Assignment-3 : Fitting Data to Models

Sai Gautham Ravipati - EE19B053

March 3, 2021

1 Abstract

This assignment is about using Python to :

- Loading data from files and parsing them.
- Analysing the data to extract information.
- Study the effect of noise on the fitting process.
- Plotting graphs.

2 Generating Data

For the generation of data, the python script 'generate_data.py' is used which generates a set of 9 columns with 101 data points each and varying noise levels

2.1 True Data

The true value of the function without noise is given by :

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

where $J_2(t)$ is the *Bessel Function of first kind and Order 2.*

2.2 Noisy data

The random noise $n(t)$ added is modelled by the standard normal distribution as :

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

The resulting noisy data is of the form :

$$f(t) = 1.05J_2(t) - 0.105t + n_{\sigma_i}(t) \tag{3}$$

where, $n_{\sigma_i}(t)$ is the noise added with standard deviation, $\sigma = \sigma_i$. These values of σ_i are chosen from the set (0.001 to 0.1) on a uniform log scale. This data is stored in the file 'fitting.dat'.

3 Visualizing the Data

The following function in the code snippet does the work of generating the true data-points when time-series is passed in as the input.

```
def g(t,A = 1.05 ,B = -0.105) :
    return A*sp.jn(2,t)+B*t
```

Once the true data-points are generated, the true curve and the rest of the nine curves are plotted on the same plot using 'matplotlib.pyplot' and the curves look as shown below :

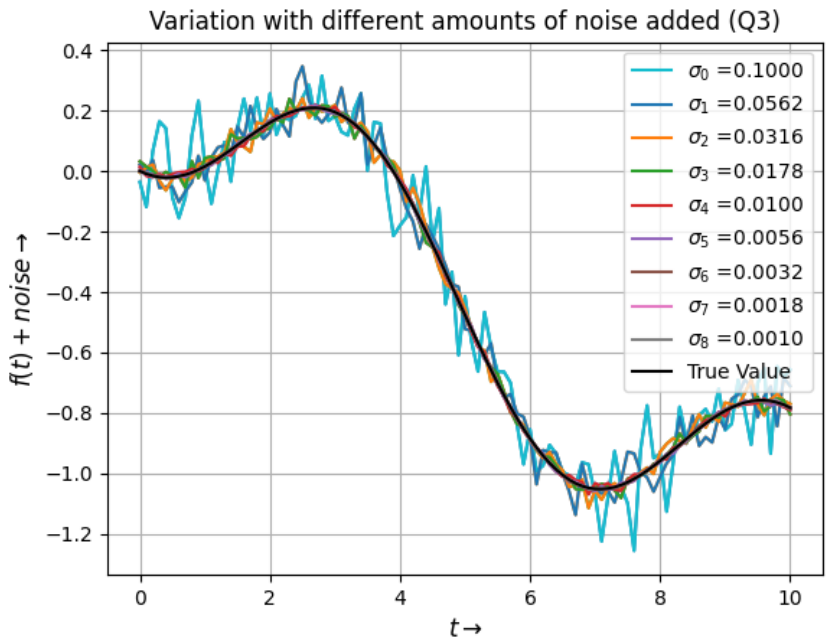

Figure 1: Varying Noise Levels and True Data

Clearly, it can be seen that as the standard deviation of the noise function is increased, the curves are more widely spread.

4 Error Bars

Error bars are obtained using the standard deviation of noisy data from the true data. To visualize how much the noisy data diverges from the true data, both true data and error bars are plotted as shown below :

