CS391L HW4: Gaussian Process

Mingyo Seo
UT EID: ms84662
Email: mingyo@utexas.edu

***Abstract*—In this assignment, an overview of Gaussian processes (GP) is presented and is applied to estimate a trajectory from human motions. I used a dataset of 5 different iterations of the same motion and mixed signals to generate a single trajectory. In particular, I used an exponent RBF kernel for trajectory processing. The regression was implemented by the gradient ascent algorithm to maximize the negative log-likelihood function, which outputs the optimal kernel hyperparameters. I also compared the use of a single global GP and a set of GP's fit to local data and studied how the local GP's hyperparameters change.**

I. Introduction

The problem of estimating a trajectory from multiple demonstrations can be applied to many areas. In this paper, in particular, I implemented a Gaussian process(GP) model to estimate a trajectory from human demonstration. GP is one of the statistical parameter-free models. By using a GP model, we can predict and estimate a likelihood at any given dataset. We can consider a GP as a multi-dimensional Gaussian Probability density function (PDF). A GP is characterized by a set of mean and variance functions. By sampling a finite or infinite-dimensional PDF, we can regress the mean and variance functions, which is formulated as,

$$\begin{aligned} f \in \mathcal{N}(\mu(\boldsymbol{x}), V(\boldsymbol{x})) \text{ sampled from PDF,} \\ \text{where} \quad \mu \in R^n,\ V \in R^n \times R^n. \end{aligned} \tag{1}$$

For the implementation of GP, I used mixed motion demonstration data from multiple datasets to estimate a motion trajectory.

The answers to the HW4 questions are included in the following sections.

- Fig. 2: Pridiction of the motion data using a global kernel
- Fig. 1: Prediction of motion data using local kernels
- Fig. 3, 4: Performances of the global/local kernels
- Fig. 5: Hyperparameter of local kernels at each frame
- Tab. I: Intervals where the kernels cluster with similar values

II. Method

A. Finite Gaussian Process

We assume that the prior distribution follows a Gaussian distribution of the mean(m_f) and ithe dentity co-variance matrix(K_{ff}), represented as,

$$f(\boldsymbol{x}) \ \sim \ GP(m(x), \kappa(x, x')) \Leftrightarrow f \in \mathcal{N}(m_f, K_{ff}), \tag{2}$$

where

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)] \\ \kappa(x, x') &= \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]. \end{aligned} \tag{3}$$

Thus, we can update the posterior from a data (X, y) By using Bayes' theorem, the GP is represented as,

$$\begin{aligned} &P(f|y) \\ &\quad = \mathcal{N}(K_{fy}^T K_{yy}^{-1}(y - m_y) + m_f, K_{ff} - K_{fy}^T K_{yy}^{-1} K_{fy}^T). \end{aligned} \tag{4}$$

The above formulas exist in an infinite dimensional space.

To apply the form of an infinite GP into a finite data set, we reformulate the GP formulation and use a Kernel function to calculate the co-variance matrix. If the optimal GP of the finite data set (X, y) is given by f, then predictions for new data X_* given by f_* are given as,

$$\begin{bmatrix} f \\ f_* \end{bmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{5}$$

Instead of sampling multiple f_* from the ensemble of equation 5, we can condition our prior of f for (X, y). If there is no noise in the observed output y, then the best solution is $f = y$, and

$$\begin{aligned} &f_*|X_*, X, f \\ &\quad \sim \mathcal{N}(K(X_*, X)K(X, X)^{-1} f, \\ &\qquad K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)) \end{aligned} \tag{6}$$

In particular, we use an exponent RBF kernel to calculate the co-variance matrix K and Gaussian noise to estimate raw data errors, given as

$$\begin{aligned} y(u) &= f(u) + \epsilon \\ k(u_i, u_j) &= e^{\sigma_f} e^{\left(\frac{1}{2} e^{\sigma_l} ||u_i - u_j||^2\right)} \\ q(u_i, u_j) &= k(u_i, u_j) + e^{\sigma_n} \delta_{ij}, \end{aligned} \tag{7}$$

where $\epsilon \sim \mathcal{N}(0, e^{\sigma_n})$. This also can be written as,

$$Q(X, X) = K(X, X) + e^{\sigma_n} I_{n \times n}. \tag{8}$$

Then, we can estimate f_* for new values X_* using equation 6, as

$$\begin{aligned} \bar{f}_* &= K(X_*, X)K(X, X)^{-1} y \\ \mathbb{V}[f_*] &= K(X_*, X_*) - K(X_*, X)Q(X, X)^{-1}K(X, X_*), \end{aligned} \tag{9}$$

