I Introduction

Gaussian process is a very important tool in machine learning. It is used to simulate the occurrence of some constantly coming data. However, data may be lost for some interval or we want to know what the exact amount should be between two neighboring data. At this time, Gaussian process can give us a pretty good approximation with lower and upper bound in a relatively smaller interval given by confidence interval. Compared to classical confidence interval method given in statistics, usually the confidence interval does not change with the change of the time and it has a relatively weak prediction when outliers coming. Gaussian process, however, can let us choose which part of the data we concentrate on and we can see the changes over time. In this way, we can get a rough idea on what the missing data should be.

A good usage of this Gaussian process is tracing the movement of a part of a body as it has to be continuous. In this report, let us discuss about how Gaussian process helps tracing it.

II Methodology

We need to first computer the kernel function, we use the square exponential which is specified as following

$$k(x, x') = \text{expl(sigman)l} + (sigmaf) * \exp(-1/2(\exp(sigmal) * (x - x')(x - x'))$$

The formula above gives us a kernel matrix K. With this kernel matrix, we can find three hyper parameters need to specify the value: sigmaf, sigmal and sigman. As the kernel plays an important role in determining logarithm of marginal probabilities, which is specified as follows:

Where t is the target value

$$\log P(t|x, sigma) = -\frac{1}{2} t^T (K + sigman^2 * I)^{-1} t - \frac{1}{2} \log|K + sigman^2 * I| - constant(①)$$

Take $\mathrm{Q} = K + sigman^2 * I$

We can apply gradient descent to the equation above to determine the best value that maximize the equation. In general, for sigman, sigmal and sigmaf, we have

$$\frac{d}{dsigma}\log(P|x, sigma) = \frac{1}{2}t^T Q^{-1}\frac{dQ}{dsigma}Q^{-1}t - \frac{1}{2}trace(Q^{-1}\frac{dQ}{dsigma})(②)$$

For $\frac{dQ}{dsigma}$ appeared in ②, we need to specify for different sigmas. That is,

$$\frac{dK}{dsigmaf} = \exp(sigmaf)\exp(-\frac{1}{2}\exp(sigmal)\,|x - x'|^2)(③)$$

$$\frac{dK}{dsigmal} = \exp(sigmaf)\exp(-\frac{1}{2}\exp(sigmal)\,|x - x'|^2)(-\frac{1}{2}\exp(sigmal)\,|x - x'|^2)(④)$$

$$\frac{dK}{dsigman} = \exp(sigman)\,I(⑤)$$

We plug ③④⑤ in to get the optimized parameters.

With the optimized parameters, For a given training set (X,t) and the test data x*, we compute three other matrix

K = k(X,X) + sigman*I

k* = k(X,x*)

k** = k(x*, x*)

Then get the mean vector m = $k^{*T}K^{-1}t$

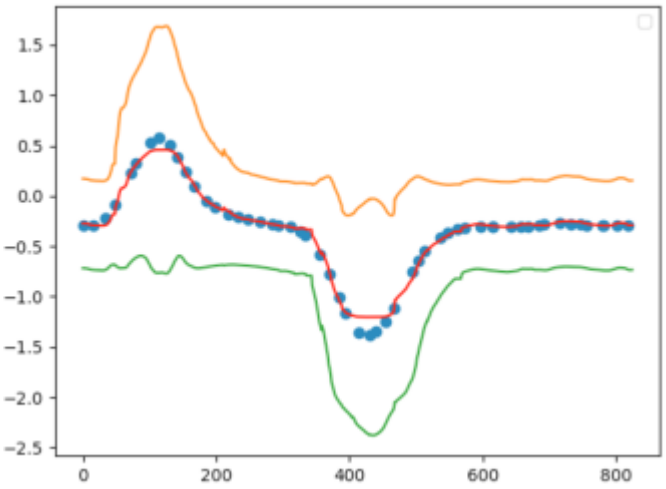
The covariance matrix is Cov = k**- $k^{*T}K^{-1}k^*$

As variance is a special case of covariance, we take the diagonal element in the covariance matrix and take square root of it and times 2.576 to be the upper rising and lower decreasing amount. In this way, we create the confidence interval for approximation.