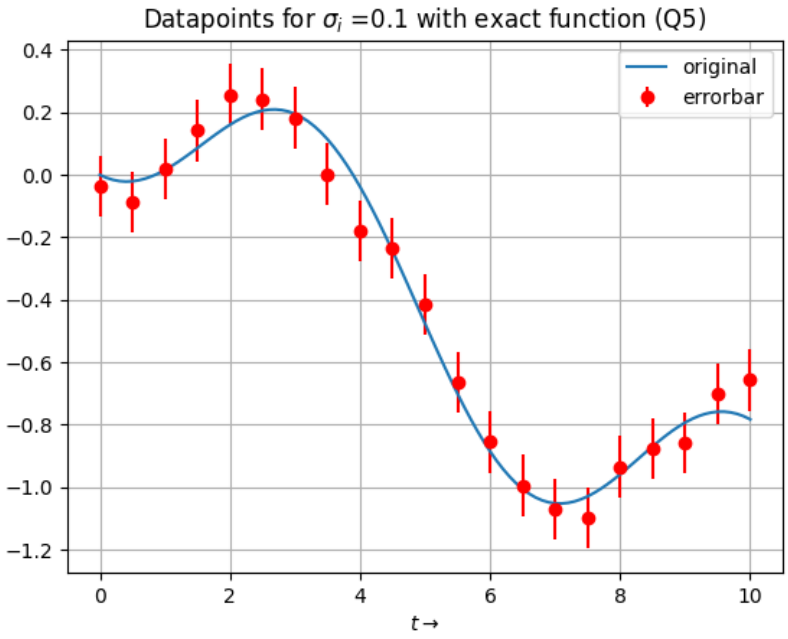

Figure 2: Error Bars for $\sigma_i = 0.1$ on top of True curve

5 Equivalent Matrix representation

The following matrix representation :

$$g(t, A, B) = M.p \tag{4}$$

where, $M.p$ is given by

$$\begin{pmatrix} J_2(t_1) & t_1 \\ \dots & \dots \\ J_2(t_m) & t_m \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} \tag{5}$$

is indeed true and the same can be verified through the following numpy operation from the python script :

```
np.array_equal(f_Mp,f_true)
```

which checks if both the matrices are equal and returns a 'True' value in this case.

6 MS Error and Contour Plots

6.1 Mean Squared Error

The mean squared error is given by :

$$\epsilon_{ij} = \frac{1}{101}\sum_{k=0}^{101}(f(t_k) - g(t_k, A_i, B_j))^2 \tag{6}$$

The following snippet of the code computes the MS error matrix, for all the 9 columns of data, and for A in (0,0.1,...,2) and B in (-0.2,-0.19,...,0).

```
for k in range(9):
    for i in range(21):
        for j in range(21):
            eps[i][j][k] = np.sum(np.square(f[:,k] - g(t,A[i],B[j])))/101
```

6.2 Contour Plots

After the MS error is calculated as shown previously, for the first column of data loaded from file, contour plot is plotted varying A,B to find constant levels of ϵ_{ij}. The curves look as shown below :

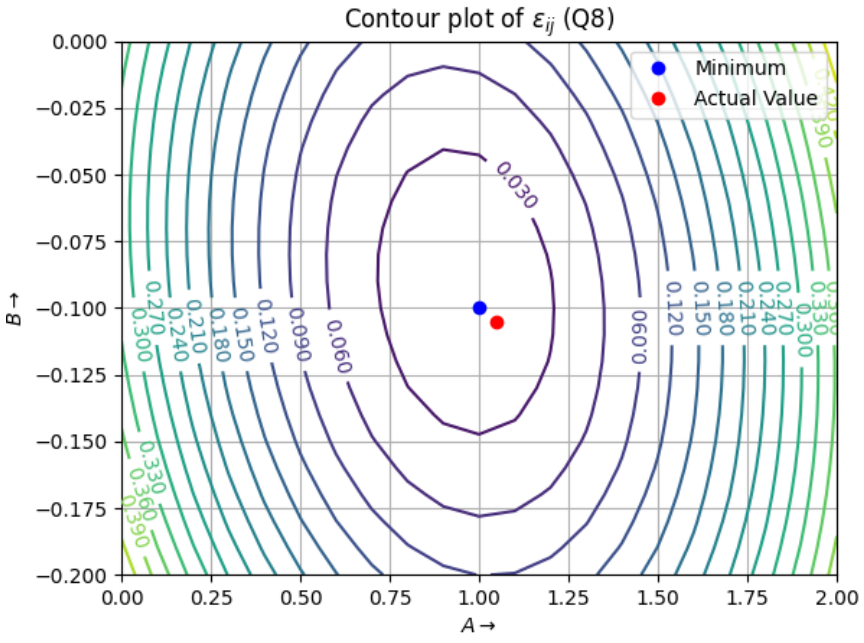

Figure 3: Contour plot for first column of data loaded

Here in this case, the contour plot has one minimum at (1,-0.099) which is close to (1.05,-0.105) with which the noise free data was generated. The following numpy operation is used to find the minima in the plot :

```
np.unravel_index(np.argmin(eps[:,:,0], axis=None), eps[:,:,0].shape)
```

7 Least Square Solution

The scipy operation 'lstsq' returns the least square solution to linear matrix equation. Here in this case the equation is :

$$M.p = g(t, 1.05, -0.105) \tag{7}$$

The MS error of predicted values of A,B with respect to (1.05,-0.105) looks as shown below :