where $K(X_*, X) = [K(X, X_*)]^T$. The variance at each x_* is given by the diagonal elements of $\mathbb{V}[f_*]$. The standard deviations from these values can be used to compute the 95% confidence interval at each x_*. Also, from f_*, we can calculate the expectation and variance of the observations y_*. Since

adding white noise does not change mean observation, the mean is given as,

$$\bar{y_*} = \bar{f_*}. \tag{10}$$

The variance is given as

$$\mathbb{V}[y_*] = Q(X_*, X_*) - K(X_*, X)Q(X, X)^{-1}K(X, X_*). \tag{11}$$

B. GP Hyperparameter regression

A regression on the hyper-parameters, $\sigma = [\sigma_f, \sigma_l, \sigma_n]$ obtains the optimal GP over the given data. Consider the log-likelihood function, given as

$$logP(y|x, \sigma) = -\frac{1}{2}y^T Q^{-1} y - -\frac{1}{2}log(det(Q)) - \frac{n}{2}log(2\pi) \tag{12}$$

The GP regression can be processed by maximizing the log-likelihood function [1], as

$$\sigma_{i+1} = \sigma_i + \eta \cdot [\nabla_\sigma (logP)]_{\sigma_i}. \tag{13}$$

Here, η is the vector learning rate. Then, the gradient of the $logP$ function is given by:

$$\nabla_\sigma (logP) = -\frac{1}{2}y^T \frac{\partial Q^{-1}}{\partial \vec{\sigma}} y - -\frac{1}{2}\frac{\partial log(|Q|)}{\partial \vec{\sigma}}, \tag{14}$$

where

$$\begin{aligned} \frac{\partial log(|Q|)}{\partial \sigma} &= trace\left(Q^{-1}\frac{\partial Q}{\partial \sigma}\right) \\ \frac{\partial Q^{-1}}{\partial \sigma} &= -Q^{-1}\frac{\partial Q}{\partial \sigma}Q^{-1}, \end{aligned} \tag{15}$$

and the partial differentials are given as,

$$\begin{aligned} \frac{\partial Q_{ij}}{\partial \sigma_f} &= K_{ij} \\ \frac{\partial Q_{ij}}{\partial \sigma_l} &= K_{ij} \times \left(-\frac{1}{2}e^{\sigma_l}||x_i - x_j||^2\right) \\ \frac{\partial Q}{\partial \sigma_n} &= e^{\sigma_l} I. \end{aligned} \tag{16}$$

III. Results

In the assignment, I trained the GP model to estimate the trajectory from multiple demonstrations of the same task. For the training process, I used the learning rate $\eta = 0.01$ and the finite time-horizon of 50 frames. For moving windows, I used the 5-frame stride. The set of the kernel hyperparameters was used for the initialization.

$$\begin{aligned} \sigma_f &= -1.9496 \\ \sigma_l &= -13.837 \\ \sigma_n &= -6.6605. \end{aligned} \tag{17}$$

For regression, I limited the maximum iteration of gradient ascent under $2,000$. To avoid divergence, I also terminated the gradient ascent process at each window if the magnitude of the gradient reached the limit under 1, as

$$||\,[\nabla_\sigma (logP)]_{\sigma_i}\,|| < 1. \tag{18}$$

In this assignment, I used the dataset of Subject *YY* and Marker *14_x*. To avoid the effects of the raw data noise, we mixed the 5 demonstrations to generate a single trace data.

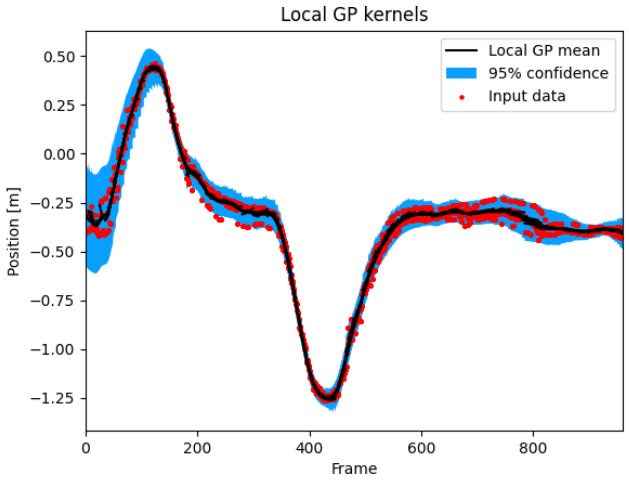

Fig. 1. Visualization of the local GP model at Subject *YY* and Marker *14_x*.

