MACHINE LEARNING

FIRST SET OF PROBLEMS

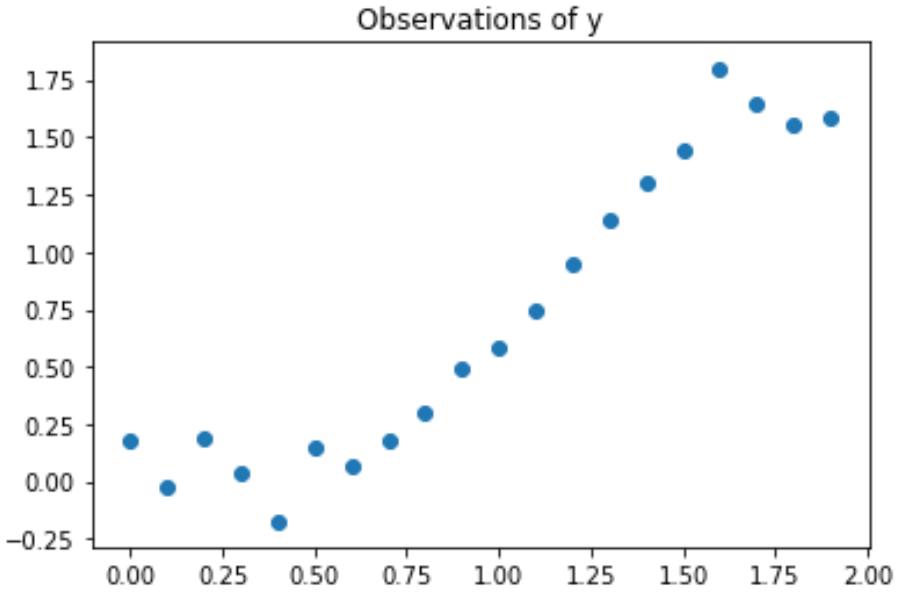
Problem 1

The first problem we have to face is the generalized regression problem defined by the following model:

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_5 x^5 + \eta \qquad (1)$$

where n corresponds to white Gaussian noise. In every case below we consider N equidistant points $x_1, x_2, \ldots, x_n$

1) At first we have N=20 observations in order to train our problem and σ^2_n=0.1. We apply the Least Squares method using the structure of the correct model (5th degree polynomial with the coefficient of the 4th power equal to zero) in order to estimate the parameter vector θ. We plot our observations with Gaussian noise and we get:

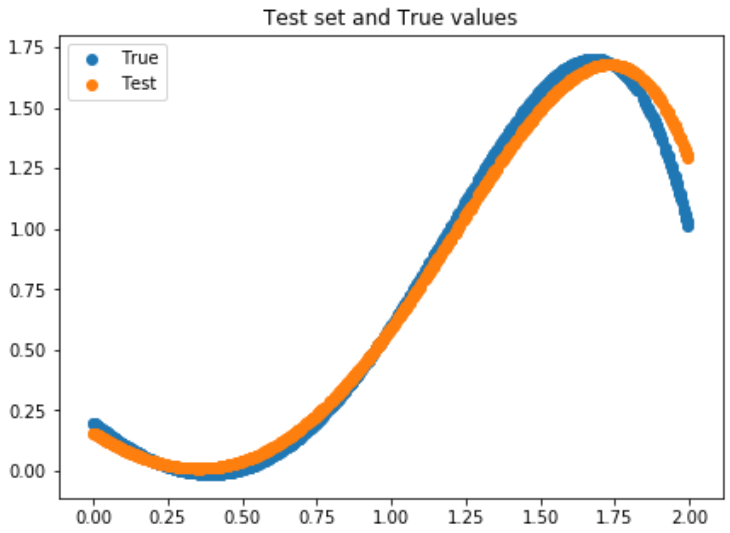


Now we build the Vandermonde matrix (Φ) which is a matrix with the terms of a geometric progression in each row and in this case its dimensions are 5x5. In order to find the parameter vector θ:

