EE2703: Assignment 3

Akilesh Kannan (EE18B122)

February 5, 2020

1 Aim

In this assignment we aim to :

- Observe the error in fitting the *Least Error Fit* function to a given set of data.
- Find the relation between the error observed and the noise in the data.

2 Procedure

The function to be fitted is:

$$f(t) = 1.05J_2(t) - 0.105t$$

where $J_2(t)$ is the *Bessel Function of the first kind of Order 2.* The true data used for fitting is obtained using this equation.

2.1 Creating noisy data

To create the noisy data, we add random noise to $f(t)$. This random noise, denoted by $n(t)$, is given by the standard normal probability distribution:

$$P(n(t)|\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{n(t)^2}{2\sigma^2}} \tag{1}$$

The resulting noisy data will be of the form:

$$f(t) = 1.05J_2(t) - 0.105t + n_{\sigma_i}(t) \tag{2}$$

where, $n_{\sigma_i}(t)$ is the noisy data function with $\sigma = \sigma_i$ in (1). Thus for 9 different values of sigma (in a log scale from 0.001 to 0.1), the noisy data is created and stored in the `fitting.dat` file.

2.2 Analyzing the noisy data

The data is read and plotted using PyPlot. The output result looks as follows:

Figure 1: Noisy Data with True Data

As we can see, the "noisiness" of the data increases with increasing value of σ. Another view of how the noise affects the data can be seen below:

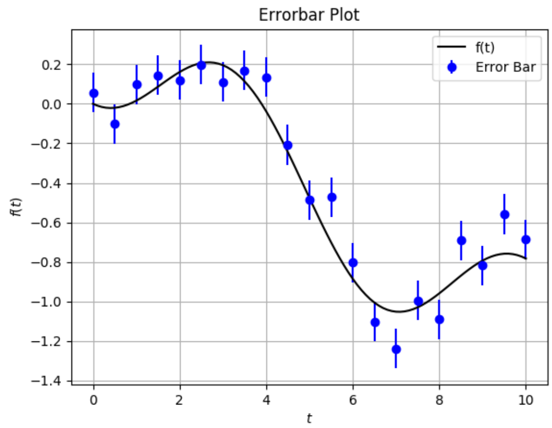

Figure 2: Noisy Data with Errorbar

The blue lines (*error bar*) indicate the standard deviation of the noisy data from the original data, at that value of t. It is plotted at every 5^{th} point to make the plot readable.

2.3 Finding the best approximation for the noisy data

From the data, we can conclude that the data can be fitted into a function of the form:

$$g(t, A, B) = AJ_2(t) + Bt \tag{3}$$

where A and B are constants that we need to find.

To find the coefficients A and B, we first try to find the mean square error between the function and the data for a range of values of A and B, which is given by:

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

where ϵ_{ij} is the error for (A_i, B_j). The contour plot of the error is shown below:

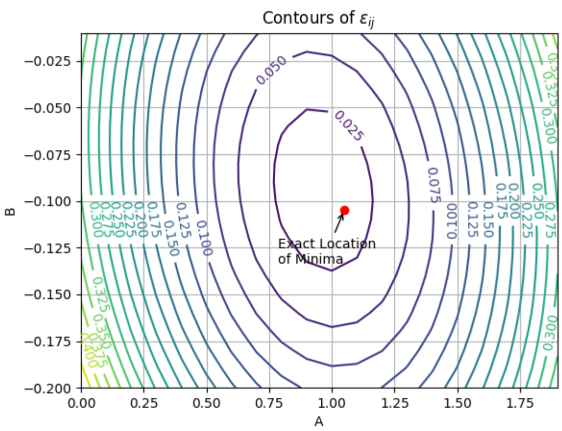

Figure 3: Contour Plot of ϵ_{ij}

We can see the location of the minima to be approximately near the orignal function coefficients.

Using the `lstsq` function in `scipy` package, we solve for:

$$M.p = D \tag{5}$$

where

$$M = \begin{bmatrix} J_2(t_1) & t_1 \\ \dots & \dots \\ J_2(t_m) & t_m \end{bmatrix}, \; p = \begin{bmatrix} A_{fit} \\ B_{fit} \end{bmatrix} \text{ and } D = \begin{bmatrix} f(t_1) \\ \dots \\ f(t_m) \end{bmatrix} \tag{6}$$

Thus, we solve for p and then find the mean square error of the values of A_{fit} and B_{fit} found using `lstsq` and the original values $(1.05, \; -0.105)$.

2.4 Finding out the variation of ϵ with σ_n

We solve (5) for different values of σ_n, by changing matrix D to different columns of `fitting.dat`. We find that the variation of the mean squared error of values A_{fit} and B_{fit} is as follows:

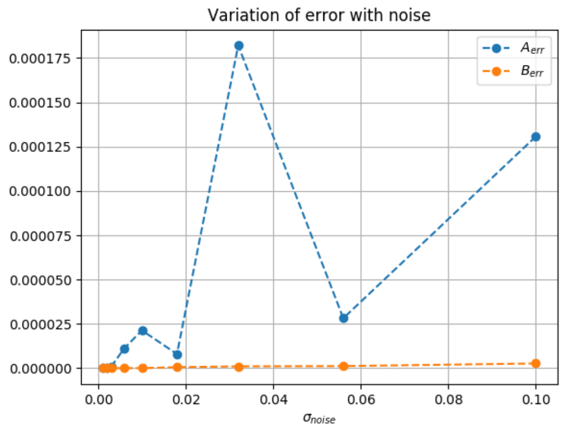

Figure 4: Mean Squared Error vs Standard Deviation

This plot does not give that much useful information between σ_n and ϵ, but when we do the `loglog` plot as below:

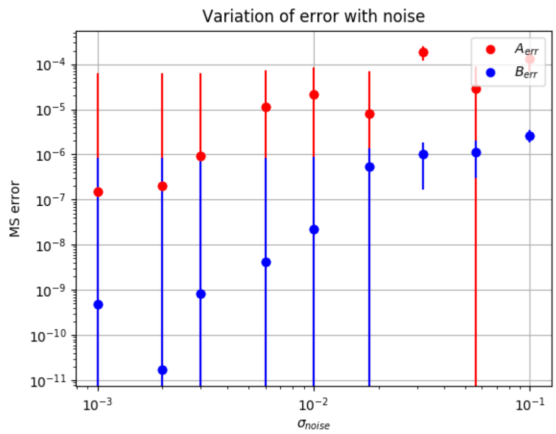

Figure 5: Error vs Standard Deviation `loglog` Plot

We can see an approximately linear relation between σ_n and ϵ. This is the required result.

3 Conclusion

From the above procedure, we were able to determine that **the logarithm of the standard deviation of the noise *linearly affects* the logarithm of the error** in the calcuation of the least error fit for a given data.