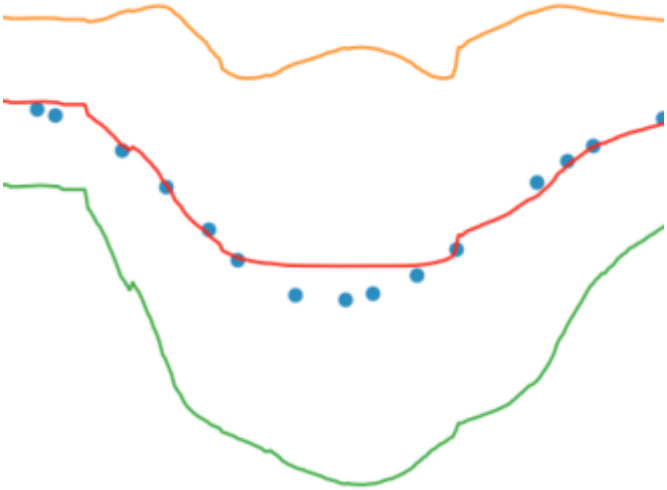
III Experiments and Result

The Gaussian process has pretty good fittings. For the first trial, we pick one data every 10 points and pick 102 points as our samples. Within i^{th} interval,

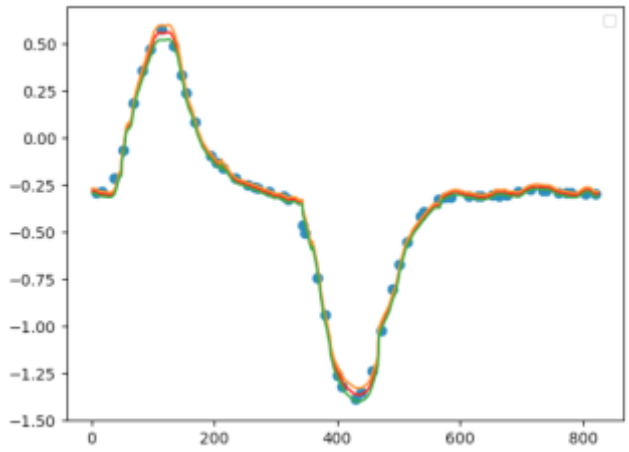
we randomly pick a number m between 0~9. In this way, we pick i*102 + m, which guarantees some randomness.
For a single input and target value, we get the following as initial guess as [1,1,0.1], that is sigmaf to be1, sigmal to be 1 and sigman to be 0.1, the graph looks like the picture below.



When we zoom in the part around frame 400, it looks like the graph below.

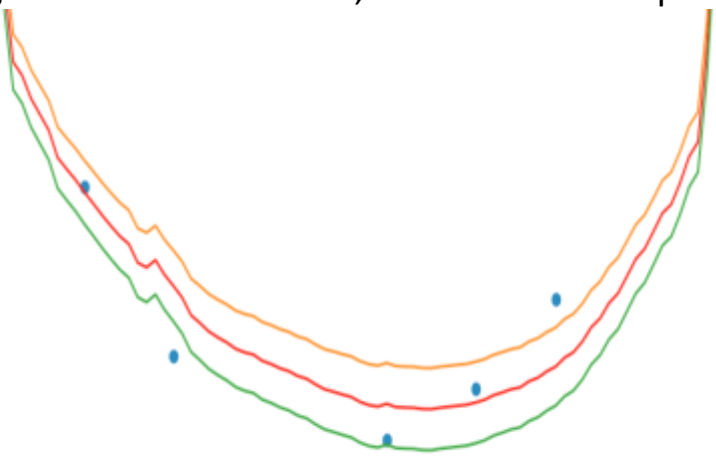


The data below reflects how it looks after running gradient descent. As we can see that the data fits in tighter than the initial value. The lower and upper bound with less difference and the data curve around 400 frames is smoother.