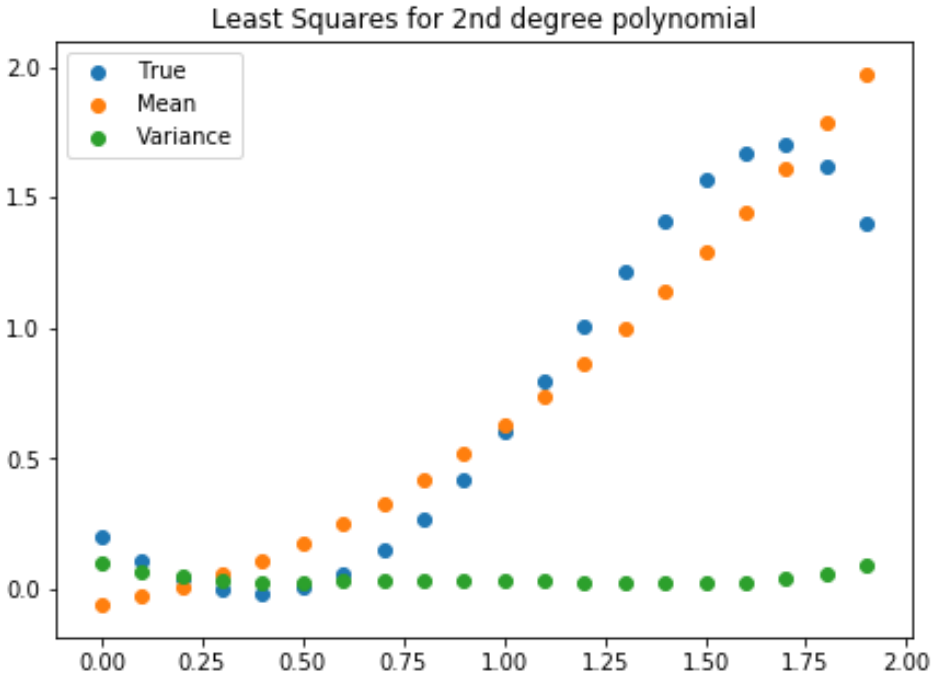
$$\theta_{ls} = (\Phi^T \Phi)^{-1} \Phi^T y$$

After calculating the parameter vector we apply it in our test set which consists of 1000 test points randomly selected in the interval [0,2] and we calculate the Mean Square Error of y (where y are the values of (1) without noise) over the training set

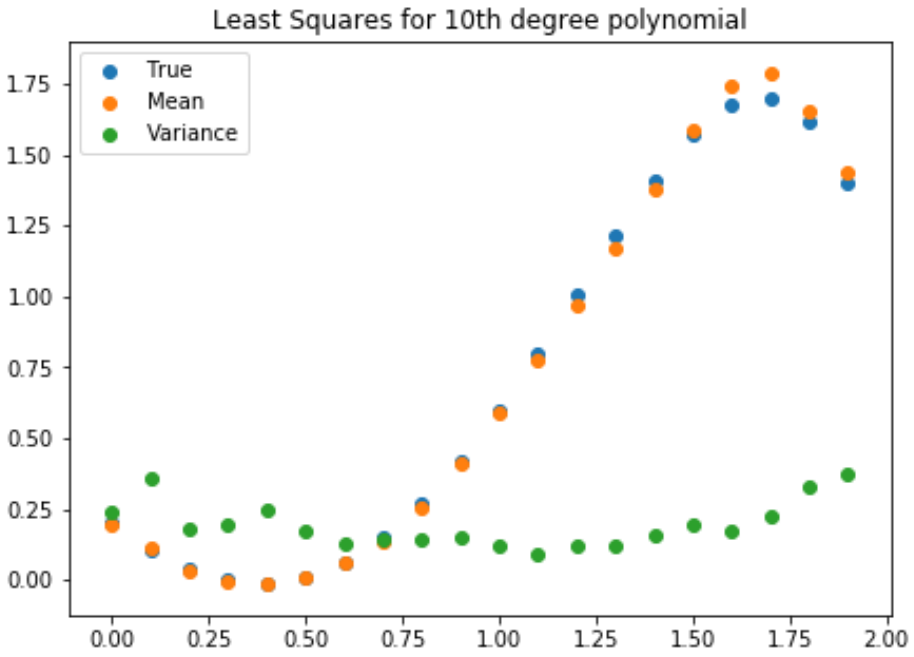
and over the test set. So the Mean square error of y over the training set is MSEtrain=0.00201 and the Mean square error over the test set is MSEtest=0.00226. As we can see both MSE are very small meaning that least square model fits the data almost perfectly. In order to understand this a little better we plot the test set points with the true values of (1):



2) Now we use again Least Square method but with a 2nd degree polynomial so our vandermonde matrix has dimensions 3x3. We do 100 experiments changing the noise term. Noise samples originating from a Gaussian distribution with mean 0 and variance 0.1*j where j is the number of the experiment. For each point of the training set we calculate the mean and variance of y over the 100 experiments and plot these quantities along with the curve obtained by the true model.



