EE2703 : Applied Programming Lab
Week 3: Fitting Data to Models

Surya Prasad S
EE19B121

March 3, 2021

Objective:

The following are the objectives of this week's assignment:

- To take data from a noisy environment and process it
- To study how to fit the data into a specified model
- To study how noise affects the fitting

Theory:

Run the code "generate_data.py" to generate a set of data following the equation

$$f(t) = 1.05J_2(t) - 0.105t + n(t) \tag{1}$$

With n(t) being various amounts of noise. The noise in each data set follows the normal distribution,

$$\mathrm{P}(n(t)|\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{n(t)^2}{2\sigma^2}\right)$$

where σ is given by sigma=logspace(-1,-3,9)

"generate_data.py" stores matrix of values in "fitting.dat" with the first column being time and the rest being different noises being added to the function. This data is to be fitted into the function,

$$g(t; A, B) = AJ_2(t) + Bt \tag{2}$$

with true values of A and B being:

$$A = 1.05,\ B = -0.105$$

For this problem, we first create the matrix M which contains t and the corresponding values of the Bessel function $J_2(t)$. p is a vector for the coefficients A and B. The function $g(t; A, B)$ is the product of M and p.

$$g(t; A, B) = \begin{pmatrix} J_2(t_1) & t_1 \\ \ldots & \ldots \\ J_2(t_m) & t_m \end{pmatrix} \cdot \begin{pmatrix} A \\ B \end{pmatrix} = M \cdot p \tag{3}$$

Next, the mean squared error of the data is calculated for each set of noisy values with $A = 0, 0.1, ..., 2$ and $B = -0.2, -0.19, ...0$ using the formula:

$$\epsilon_{ij} = \frac{1}{101} \sum_{k=0}^{101} (f_k - g(t; A_i, B_j))^2 \tag{4}$$

Various plots are plotted for visualisation of the errors for different A and B values.

Plotting Analysis:

Extraction and Visualisation of Data:

1. The data was obtained by running the given "generate_data.py", which is plotted as follows:

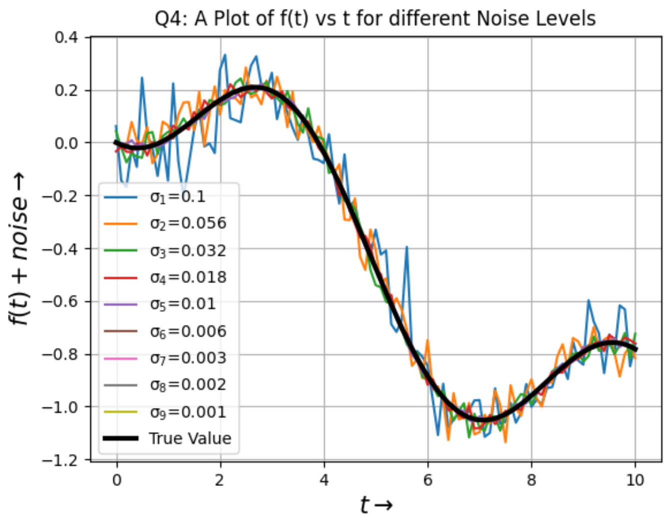

Figure 1: Data Plot

2. A file named "fitting.dat" is generated as the output. The binary file contains 10 columns of data: first column corresponding to the time stamps and the other 9 columns correspond to the function along with the noise.

Obtaining the Function:

1. The python function to compute $g(t; A, B) = AJ_2(t) + Bt$ is as follows:

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

2. We compare the actual value with the values produced by "generate_data.py" in Figure 1 above.

Errorbar Plot:

1. The library matplotlib offers a very convenient way to visualize the functional values along with their errors through Errorbar plots. The Errorbar plot of the first column is plotted below:

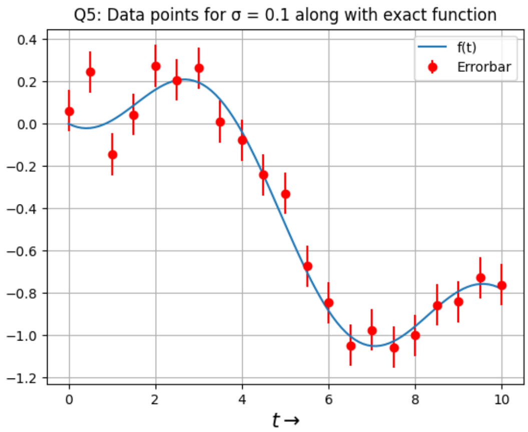

Figure 2: Errorbar plot of the first column

2. The bright red dot in the center is the value of the first column

Predicting A and B for least MSE:

The Mean Squared Error has been computed for the given range of A and B with the code as follows:

```
a = np.linspace(0,2,21)

b = np.linspace(-0.2,0,21)

MSE = np.zeros((len(a),len(b)))

for i in range(len(a)):

    for j in range(len(b)):

        MSE[i, j] = ((y[:, 0] - g(x,a[i],b[j]))**2).mean(axis=0)
```

Contour Plot:

1. Contour plots are plotted using the library matplotlib of the mean squared error with various values of A and B

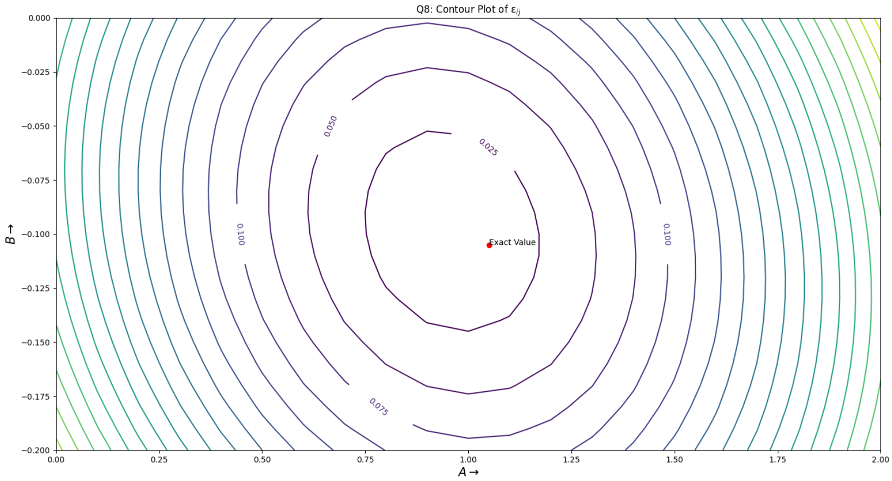

Figure 3: Contour Plot

From the graph, the contours has only one minimum and seems to converge near the true value at $A = 1.05$ and $B = -0.105$.

Error Analysis with plots:

1. The plot of error in approximation with the standard deviation of the noise in the data is as follows:

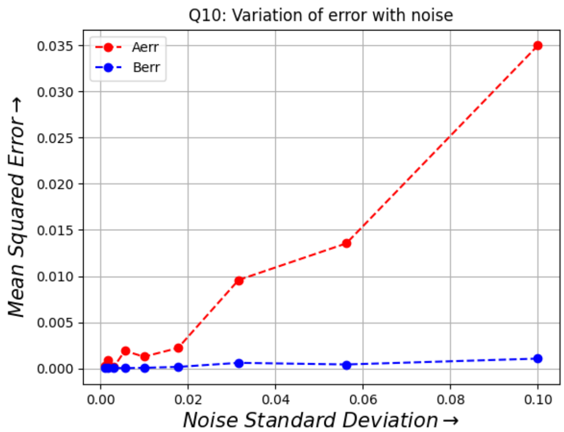

Figure 4: Error Plot

This plot shows the large increase in the error of A as the standard deviation of the noise increases while the error in the estimation of B remains almost constant.

2. The plot of error in approximation with the standard deviation of the noise in the data in logarithmic scale is as follows:

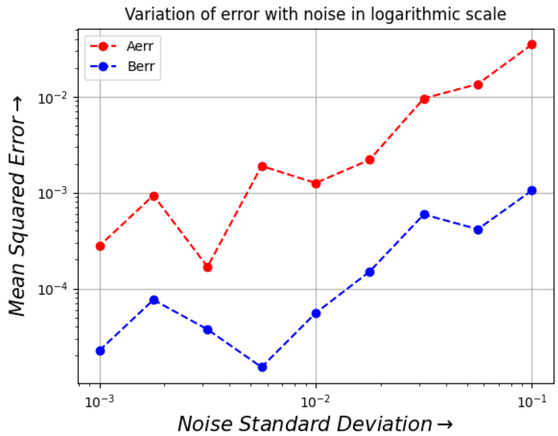

Figure 5: Error Plot in logarithmic scale

Here, both estimations show an approximately linear variation with the standard deviation of noise. The reason why there was very little change in the previous plot was that the magnitude of change of B is small but the exponential change is large.

3. This is the Stem plot to show the logarithmic values of A and B:

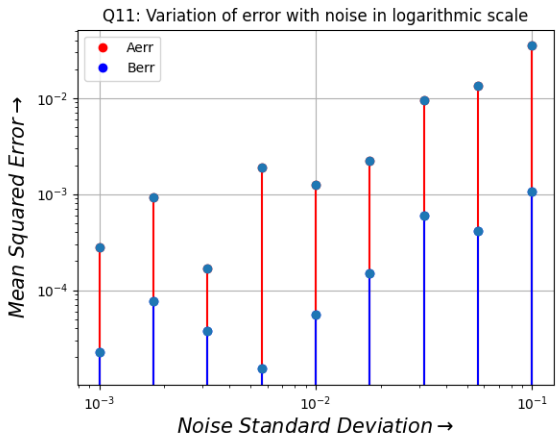

Figure 6: Stem Plot in logarithmic scale

Here, both estimations show an approximately linear variation with the standard deviation of noise. The reason why there was very little change in the previous plot was that the magnitude of change of B is small but the exponential change is large.

Conclusion

In the contour plot, it is seen that the mean squared error of the data converges close to the true value and minimizing it using the least squares method gives a very good estimation with error less than 5% with the standard deviation 0.1.

The error is seen to increase approximately linearly in the logarithmic scale with increase in noise. The value of B parameter changes very slowly compared to value of A in the regular scale but changes at a more rapid rate in the logarithmic scale.