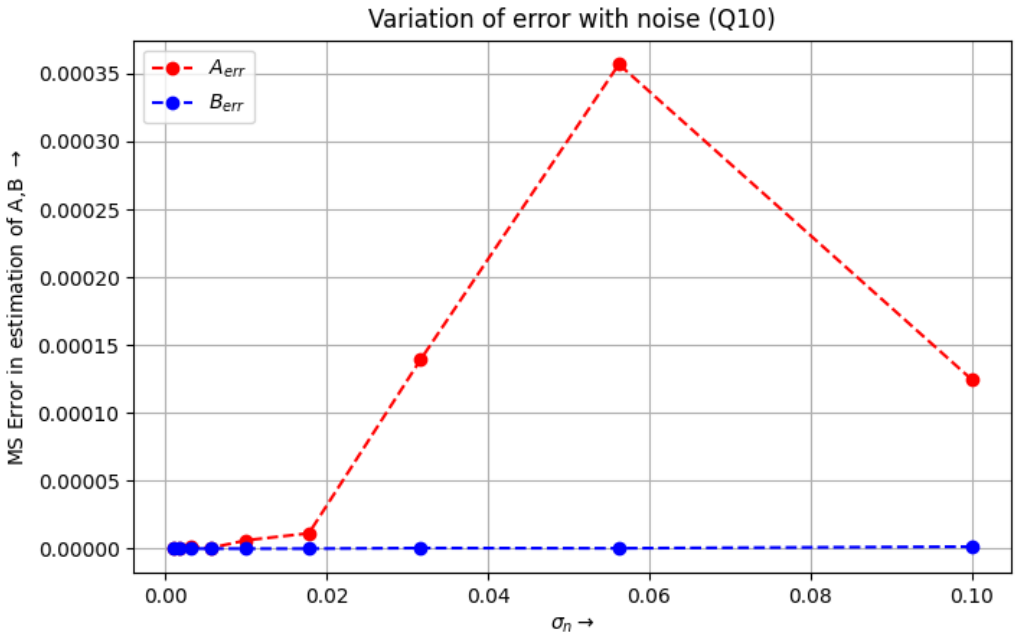

Figure 4: MS Error in Linear scale

Clearly, the plot doesn't appear to be linear. When the same plot is taken in a log-log scale the plot appears to be more linear when compared to the previous linear plot. This may be due to the fact that noise is standard normal with standard deviation σ_i, thus making the graph comparatively linear in log scale when compared to linear scale. Further in this case σ_i are sampled on a uniform log scale. The log-log plot looks as ahown below :

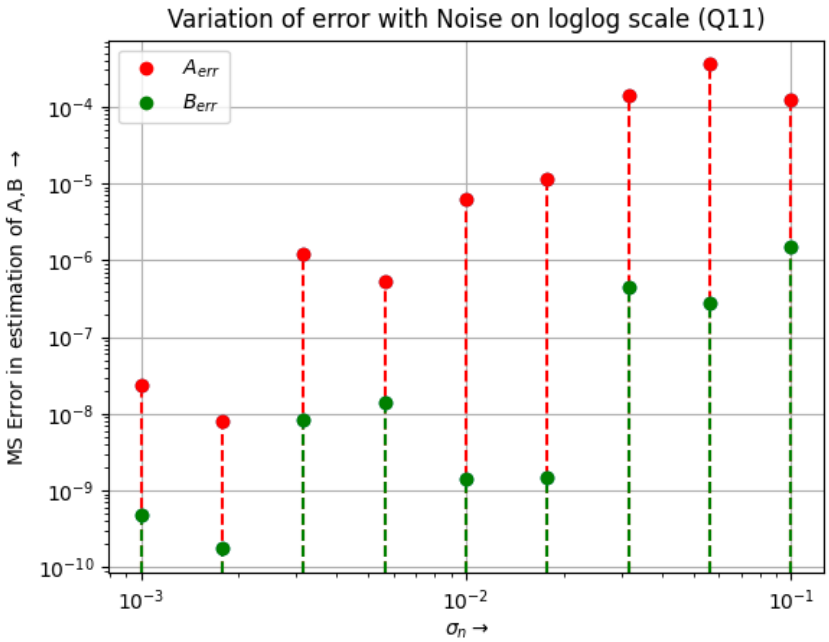

Figure 5: MS Error in Log-Log scale

8 Conclusion

- Error for various noise levels varies linarly in a log-log scale and is non-linear in a linear scale.
- The contour of MS error has a minimum, and in most cases the minimum and the true location are closeby.
- As the noise level is increased the spread in the noisy curve when compared to the ground truth curve is also increased.