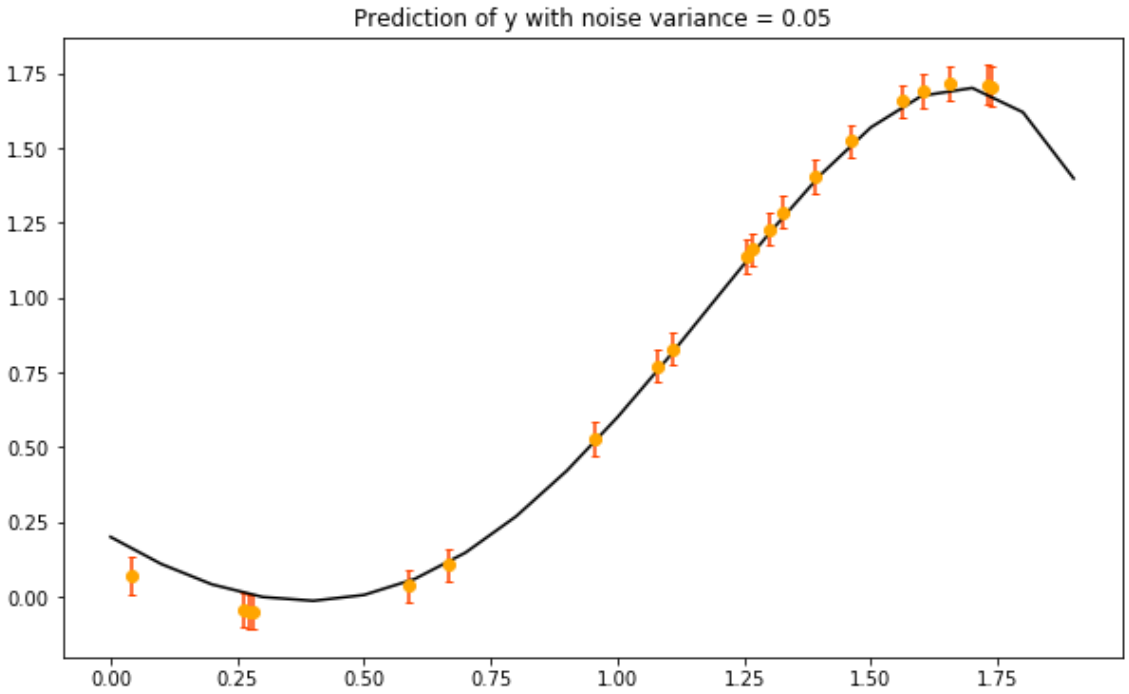
We repeat the same method in order to find the mean and variance of y over the 100 experiments with the same noise distribution and values and we plot the results.

