Assignment 3: Fitting Data To Models

V.S.S.P.R.KOUSHIK [EE19B061]

march,2021

Abstract

This week's assignment topic is fitting data to models.
Reading data from files and parsing them

- Analysing the data to extract information
- Study the effect of noise on the fitting process
- Plotting graphs

1 Extracting and loading the data

Run the python code *generate_ data. py*, to create file *fitting. dat*.

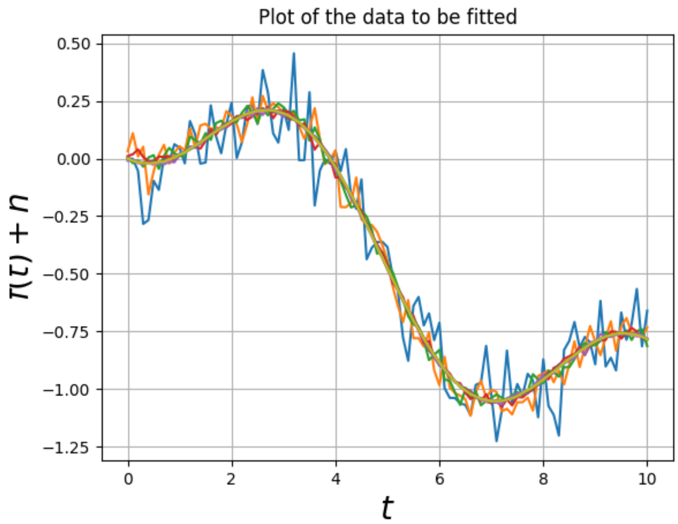

Figure 1: Data plot

This file contains 101 rows and 10 coloums of data. The first column is the time values and the next nine rows are the noisy values of a function shown in graph. Each column of data has different standard deviation which is given by the python command.

```
stdev = logspace(-1,-3,9)
```

2 Plotting the true and noise added plots

First coloum is assigned to t as they are time values. Remaining coloums define the the 9 noisy data plots. second column is assigned to d. The function is defined here.

- f(t) =1.05J2(t)-0.105t

```
def g(t,A,B):
    return A*sp.jn(2,t) + B*t
```

By taking all 9 noisy f(t) values from data and plotting against time. from the function the true value is known and plotted.

```
for i in range(1,10):
    plot(t,data[:,i],label="a=%.4f"%stdev[i-1])
plot(t,ft,label="True Value",color='black',linewidth=2)
```

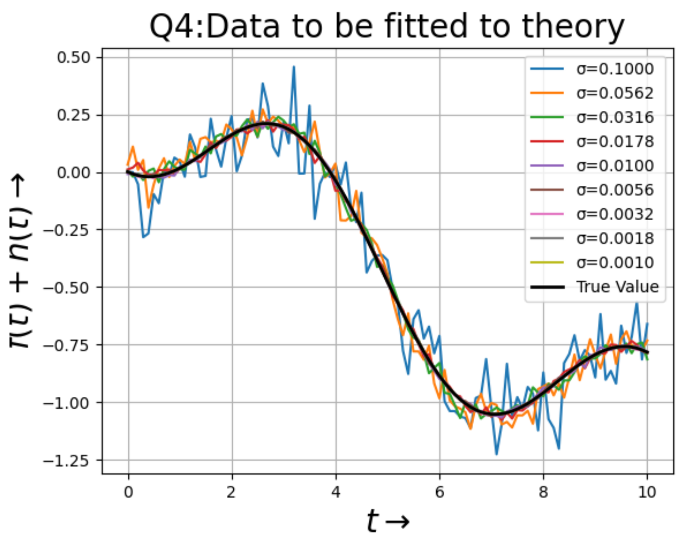

Figure 2: True and 9 noisy added plots

3 The Error Plot

Error bars are graphical representations of the variability of data and used on graphs to indicate the error or uncertainty in a reported measurement. They give idea about precision and shows variation from reportd value to true value.In this question, the error bars are to be shown for every 5th element of first noisy data and true value.

```
plot(t,ft,label="True Value",color='black',linewidth=2)
errorbar(t[::5],d[::5],stdev0,fmt='ro',label='Noise')
```

The graph obtained by plotting every 5th data point with errorbars and the original data is as follows:

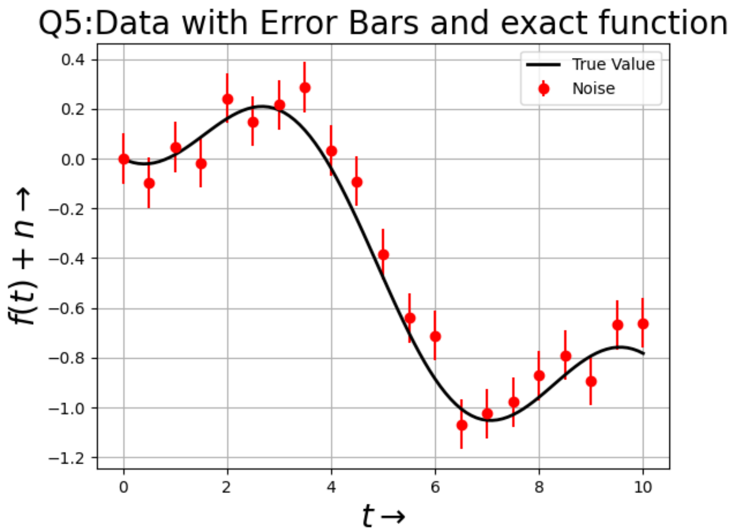

Figure 3: Errorbar plot

4 Matrix Equation

The matrix M is created and multiplied with (A,B) matrix will give rise to the true value function. This can also be verified by substituting $A = 1.05$ and $B = -0.105$. In order to compare 2 matrices, we use the function `array_equal()`. The python code is shown below:

```
M = empty((n,2))
for i in range(n):
M[i] = (ss.jn(2,t[i]),t[i])
P = array([1.05,-0.105])
Q = dot(M,P)
```

```
if array_equal(Q,ft):
print("Two vectors are equal.")
else:
print("Two vectors are not equal.")
```

5 The Mean Squared Error

The mean squared error is the average of squares of error between the noisy data and the true functional data. The equation is

$$\varepsilon_{ij} = (\frac{1}{101})\sum_{k=0}^{101}(f_k - g(t_k, A_i, B_j))^2$$

The error is calculated for $A = 0, 0.1, ..., 2$ and $B = 0.2, 0.19, ..., 0$ and a contour plot is plotted.

```
A = array([0.1*i for i in range(21)])
B = array([-0.2+0.01*i for i in range(21)])
E = zeros((21,21))
for i in range(21):
    for j in range(21):
        for k in range(n):
            E[i][j] += ((d[k]-g(t[k],A[i],B[j]))**2)/n
K,L = meshgrid(A,B)
```

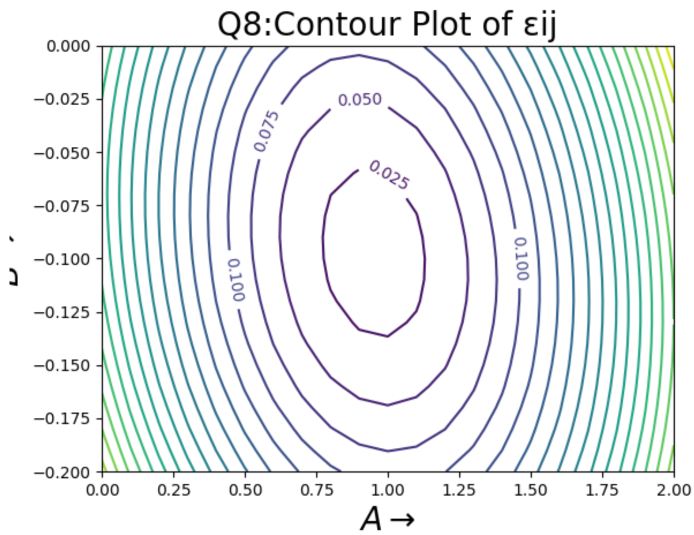

Figure 4: Contour plot

Conclusion

From the above plot, we can conclude that there exist only one minimum for ε.