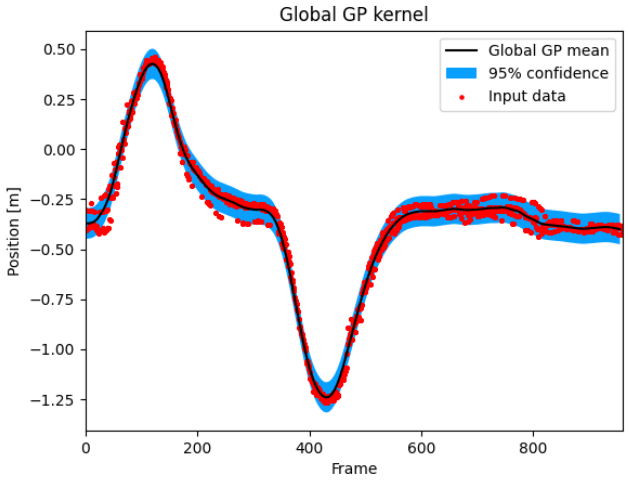

Fig. 2. Visualization of the global GP model at Subject *YY* and Marker *14_x*.

A. Kernels

The GP model with the local kernel is presented in Figure 1. The GP model with the global kernel is presented in Figure 2. The optimal set of the hyperparameters used in the global kernel were computed as below,

$$\begin{aligned} \sigma_f &= -2.6004 \\ \sigma_l &= -12.206 \\ \sigma_n &= -6.5438. \end{aligned} \tag{19}$$

To compare the performance of the local GP model and the global GP model, I consider two metrics: log-likelihood, the target function of the gradient ascent, and square errors. The log-likelihood of the models is presented in Figure 3. In the plots, the global GP model has high likelihood values at the beginning of the process, but the local GP model shows higher likelihood values in most of the processes. From this, we can find that the local GP kernel is poorly initialized

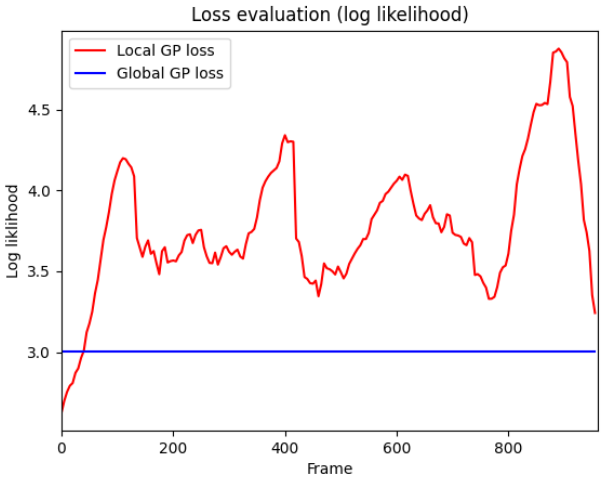

Fig. 3. Log-likelihood comparison between the local and global GPs.

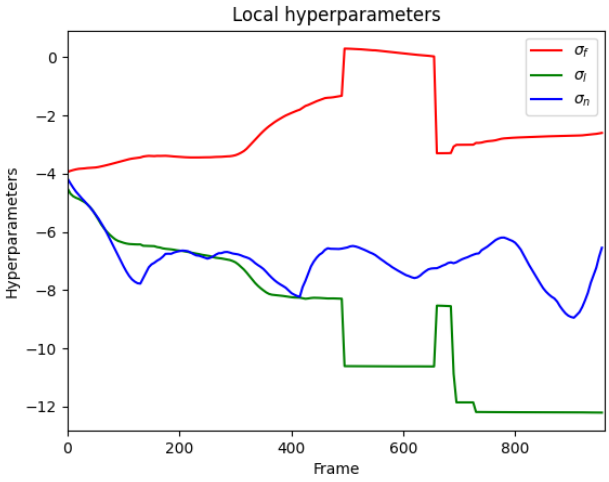

Fig. 5. Hyperparameters of the local GP model.