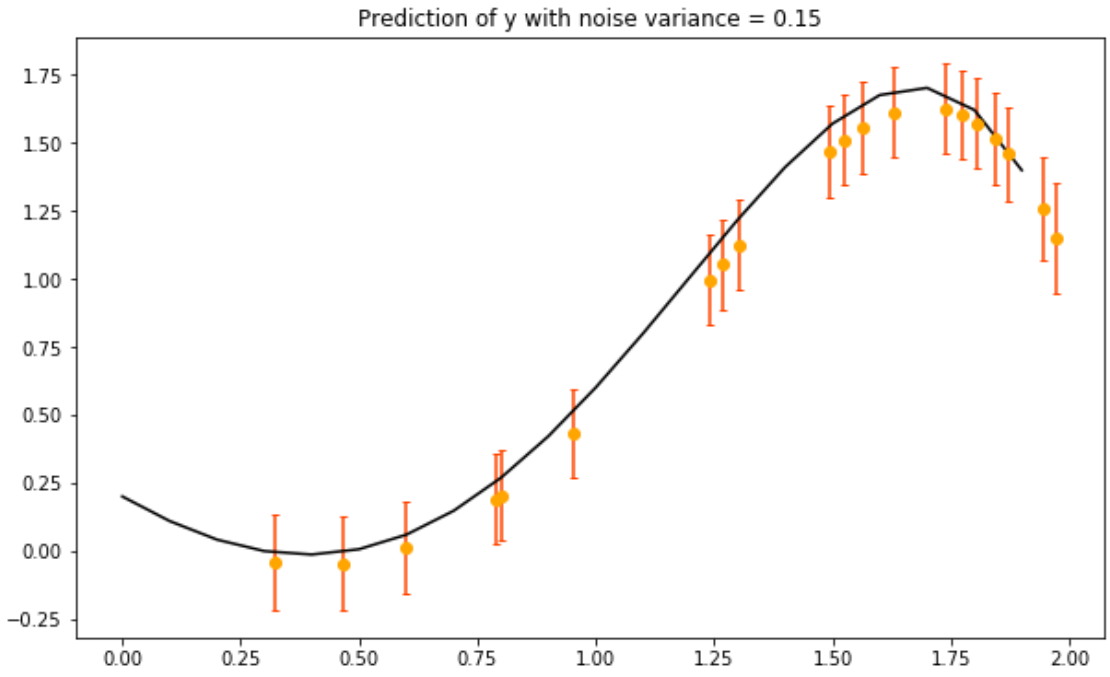


Using the 2nd degree polynomial for the Least Square regression method we have a polynomial with high bias and we are pretty sure that our model will not overfit. However the model is pretty simple and it underfits the training data as we can see from the 1st plot and the variance is close to 0. Using the 10th degree polynomial we have higher variance and lower bias than the previous example. This polynomial is more complex but as we can see from the plot it fits the data perfectly and its variance is close to 0.25. To sum up the lower variance 2nd degree problem produces lower variance predictions that don't fit the true values of (1) (as expected) because the polynomial degree is lower than the degree of the true model (2<5). On the other hand 10th degree polynomial has higher variance than the previous polynomial estimation but fits almost perfectly the true model dataset.

3) In this step we repeat experiment (1) implementing the Ridge Regression method with λ varying from 0 to 2 with step 0.01. We use again the Vandermonde matrix, we just add a bias term that will differ this method from the Least Square method. We decide to use values closer to 0 because we observred that while the λ increases the MSE also increases (in general but there are exceptions). There are several values that give better MSE than the MSE from the least square error. For example for λ=0.06 and for the training set MSE=0.00150 and for the test set MSE_{ridge}=0.00189. As we know a bias is always better from an unbiased one in certain values.

4) We encode our prior knowledge for the unknown parameter vector via a Gaussian distribution G(θ) with mean θ_0 equal to the true parameter vector in equation (1) and covariance matrix $\Sigma_\theta = \sigma_\theta^2 I,\ \sigma_\theta^2 = 0.1$. Also we use the structure of the true model and perform Bayesian Inference in order to estimate y for 20 randomly selected test points belonging [0,2]. We repeat the experiment twice for different values of noise variance. (0.05 and 0.15). We expect the values and the error of y for σ^2_n=0.05 be smaller from the second one due to the existence of smaller noise.