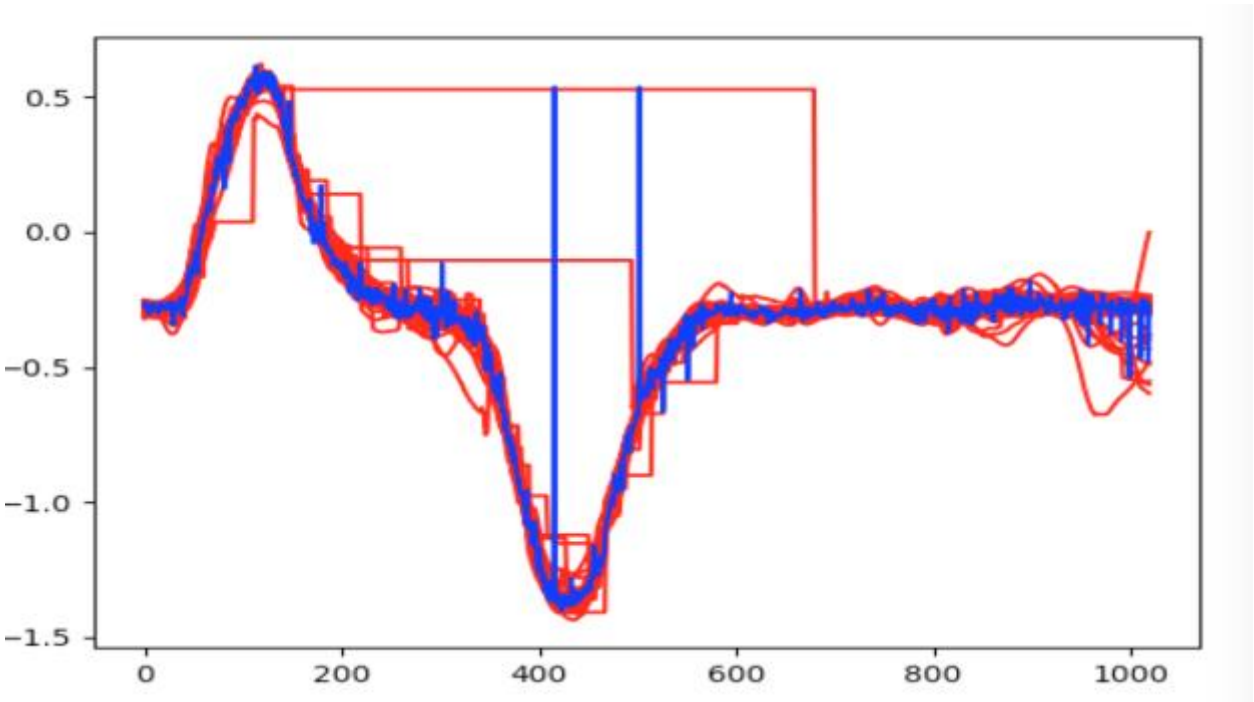


.

When zooming in the same part, we can find pretty good fit as described below: It traces the curve much smoother than our initial guess. Also, it gives a smaller interval, which makes the prediction more meaningful.



Also, we can combine multiple pictures together. That is, pick data not only from one source but 60 sources and for a specific frame, we pick one data from all 60 possible inputs. As everyone is doing the same thing, picking random point can greatly reduce the accidental error coming from a single point can have on our prediction.

