Gaussian Process Regression

Nicolas Brissonneau , UTEID:nb24488 , email:nicolasb@utexas.edu

I. Introduction

I am considering the problem of fitting the x, y or z coordinate of a human body-part. In order to achieve this, I will make use of Gaussian Process Regression (GPR) by using combinations of multivariate distribution with mean vector μ and covariance matrix Σ and use gradient descent to optimize for hyper-parameters.

II. Method

Let's consider the Gaussian process $f(t)\tilde{}\mathcal{N}(m_f(t),K_{ff}(t,t))$, a multivariate normal distribution for $t=[t_1,...,t_n]$ thus defining f as a prior to which we add noise $n\tilde{}\mathcal{N}(0,\sigma^2 I)$ to get y in:

$$y=f+n \tag{1}$$

$$p(y,f)=p(y|f)p(f)=N\Big(\Big[\begin{array}{c} f \\ y \end{array}\Big];\Big[\begin{array}{c} m_f \\ m_y \end{array}\Big],\Big[\begin{array}{c} K_{ff}K_{fy} \\ {K_{fy}}^T K_{yy} \end{array}\Big]\Big) \tag{2}$$

We can thus update the posterior using a new input $y(t_i)$:

$$f|y(t_i)\tilde{}\mathcal{N}(K_{fy}{K_{yy}}^{-1}(y(t_i)-m_y),K_{ff}-K_{fy}{K_{yy}}^{-1}{K_{fy}}^T) \tag{3}$$

Now in order to use gradient descent, we redesign the kernel with hyper-parameters σ_f, σ_l and σ_n such that:

$$k(t_i,t_j)=e^{\sigma_f}e^{-\frac{1}{2}e^{\sigma_l}|t_i-t_j|^2}+e^{\sigma_n}I \tag{4}$$

Where σ_n weights the noise, σ_l represents the weight each data point's neighbours should have on him and σ_f scales. We can then use the log of the likelihood for our gradient descent with:

$$\frac{\partial k}{\partial \sigma_f}=e^{\sigma_f}e^{-\frac{1}{2}e^{\sigma_l}|t_i-t_j|^2} \tag{5}$$

$$\frac{\partial k}{\partial \sigma_l}=e^{\sigma_f}e^{-\frac{1}{2}e^{\sigma_l}|t_i-t_j|^2}\times(-\frac{1}{2}e^{\sigma_l}|t_i-t_j|^2) \tag{6}$$

$$\frac{\partial k}{\partial \sigma_n}=e_n^{\sigma}I \tag{7}$$

We will cover two methods which consist on either relying on the global or the local kernel. A global kernel is defined as a set of hyper-parameters which cover the whole data, while a local kernel tries to fit pieces of the data one after the other, by "sliding" along it and updating the hyper-parameters as it goes.

III. Results

Sklearn libraries have been used to provide the implementation of GPR and the results which follow have been gathered with it. Moreover, the data shown here is the product of a random mix between 5 sets of data.

