EE2703: Assignment 3

Sarthak Vora (EE19B140)

March 03, 2021

1 Abstract

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 Introduction

From linear algebra, we can condense any parameter estimation problem to a simple matrix equation of the form:

$$\begin{pmatrix} F_1(t_1) & F_2(t_1) & ... & F_n(t_1) \\ F_1(t_2) & F_2(t_2) & ... & F_n(t_2) \\ ... & ... & ... & ... \\ F_1(t_m) & F_2(t_m) & ... & F_n(t_m) \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ ... \\ p_n \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ ... \\ a_n \end{pmatrix} \tag{1}$$

where,

$$f(t; p_1, p_2, .., p_n) = \sum_{i=0}^{n} p_i F_i(t) \tag{2}$$

is the function to be estimated.

Since we only have to "fit" the real-time data we can choose a function of our choice by making some educated guesses for these functions on looking at the plots of these real-time data.

Equation (1) can be written as:

$$F.\vec{p} = \vec{a_0} \tag{3}$$

But in a real world situation, there will always be noise associated with the data. To account for that, we need to slightly modify (3) to:

$$F.\vec{p} = \vec{a_0} + \vec{n} = \vec{a} \tag{4}$$

where $\vec{n}$ accounts for the noise in the data.

However, the above equation is not satisfied always (as number of equations is N, and number of measurements is M)

So we make a few assumptions about the noise in our data and then try to guess the best solution, i.e., the error between the ideal solution, in the absence of noise and the one obtained has to be as minimum as possible.

The assumptions that we make about noise are that it has zero mean and a standard deviation of σ.

The final result on ***Least Squares Estimate*** of $\vec{p_0}$ is given by -

$$\vec{p_0} = (F^T F)^{-1} F^T \vec{a} \tag{5}$$

However, the above results hold true only if the initial function $F_i(t)$ is independent and that the noise is same for different measurements. This is because, we have to give *lesser importance (**weight**)* to those values which have more noise, as they are more unreliable.

3 Procedure

The function to be fitted is:

$$f(t) = 1.05 J_2(t) - 0.105t \tag{6}$$

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

3.1 Creating noisy data

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

$$P(n(t)|\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n(t)^2}{2\sigma^2}} \tag{7}$$

The resulting noise data will be of the form:

$$f(t) = 1.05 J_2(t) - 0.105t + n_{\sigma_i}(t) \tag{8}$$

where, $n_{\sigma_i}(t)$ is the noisy data function with $\sigma = \sigma_i$ in (8). Thus for 9 different values of σ (in a log scale from 0.001 to 0.1), the noisy data is created and stored in **fitting.dat** file.

