will apply the theory of orthogonal projection in this least squares regression problem to find the best fit curve. The proposed approach in this research work can be summarised by the following steps mentioned in detail.

The steps for our proposed approach are as follows :-

- The first step is to plot the data points and find a suitable mathematical function that demonstrates the trend in data points. Some example of commonly used functions are straight line, parabola , ellipse , sine/cosine functions etc.
- Of course there will be some error since these mathematical function F(X) will not perfectly fit all data points.
- Apply Method of least sqaure and determines the coefficients such that the sum of the square of the deviations between the data and the curve-fit is minimized.
- The resulting best-fit function that we obtained will minimise the summation of the squares of the vertical distances from the curve of y=f(x) to our original given data points.

III. EXPERIMENTAL ANALYSIS

Here we have generated data points since we don't have any already. We have used python numpy library and genearated suitable data points for our propsed approach using random number generator in this python library. Data points were saved in appropraite file format to be fetched in python script later. Here we have total 50 data points generated randomly.

Plotting of data points and visualisation of plots is performed using python libray matploit. [4] Our Plot data points is shown in Figure 5.

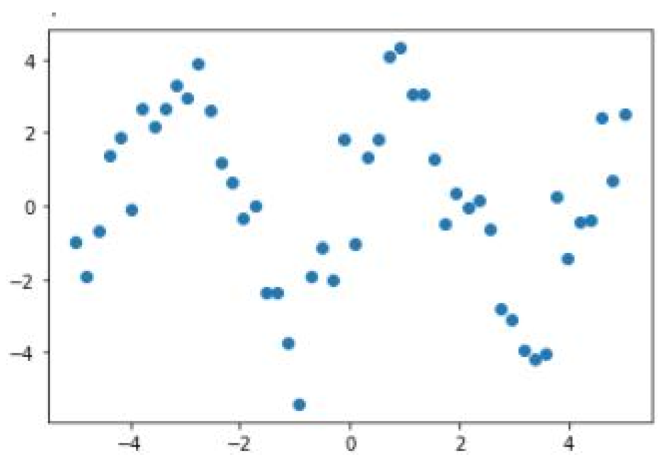

Fig. 6: Plotting of data points and it's visualization

Since our goal was to demonstrate curve fitting of sinusoidal waveform on data points hence we choosen data points suitable for this experiment. Our data points are scattered here around a sine function .

The Resultant "Best fit " curve that we achieved in this experiment was plotted using python library matploit. The Least squares method is impplemenated using Python scipy library. [4] The curve fit function is used from this python library which implement Least sqaure method for determing the best fit curve. it is shown in Figure 7 . Here a best fit curve is plotted for 50 given data points.

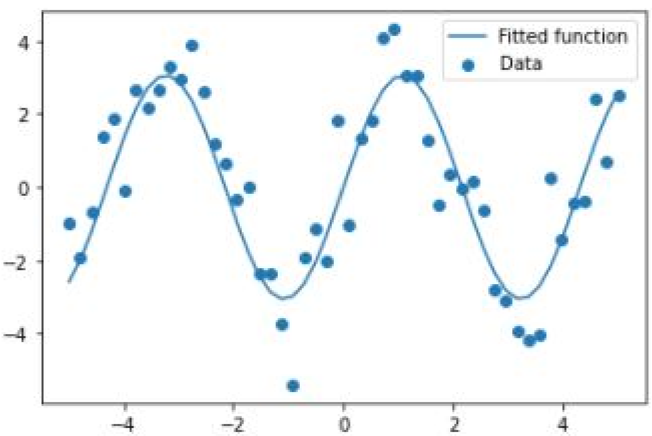

Fig. 7: Final result for "Best fit" curve

IV. Applications

Least squares approximation method are as follow:

Some prominent applications of the curve fitting models in engineering are:

1) Trend Analysis: Predicting values of dependent variables, may include extrapolation beyond data points or interpolation between data points.
2) Hypothesis Testing: Comparing existing mathematical models with measured data.
3) Regression analysis
4) Few of the real life applications of curve fitting are :
   a) Analysis of the structures in the Architectural buildings.
   b) To develop numerical calibration procedures for a river water quality model.
   c) Analysis of the shape of the cooling towers at Thermal Power Plant.

V. Conclusion

In this paper, we have demonstrated how we can solve a curve fitting problem in data modelling by using method of Least square approximations. We generated data points suitable for our research work . Fitting a curve onto onto some data points is clearest and crucial application of Method of Least squares. Experiment results demonstrates that the proposed approach obtains best fit curve for Curve fitting to a sinusoidal function wave form data points.

Acknowledgment

I would like to thank our Course instructor Dr Mohammed Javed and our mentor, Mr. Rajesh Bulla, who have provided the great opportunity to do this wonderful work on the subject of Singular Value Decomposition.

References

[1] Strang, Gilbert, et al. Introduction to linear algebra. Vol. 3. Wellesley, MA: Wellesley-Cambridge Press, 1993.
[2] Margalit, Dan, and Joseph Rabinoff. "Interactive Linear Algebra." Georgia Institute of Technology (2018).
[3] Bagai, 2020. Least Square Method. [online] Slideshare.net.Available at: https://www.slideshare.net/somyabagai/least-squaremethod-20593774.
[4] Scipy-lectures.org. 2020. 1.6.12.8. Curve Fitting-Scipy Lecture Notes. [online] Available at: https://scipylectures.org/intro/scipy/auto_examples/plot_curve_fit.html
[5] Zhang, Jian Qiu, et al. "Sinewave fit algorithm based on total least-squares method with application to ADC effective bits measurement." IEEE transactions on Instrumentation and Measurement 46.4 (1997): 1026-1030.
[6] Sargent, Thomas, and John Stachurski. Quantitative economics with python. Technical report, Lecture Notes, 2015.
[7] python4mpia leactures notes [online] Available at: https://python4mpia.github.io/fitting_data/least-squares-fitting.html
[8] Lecture notess Author: Vadim Olshevsky [online] Available at: https://www2.math.uconn.edu / olshevsky/classes/ 2018_Spring/math3511/ orthogonality.pdf

Appendix

To Run this code please follow the following steps :

1) Save the given code with (.py) extension as a python file
2) Make sure your System has python compiler installed on it.
3) Install the following python library using following command in python CLI.
   pip install numpy
   pip install matplotlib
   pip install scipy
4) Run the Filename.py file

Code for implementation of this paper is given below :

```
import numpy as np

# Seed the random number generator for reproducibility
np.random.seed(0)

x_data = np.linspace(-5, 5, num=50)
y_data = 2.9 * np.sin(1.5 * x_data) + np.random.normal(size=50)






# And plot it
import matplotlib.pyplot as plt
plt.figure(figsize=(6, 4))
plt.scatter(x_data, y_data)

from scipy import optimize

def test_func(x, a, b):
    return a * np.sin(b * x)

params, params_covariance = optimize.curve_fit(test_func, x_data, y_data,  p0=[2, 2])

print(params)




#Plot the final best fit curve diagram
plt.figure(figsize=(6, 4))
plt.scatter(x_data, y_data, label='Data')
plt.plot(x_data, test_func(x_data, params[0], params[1]),
         label='Fitted function')

plt.legend(loc='best')

plt.show()
```

Listing 1: Code for this paper