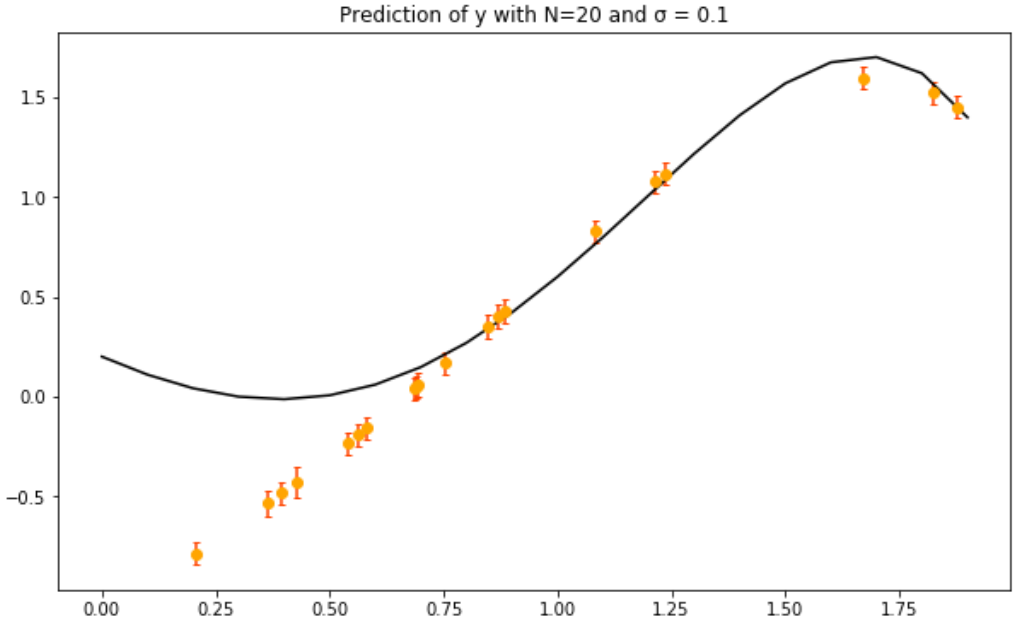


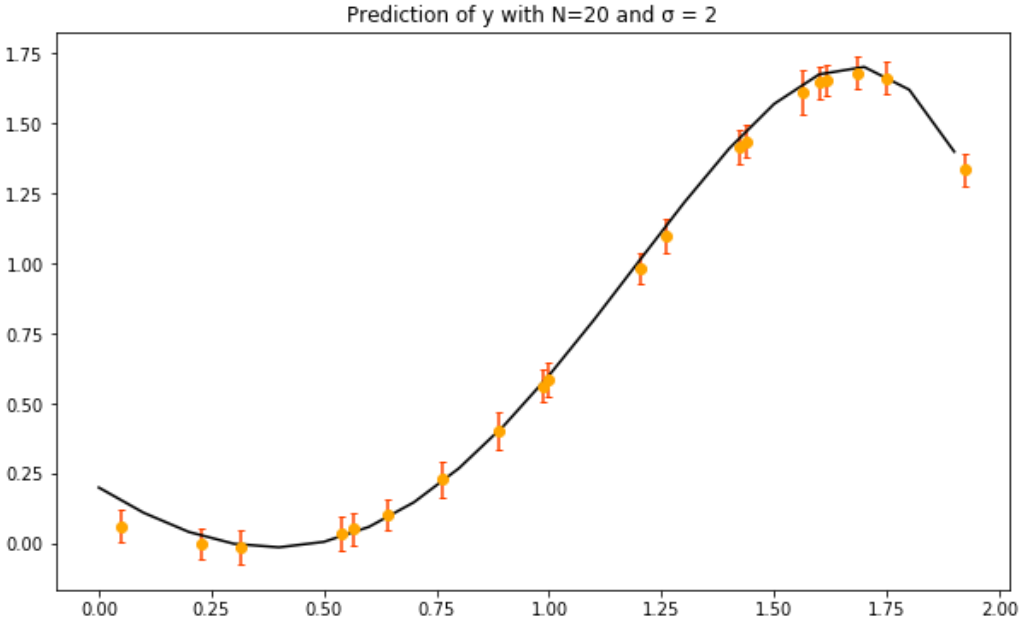
The prediction of y for σ^2_n=0.05 are closer to the true values (as expected) and the error bars are also smaller. Both estimations can be considered correct due to the fact that in both plots $y_{predict} \pm \Delta y = y_{true}$.