3.2 Analyzing the noisy data

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

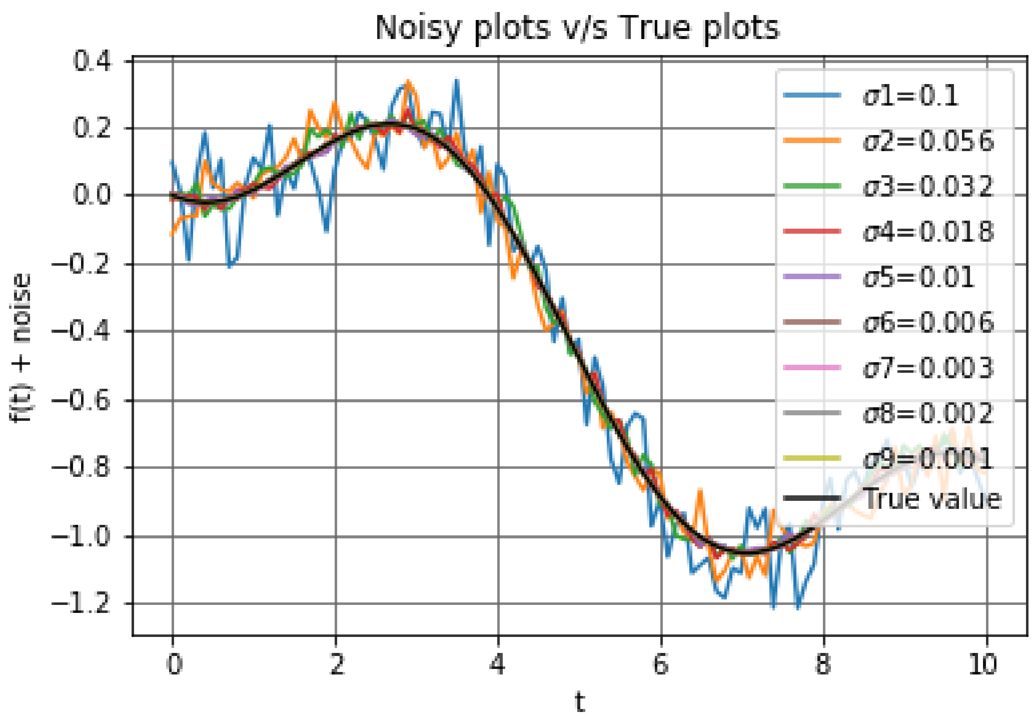

Figure 1. Noisy data with True data

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

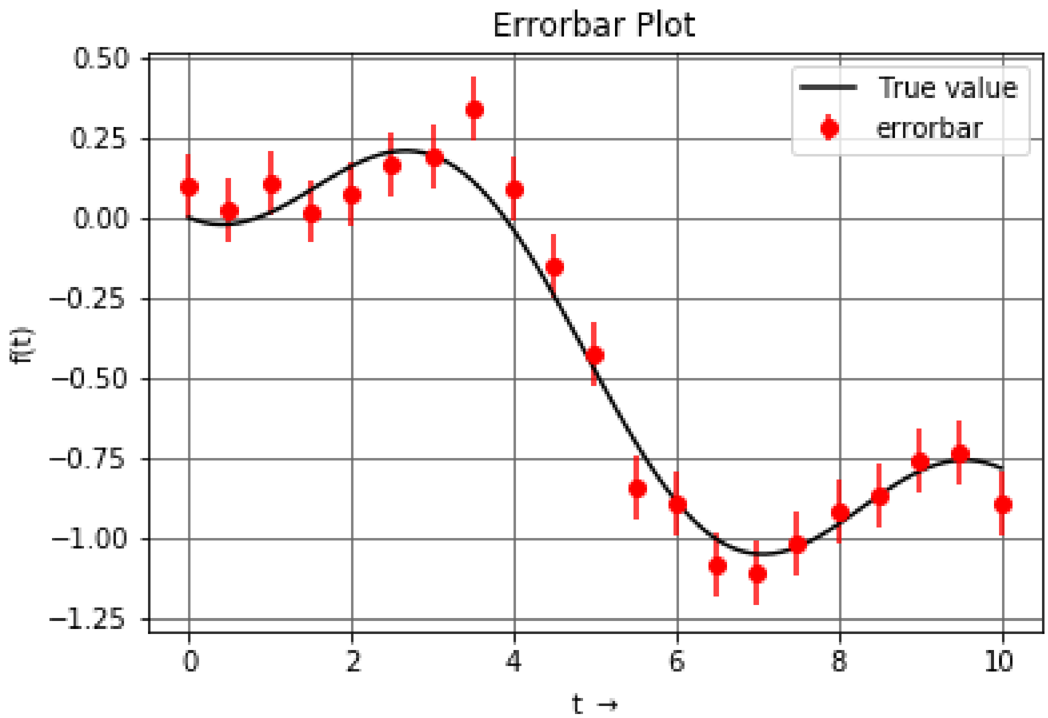

Figure 2. Noisy data with Error bar

The red 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.