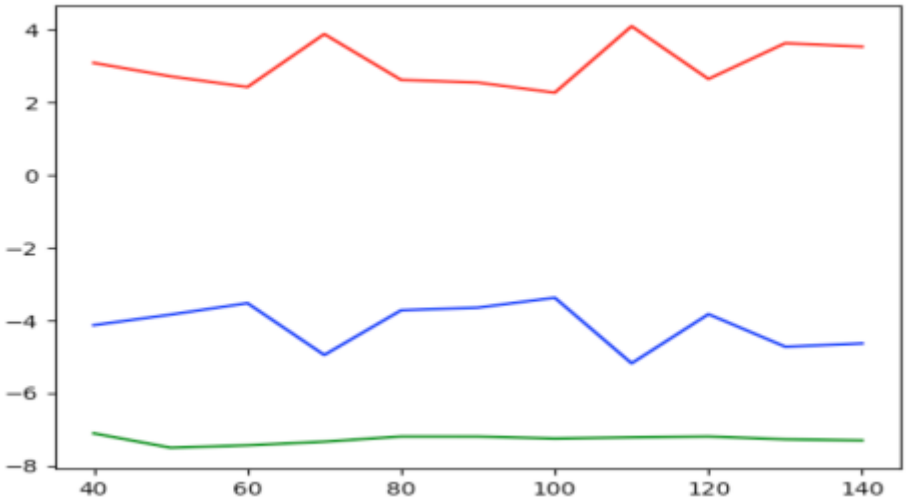


Discussion on Parameters

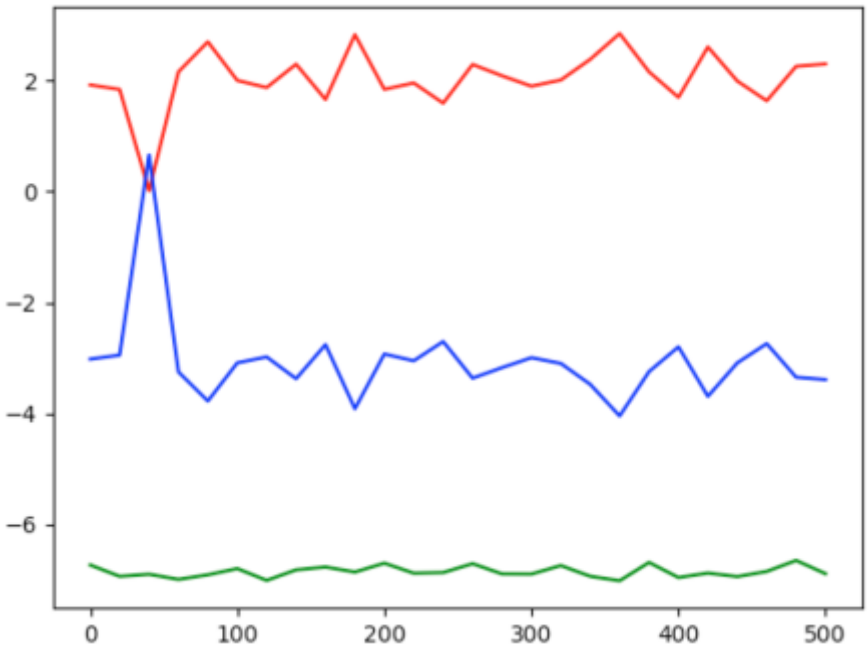
For the case of the graph shown above, we get

[sigmaf, sigmal, sigman] = [3.43923495 -4.53007613 -7.20916514]

Let us try different number of points and see if the parameters will change accordingly. In the graph below, the xaxis is the gap between two points and the y axis reflects the hyper parameter value. From top down, red line reflects the change of sigmaf, blue line reflects the change of sigmal and the green line reflects the change of sigman. As we can see, they all perturbed within a relatively small interval, meaning that the number of points we pick does not have much effects on the changing of hyper parameters.

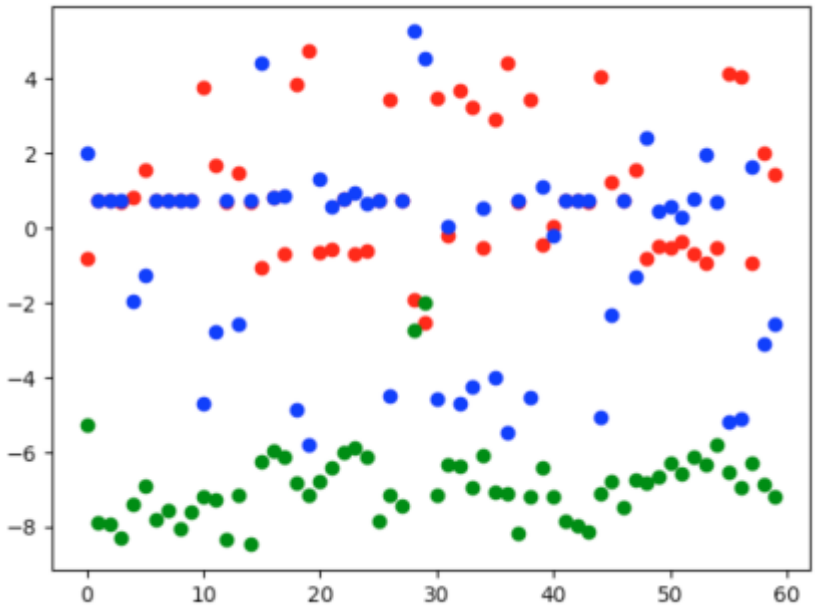


Now consider the change of interval we are interested in. For example, we take 50 points from length 500 interval with every 10 points taking 1 point. The graph's color is same with the label above.
Sigmaf—red Sigmal—blue Sigman—green



The data is not that stable this time. As we can see that sigmal with overall stays around -3 but can reach positive occasionally. The sigman is around [1.5,2.5]. This is due to the shape of the graph changing rapidly at one point but the other intervals are rather stable. Let me take further examples to