6 Error Computation in estimation of A and B

It is possible to try and compute the best measure for A and B from the matrix M by using the *lstsq()* function form *scipiy.linalg*. Using this we can calculate the error in the values of A and B. The python code snippet is as follows:

```
Ea = empty((9,1))
Eb = empty((9,1))
for j in range(9):
    AB = linalg.lstsq(M,data[:,j+1],rcond=None)
    Ea[j] = abs(AB[0][0]-A0[0])
    Eb[j] = abs(AB[0][1]-A0[1])
```

The plot of the error in A and B against the noise standard deviation is:

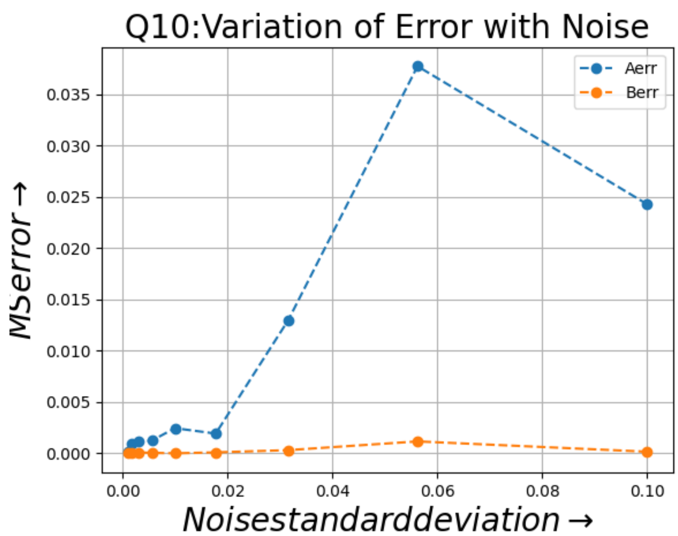

Figure 5: Error vs Standard deviation

Conclusion

From the plot , we can see that the plot is not varying linearly with noise.

7 Error Computation in estimation of A and B in Loglog.

It is able to compute the best measure for A and B and plot in loglog b using the *loglog* function from *matplotlib.pyplot*.The python code is as shown:-

```
figure(4)
loglog(stdev,Ea,'ro',label='Aerr',)
errorbar(logspace(-1, -3, 9), Ea, std(Ea), fmt='ro')
loglog(stdev,Eb,'go',label='Berr')
errorbar(logspace(-1, -3, 9), Eb, std(Eb), fmt='go')
title("Q11:Variation of Error with Noise",size=20)
xlabel(r'$n\rightarrow$',size=20)
ylabel(r'$MS error\rightarrow$',size=20)
grid(True)
legend()
```

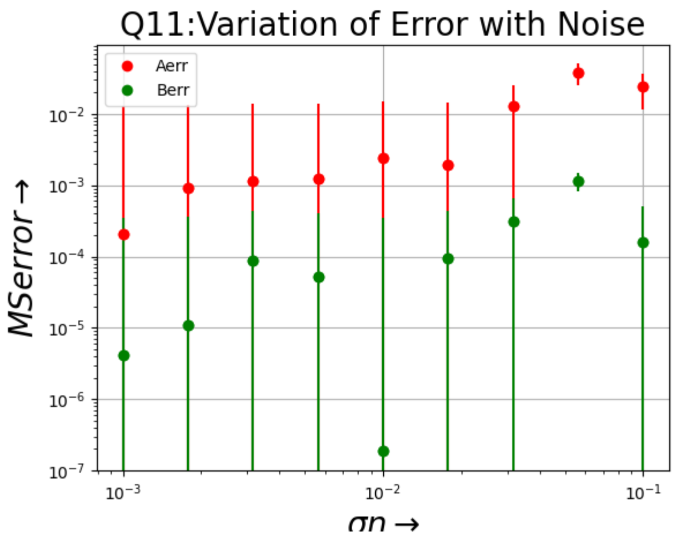

Figure 6: loglog. plot of (Error vs Standard deviation)

Conclusion

From the plot we can see that log error is not linearly varying with log noise.

Inference

The given noisy data was extracted and the best possible estimate for the underlying model parameters were found by minimizing the mean squared error. This is one of the most general engineering use of a computer, modelling of real data. The method of least squares assumes that the best fit curve of a given type is the curve that has the minimal sum of deviations, i.e., least square error from a given set of data. It reduces the error and gives the best fitting data.