3.3 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:

$$g(t, A, B) = AJ_2(t) + Bt \tag{9}$$

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:

$$\epsilon_{ij} = \frac{1}{101}\sum_{k=0}^{101}(f(t_k) - g(t_k, a_i, b_j))^2 \tag{10}$$

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

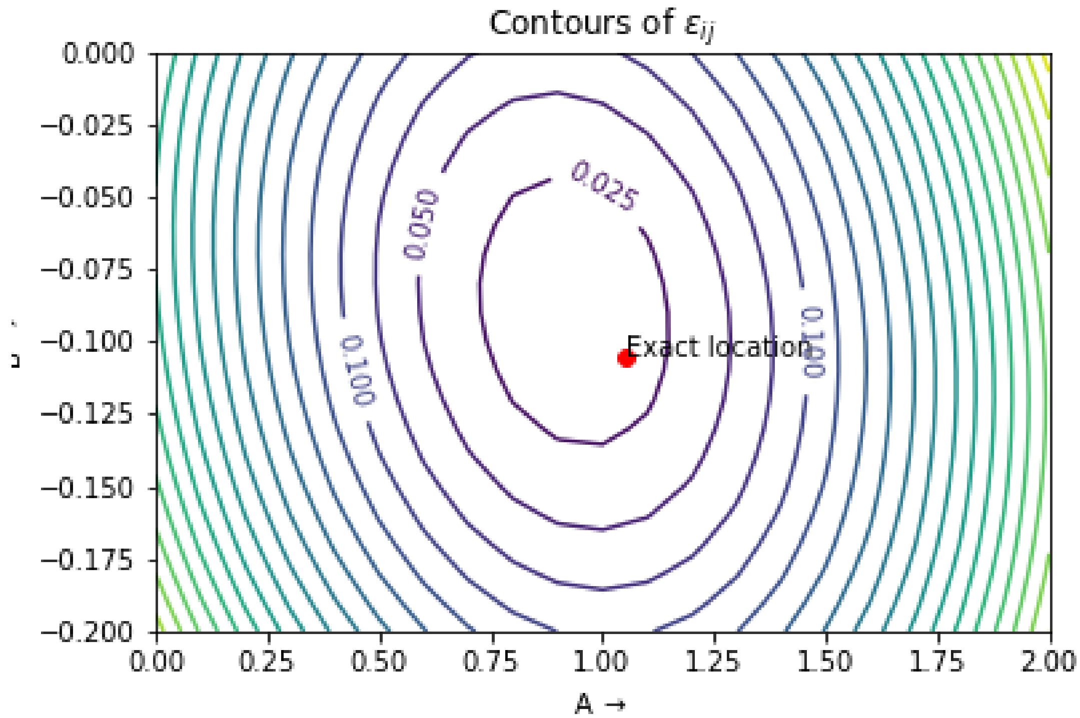

Figure 3. Contour plot of ϵ_{ij}

We can see the location of minima to be approximately near the original function coefficients.
Using the ***lstsq*** package in scipy, we solve the equation:

$$M.p = D \tag{11}$$

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)**.

3.4 Finding out the variation of ϵ with σ_n

We solve (11) 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 the values A_{fit} and B_{fit} is as follows:

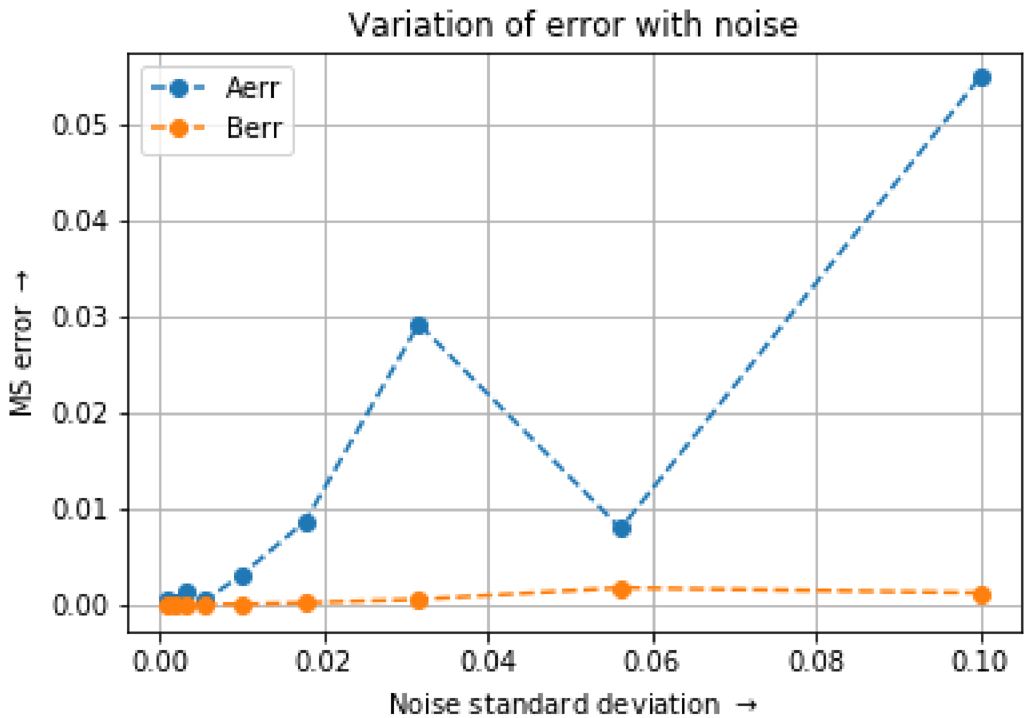

Figure 4. Mean Squared Error v/s Standard deviation

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

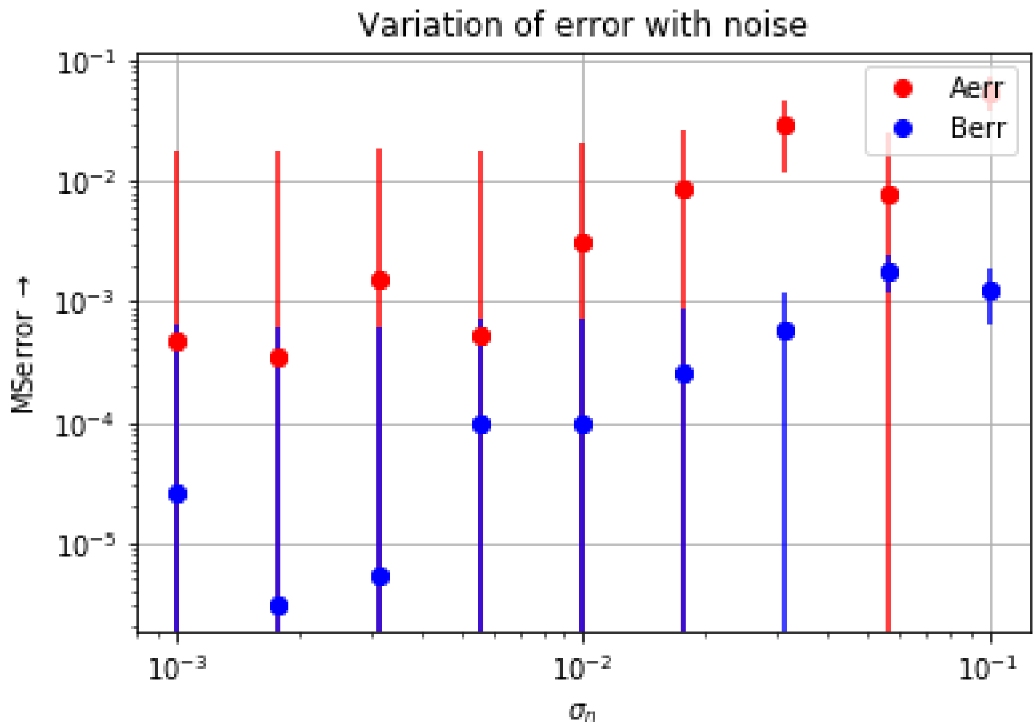

Figure 5. Mean Squared Error v/s Standard deviation *loglog*plot

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

4 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 calculation of the least error fit for a given data.