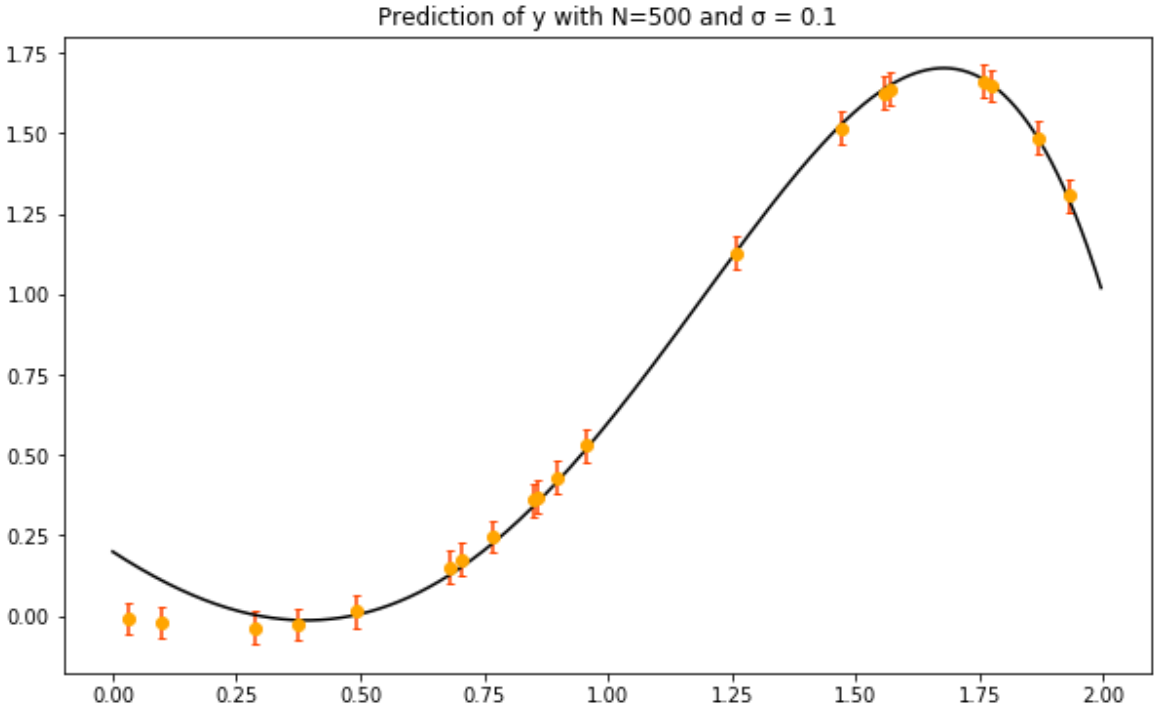
5) We repeat experiment (4) changing the mean vector for G(θ):

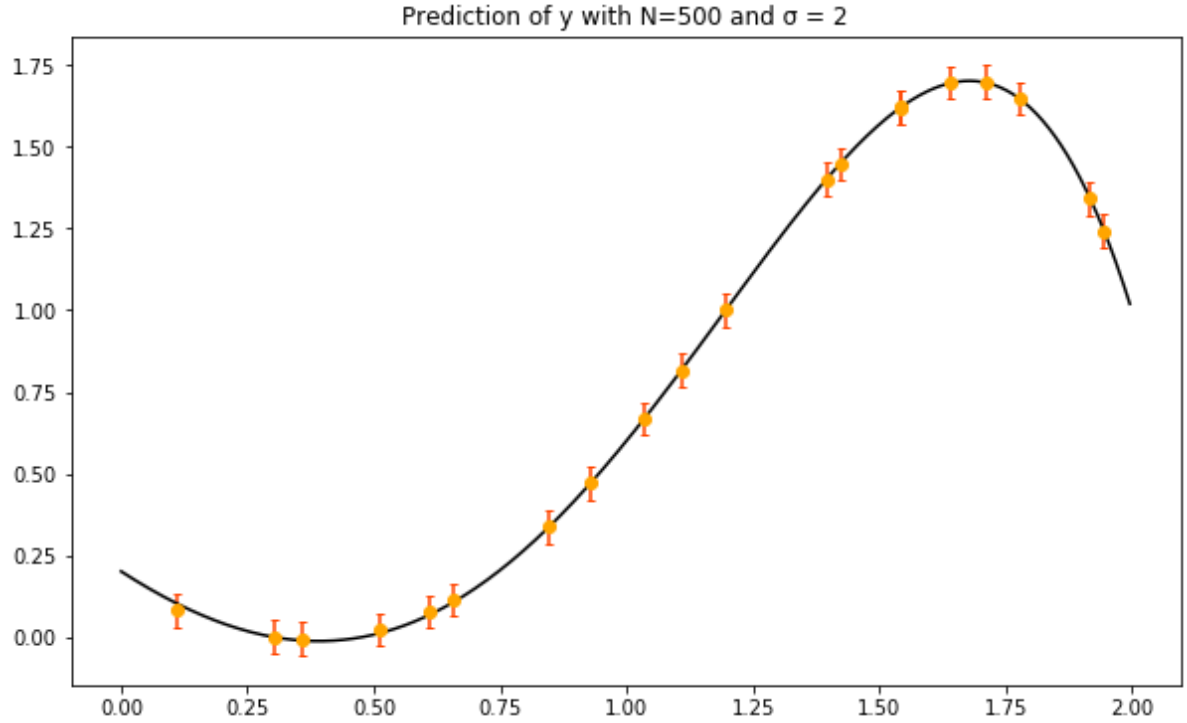
$$\boldsymbol{\theta}_0 = [-10.54, 0.465, 0.0087, -0.093, -0.004]^T$$

