Assignment 3: Fitting Data To Models

Hariharan P EE20B042

February 13, 2022

Brief Up

This assignment is analysing a noisy data and fitting MS fit to the given model.
Tasks:

- Extracting information from the generated data file.
- Analysing the effects of noise on the fitting process.
- Plotting different types of plots.

1 Obtaining the data from given python file:

Run the python code *generate_ data. py* to obtain the file *fitting. dat* which has the dat to analyse.

File : 10 columns X 101 rows of data. The first column has the time values and the next nine columns are the values of a function along with noisy . Each column of data has different standard deviation of the form

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

2 The g() function and noise:

The g() function is defined as the following code snippet:

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

On plotting the actual function's value along with all the 9 noise added values, the following plot is generated. The Figure 0 that was asked in Q.3 and Q.4 is obtained. The python code snippet for plotting this graph is :

```
plot(time,act_f,label="actual Value",color='black',linewidth=4)
for i in range(1,10):
plot(time,givendata[:,i],label="stdev=%.4f"%stdev[i-1])
```

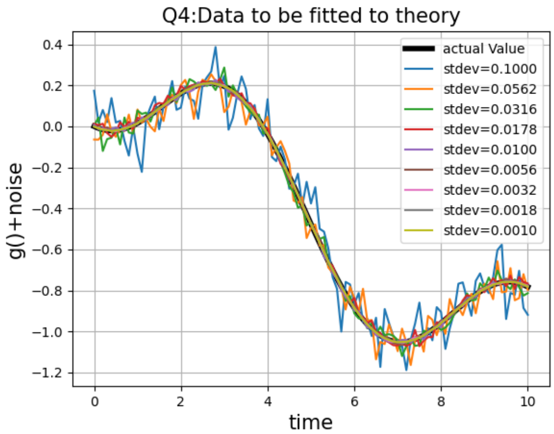

Figure 1: Data to be fitted to theory

3 The Errorbar plot -Error visualiser:

The errorbars for the first "data" column are plotted using the **errorbar()** function. The python code snippet to plot that:

```
plot(time,act_f,label="f(t)",color='black',linewidth=4)
errorbar(time[::5],col2[::5],stdev[0],fmt='ro',label='errorbar/noise')
```

The plot created for every 5th data point with errorbars and the original data is:

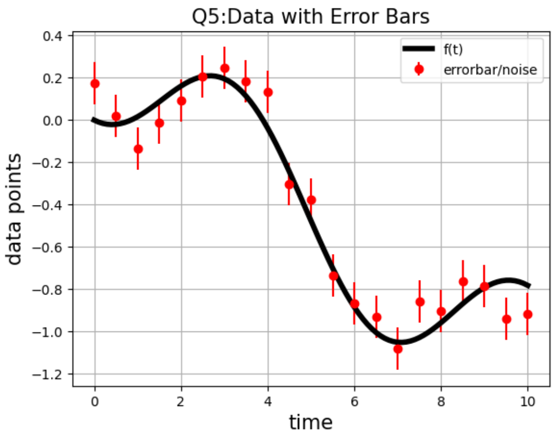

Figure 2: Errorbar plot

4 Verfication of matrix with the function:

The actual value function is created using a matrix equation. The matrix M created using two column vectors, when dot multiplied with P matrix, which is (A,B) should give rise to the actual function. This is verified as a part of Q6 by substituting $A = 1.05$ and $B = -0.105$. we compare 2 matrices using the function `array_equal()`. The python code snippet is :

```
col1=zeros((x,1))
for j in range(x):
col1[j] = (special.jn(2,time[j]))
M=c_[col1,time]
p = array([1.05,-0.105])
G = dot(M,p)
if array_equal(G,g(time,1.05,-0.105)):
print("generated matrix matches with the actual values")
else:
print("generated matrix does not match with the actual values")
```