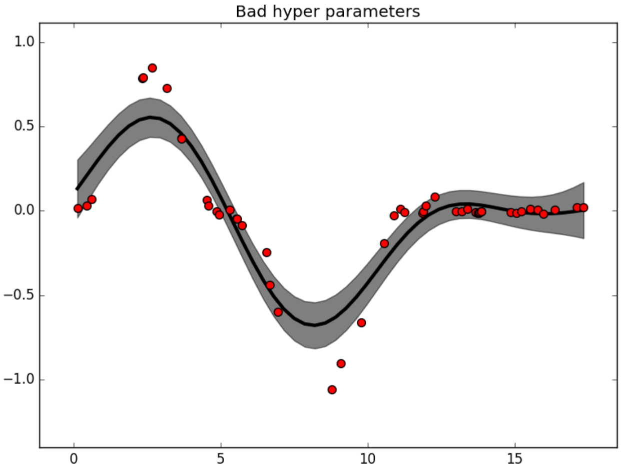

Fig. 1. Wrong hyper-parameters bounds

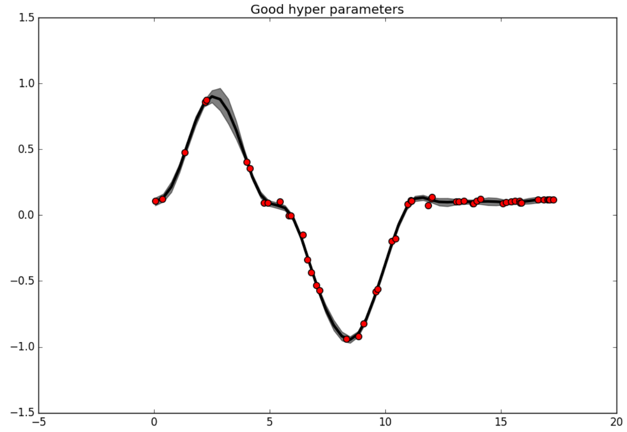

Fig. 2. Good fit

A. Global kernel

In this subsection, each plot is made of 50 points randomly chosen over the approximately 1300 existing in the data. Each point's index is then randomly selected among one of the five sets of data. Figure 1 and Figure 2 show how our GPR algorithm can fit similar data set very differently depending on the choice of hyper-parameters. In Figure 1, the covariance is very large even when a lot of data points demonstrate knowledge about a specific segment of data. In Figure 2 however, we only notice the covariance increasing around $t_i=2.5s$ as very little information (data points) are present near this part of the curve.

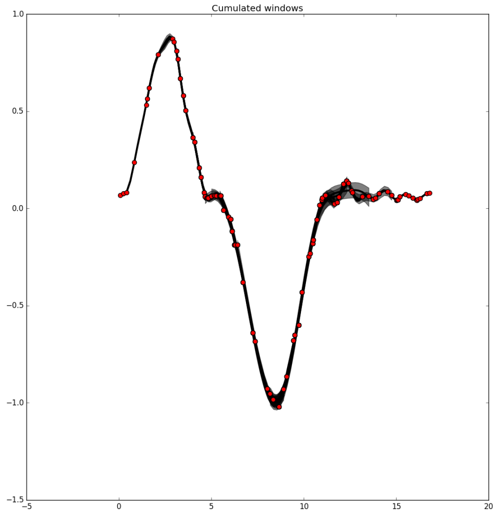

Fig. 3. Superposition of results when using windows

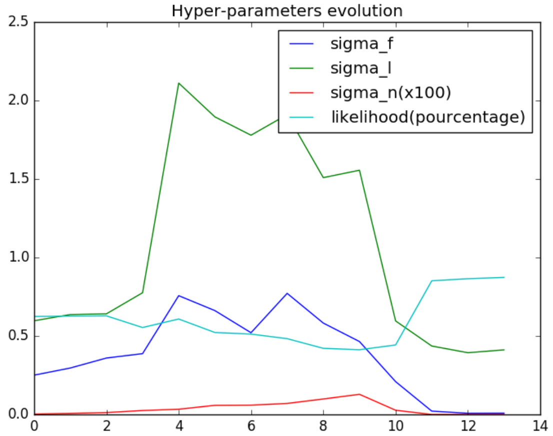

Fig. 4. Evolution of hyper-parameters

B. *Local kernel*

Here we propose local kernel, meaning that we apply the same technique as described above while only focusing on a specific window of the plot instead of the whole plot. Here we have picked 100 points from the data set, but we will only take 30 points at a time into account, and after every iteration we ”translate” this window by an increment of 10 points until we have covered all 100 points at least once. Figure 3 seems very similar to Figure 2 but it actually shows the superposition of the predictions, hence the darker areas. We notice that the fitting works very well too for the sliding windows. Figure 4 shows the evolution of the hyper-parameters fitted at each step. We have decided to represent the likelihood as a percentage and to magnify $sigma_n$ a hundred times to help with the graph's visibility. Figure 5 shows the averaging of the cumulated fitting from Figure 3

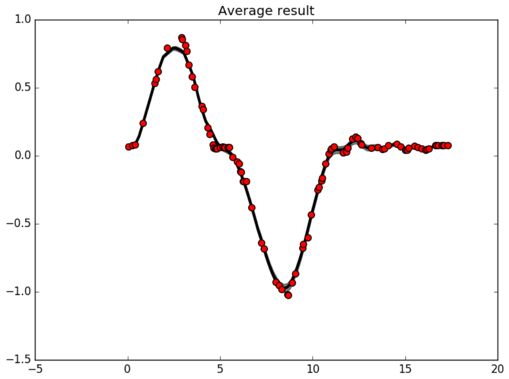

Fig. 5. Averaging cumulated results

C. *Comparison*

After experimenting and alternating between Global and Local kernels, it seems that the fitting is more elegant in the case of the Global kernel as shows the likelihood of Figure 6 compared to the likelihood's for each window in Figure 4, however there might be valuable information gained from using the Local kernel. Indeed, even though the local kernel shows overall worse fitting than the Global one, it shows the fluctuations of the best fitting local hyper-parameters.