We also repeat the experiments 4 times with different values of σ^2_θ (0.1 and 2) and N (20 and 500). We plot the results for these 4 different combinations:



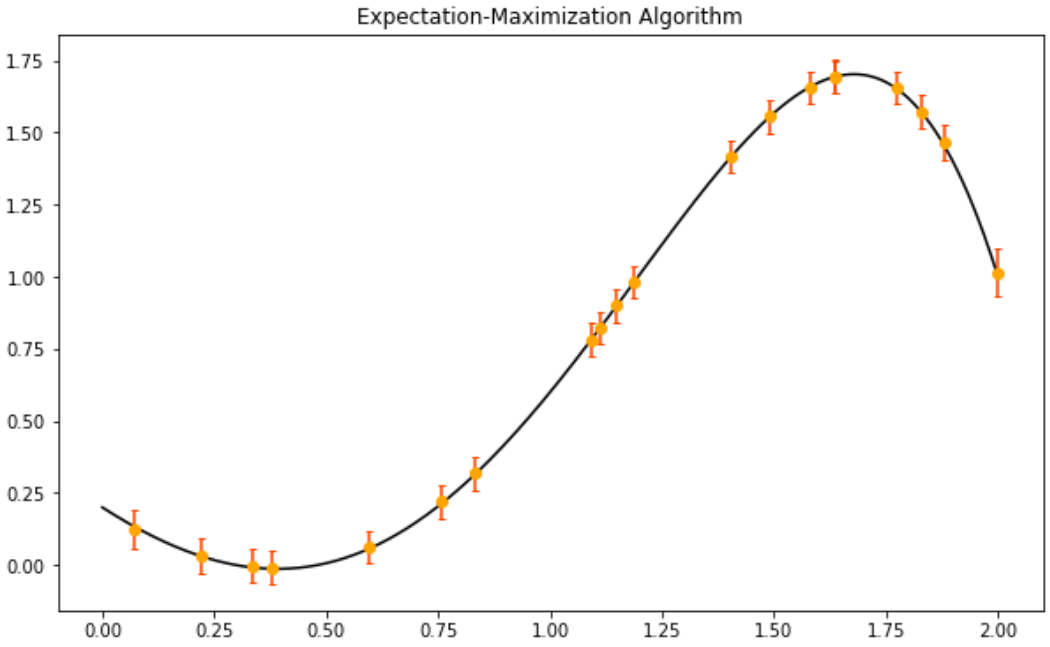






In this set of figures we observe that using larger variance for the prior we get better results than using a small variance (as expected) and having a larger training data set the results for the parameter vector is very good. The best estimation of y happens when the variance for the prior and also the number of points in the training data set get high values.

6) Last but not least for this exercise we construct a training set with N=500 and σ^2_n=0.05. We use the Expectation-Maximization method and we initialize the algorithm with $\alpha=\sigma^{-2}_\theta=1$, $\beta=\sigma^{-2}_\eta=1$. After the convergence of the algorithm we estimate the y's and their errors over a test set of 20 points randomly selected in the interval [0,2]. Below we plot these quantities on the plane, along with the true model curve.



Due to the large number of training points we get a good estimation of y's that follows the distribution of our true model.