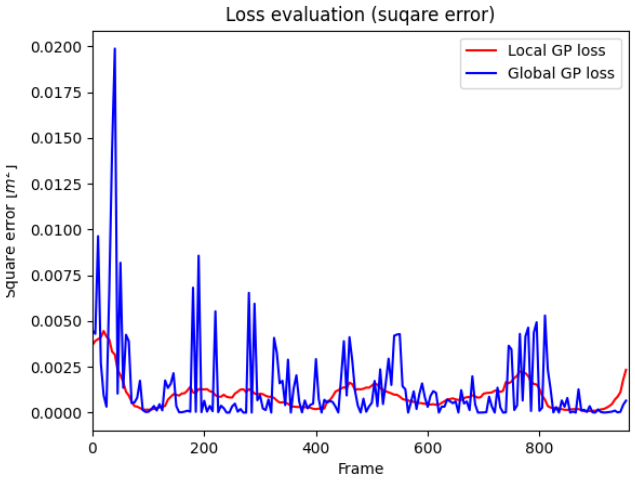

Fig. 4. Square error comparison between the local and global GPs.

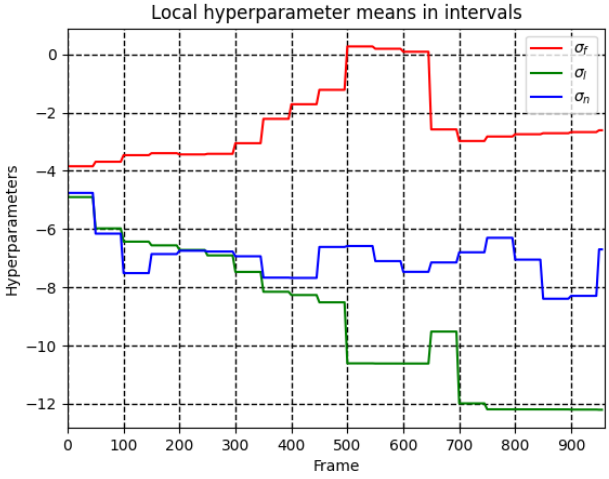

Fig. 6. Correlation between each source signal and the corresponding the restored signal

but becomes fitting well on the dataset as the regression is processed. On the other hand, from the square errors of the models, as presented in Figure 4, the global GP model has larger variance and magnitude on square errors. This implies that the globally optimal kernel hyperparameters may not fit on the local data. Also, the results of better performance in the local GP models imply the correlation of the two different metrics.

B. Hyperparameters

The kernel hyperparameters of the local GP model are presented in Figure 5. For the analysis of intervals with similar values, I visualized the means of the hyperparameters at each interval in Figure 6. In Table I, intervals with similar values are represented with interval IDs. In particular, we can find that σ_f and σ_n have shared intervals, which implies those two parameters are correlated. Also, in the Interval 18-19 (Frame 850-950), where the trajectory in Figure 1 does not change aggressively, all the parameters have similar values. From this result, we can confirm that kernel hyperparameters highly depend on the characteristics of trajectories.

IV. Summary

The GP model and gradient ascent for regressing the model from raw motion data are implemented in the assignment. For training, I mixed 5 human motion datasets of the same task demonstration and estimated the trajectory by regressing the GP model. By the regression, I optimized the optimal set of hyperparameters for the finite GP kernel from the given data. From the results of the experiments, we can confirm that a sliding window kernel can fit the given data more accurately than a global kernel function. This is verified by taking the mean of log-likelihood and square errors over all the points it spans. Also, we confirmed that local kernels have similar

TABLE I
INTERVALS WITH SIMILAR VALUES

Interval ID	Frame	σ_f	σ_l	σ_n
1	0-50			
2	50-100			
3	100-150	3-6		
4	150-200	3-6	4-7	
5	200-250	3-6	4-7	
6	250-300	3-6	4-7	
7	300-350		4-7	
8	350-400			
9	400-450			
10	450-500		10-11	
11	500-550	11-13	10-11	11-13
12	550-600	11-13		11-13
13	600-650	11-13		11-13
14	650-700			
15	700-750			
16	750-800	16-20		16-20
17	800-850	16-20		16-20
18	850-900	16-20	18-19	16-20
19	900-950	16-20	18-19	16-20
20	950-1000	16-20		16-20

hyperparameters when the behavior of the motion data does not change aggressively.

REFERENCES

[1] Christopher M. Bishop (2006) *Pattern Recognition and Machine Learning*, Springer.