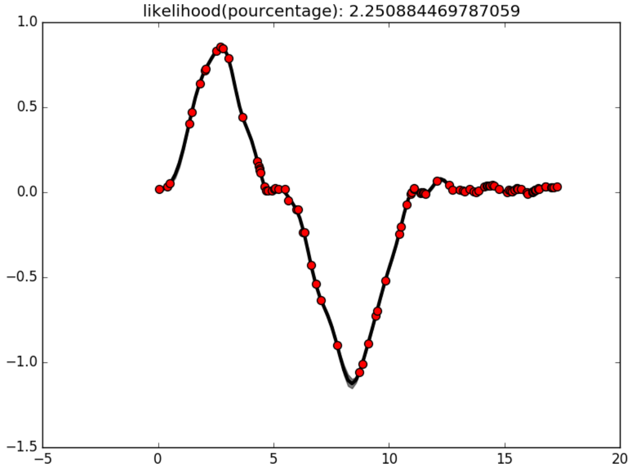

Fig. 6. Likelihood for Global kernel

IV. Conclusion

The Global kernel is useful for fitting rapidly a large set of data with good performance, while the Local kernel requires more time and sometimes mores total data points to achieve similar results. However the Local kernel also shows the fluctuations of the hyper-parameters which may indicate additional information about the person moving such as co-contraction. Paying attention to sudden changes in the fitted hyper-parameters in all 5 subplots from Figure 7, we notice their sudden fluctuation near $t_i = 8s$ in each subplot, while having relatively similar values before and after.

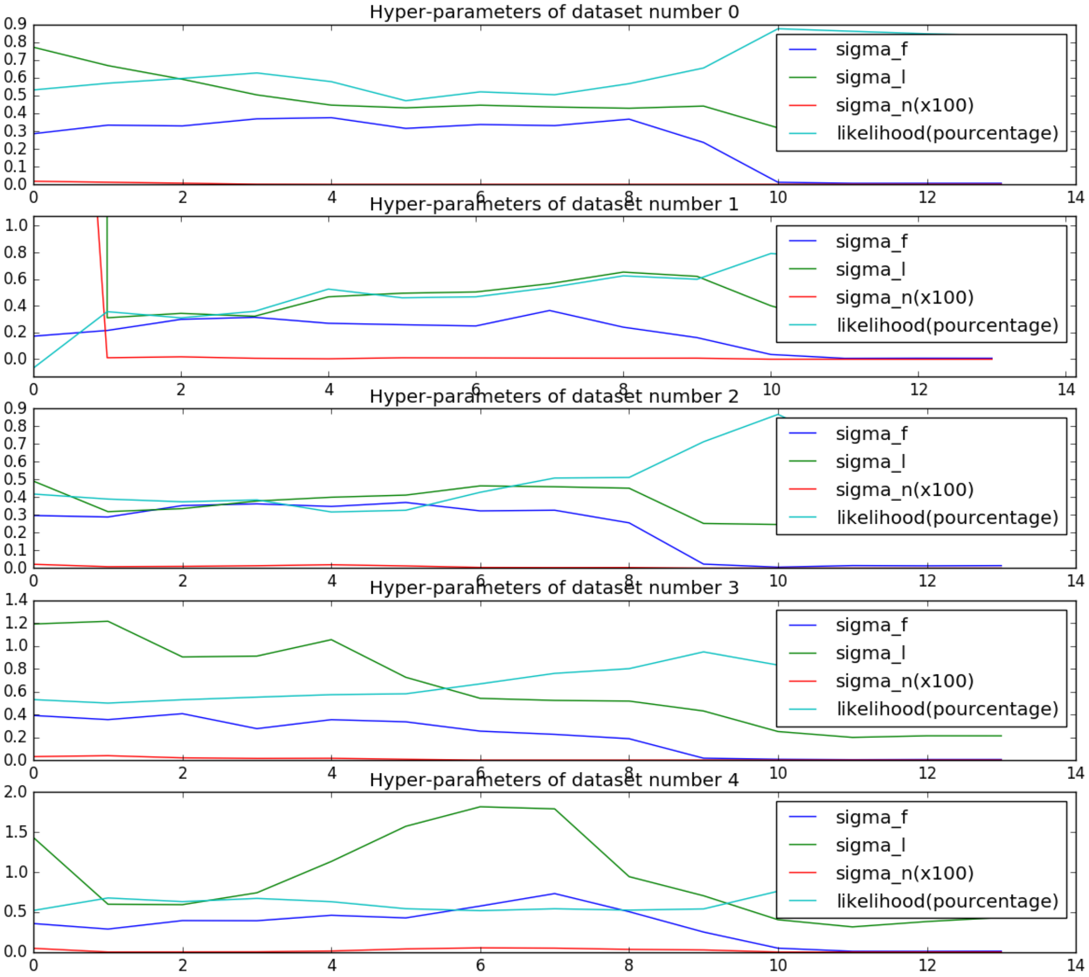

Fig. 7. Table of comparison between 5 samples hyper-parameters