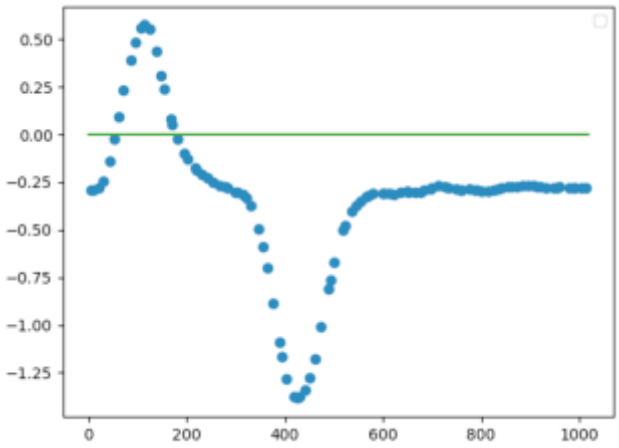
illustrate it. For example when we consider following three time frame [600, 700],[700,800] the hyper parameters are given below:
[-1.43756072 0.85739487 -6.87414643],
[-1.39945084 0.82246638 -6.60676818],correspondingly.
We can see that these two tuples are approximately similar to each other, which indicating that the occurrence of place of the pattern does not have much effects on the parameters.
Consider another pair of fram intervals:[100,200] and [600,700], at [100,200] by observation, the curve is going up. At [600, 700], the curve is relatively flat. When we fit them respectively with the hyper-parameters, we get [0.64279412 -1.6600737 -7.18676927] and[-0.33884248 -0.68175075 -7.27127873] correspondingly. Also, these parameters are different from parameters for whole data set, this also shows that the parameters will change with the changing of the data we take into consideration.
So we find out the hyper-parameters will vary across the data traces but need to under the condition that the data pattern is different. If we have similar patterns, the hyper parameters are similar.
Also, we can see different parameters for all traces for whole data set. In the data set given, we have 60 set of all tracing the xaxis of finger. In the graph: Sigmaf—red Sigmal—blue Sigman—green



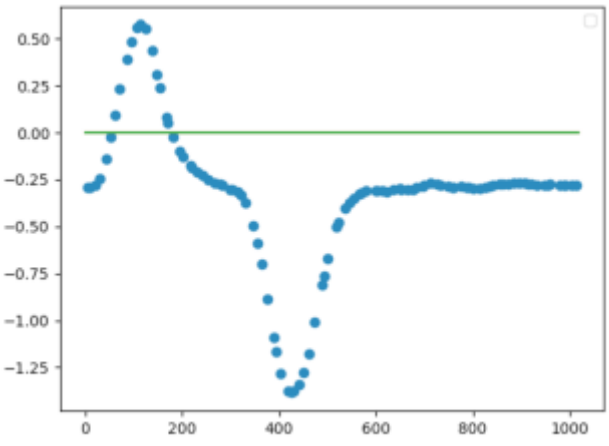
From the graph above, we can say that though everyone tracing the same object, they have different optimized hyper parameters. For sigman, it is usually the most stable one and it does not have much variance between object and object as we can see most of them lies around the interval [-6,-8]. The sigmaf and sigmal are rather unstable. They don't have a center for the distribution.

Finally, the initial guess is really important. If our initial guess is too far away from the optimal solution, we may not get the value we want. Let me illustrate every parameters respectively.

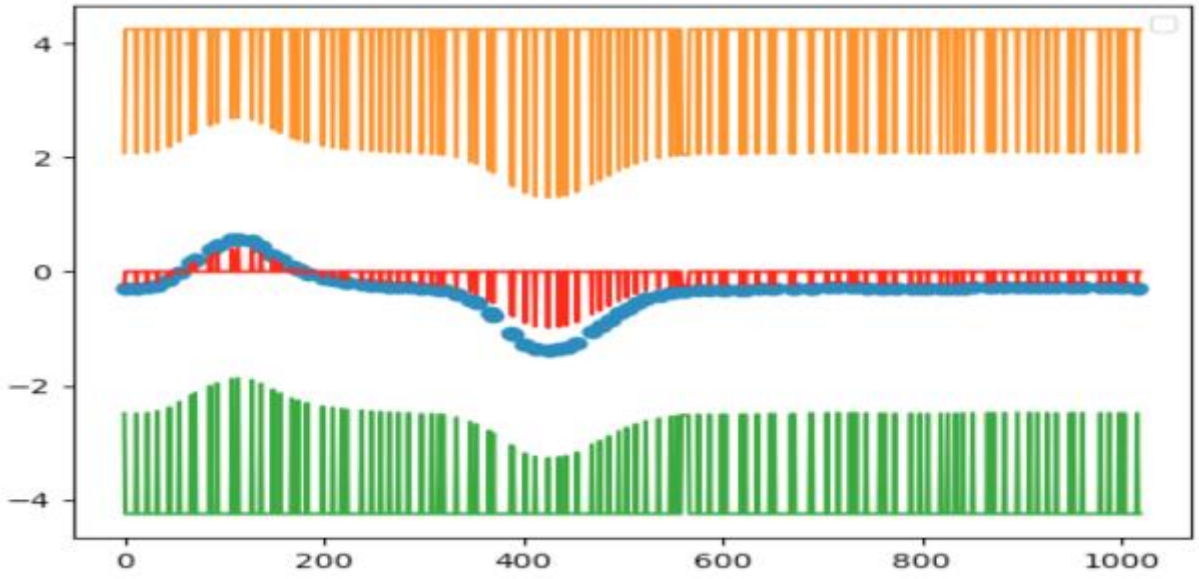
For example, rather than taking [1,1,0.1] as initial guess as described above, we make sigmaf = -100, that is ,taking[-100,1,0.1], we have the graph as follows:



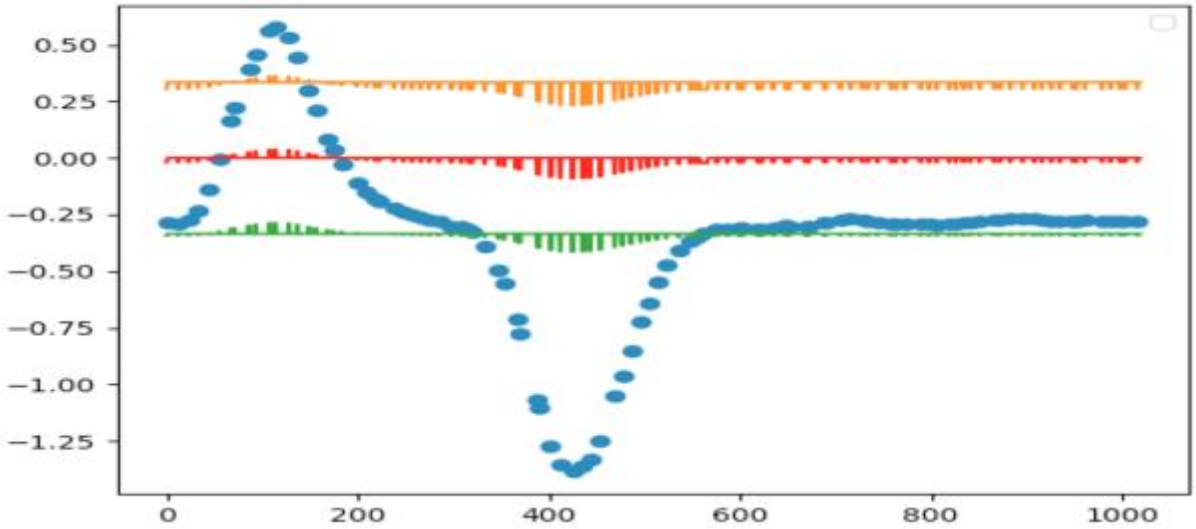
When finishing the Gaussian process, we get



It seems like the process does not converge at all. Indeed, it is this case. By the value returned, the log likelihood value is 77.16, which is $e^{76} \approx 3 * 10^{33}$
This huge error reflects the diverging facts.
Similar issues happen with sigmal as well.
we make sigmal= 100, that is, taking [1,100,0.1], we have the initial guess graph as



and after finishing the Gaussian process, we get



It seems like the process does not converge at all. Indeed, it is this case. By the value returned, the log likelihood value is 76.03, which is $e^{76} \approx 10^{33}$
This huge error reflects the diverging facts.

Similar issues happen with sigman as well. When we set the initial guess to be 1000, the function value is still about 76 and gives similar big error.

IV Conclusion:

As described above, we find that Gaussian process is a good way of modeling the data we got. If we pick relatively sufficient number of data, the hyper parameters do not change over time. If we were working on the same or similar patterns of data, the hyper parameters do not change either. If we can have multiple trace of the same thing, the combined version of the data can actually lead to better result. When some data is missing, by using Gaussian process, we can get a relative tight interval of the theoretical value of data should be. This theoretical value can guide our further work if necessary. When using Gaussian process, we need to be aware that the initial guess should be in a reasonable interval. If not near the optimal solution, the process may give a really bad result. Also, with the changing of the data pattern and the object, the hyper parameters will change accordingly. This actually teaches us the hyper parameter we need will be decided by the actual situation we are facing and it is not likely that one parameters can works best for every case, even if only the diffence lies on the tracing object.

IV Reference

Marsland, Stephen. Machine learning: an algorithmic perspective. CRC Press, 2015.