5 The mean squared error between the data (f(k)) and the assumed model:

The mean squared error is one of the method to deduce the representation of data from a model.The mean squared error is calculated between the noisy data and the actual functional data. The formula for calculating is :

$$\varepsilon_{ij} = (\frac{1}{101})\sum_{k=0}^{100}(f_k - g(t_k, A_i, B_j))^2$$

The python code snippet to calculate the mean squared error is:

```
Ai = [0.1*i for i in range(21)]
Bi = [0.01*i-0.2 for i in range(21)]
Err = zeros((21,21))
for i in range(21):
for j in range(21):
for k in range(x):
Err[i][j] += ((col2[k]-g(time[k],Ai[i],Bi[j]))**2)
Err=Err/x
```

The contour plot for εij (Errij) for various values of A and B is:

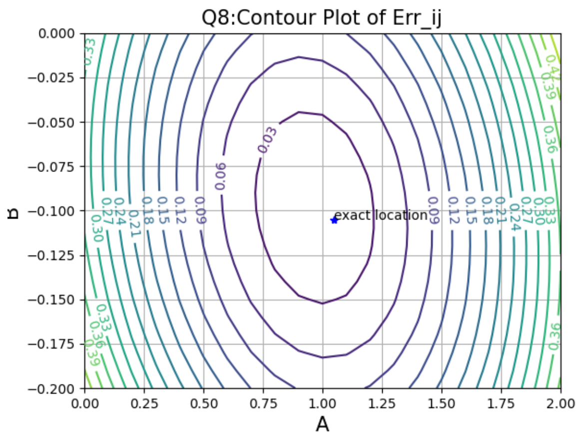

Figure 3: Contour plot for values of A and B

Conclution

From the above plot, we conclude that there exists only one minimum for εij.

6 Deduction of best estimate, error for A and B:

Now we deduce the best estimate for A and B from the matrix M by using the *lstsq()* function form *scipiy.linalg* by which we calculate the error in the values of A and B. The python code snippet is:

```
Err_A = zeros((9,1))
Err_B = zeros((9,1))


for i in range(1,10):
best_est_AB = linalg.lstsq(M,givendata[:,i],rcond=None)
Err_A[i-1] = abs(best_est_AB[0][0]-1.05)
Err_B[i-1] = abs(best_est_AB[0][1]+0.105)
```

The plot of the error in A and B against the standard deviation is shown in the next page:

We can also plot the same graph in loglog scale. This plot is also shown in the next page :

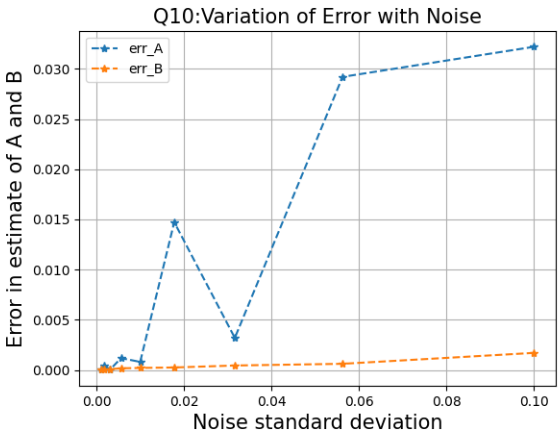

Figure 4: Error vs Standard deviation

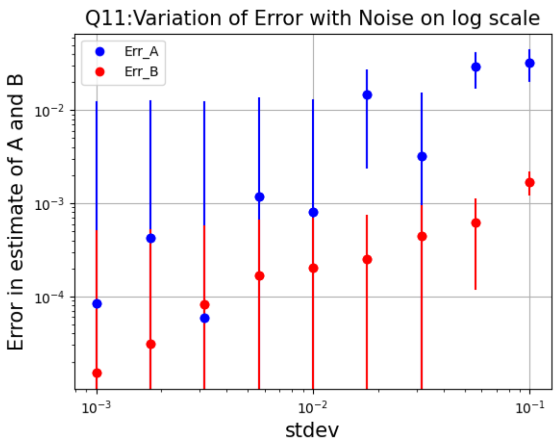

Figure 5: Error vs Standard deviation in loglog scale

Conclusion

From Figure 4, it is evident that the plot is not linear for the error of A but the error of B is slightly linear. Also from Figure 5, it is made sure that the plot is not linear in the loglogscale case too for error of A but the error of B is approximately linear .

Inference

The given data with noise is extracted, analysed and the best estimate for the model parameters ie. A and B are found using the minimum value of the mean squared error.