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. We used a dataset of 5 different iterations of same motion and mixed signals to generate a single trajectory. In particular, we used an exponated RBF kernel for the trajectory processing. The regression was implemented by the gradient descent algorithm to maximize the negative log likelihood function, which outputs the optimal kernal hyperparameters. We 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 estimate a trajectory from multiple demonstration can be applied to many areas. In this paper, in particular, we implemented a Gaussian process(GP) model to estimate a trajectory from human demonstration. GP is one of 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 an multi-dimensional Gaussian Probability density function (PDF). A GP is characterized by a set of the mean and variance functions. By sampling an finite or infinite dimensional PDF, we can regress the mean and variance functions, which is formulized 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, we used a mixed motion demonstration data from multiple datasets to estimate a motion trajcetory.

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

- Fig. 1: Visualization of base, mixed, recovered signals
- Fig. 2, 3, 4, 5, 6: Correlation between source and recovered signals

II. Method

A. Finite Gaussian Process

We assume that the prior distribution follows a Gaussian distribution of mean(m_f) and identity 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 exponated 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 also 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

a starting point of $\vec{\sigma}_0 = [-1, -8, -8]$ with $\eta = 10^{-3}$ was stable for almost all data and markers in the experiment. The gradient ascent was stopped when the magnitude of the gradient fell below a tolerance value (tol), i.e. $||\vec{\nabla}_\sigma(Q)||_1 < tol$. A tolerance of 1 was found to best based on computation time and goodness of fit.

There is a strong negative correlation between the hyperparameters $sigma_f$ and $sigma_l$. When multiple dataset were mixed together to produce noisy data, then this negative correlation was not as apparent.

The ICA implementation of using MLE and gradient descent described in Section II is implemented under the environment of python scripts (3.8.5) and numpy (1.17.5). The model is trained and evaluated by the sound dataset from the class website [**?**]. We used random mixes of $n = 3$ source signals from the dataset and initialized W_0 to be random matrices where each entry in $[0, 1]$.

A. Signal restoration

In the assignment, the performance of the model is evaluated by correlation between original signals and recovered signals. We paired a restored signal with the orignal signal that has the largest magnitude of correlation. In the experiment, we used $n = 3$, $m = 3$, $\eta = 0.01$, $t = 16$ and the maximum interation $k_{\max} = 5000$.

The source signals, the mixed signals, and the restored signals after $k_{\max}$ are presented in Fig. 1. The plot of correlation between the source signals and the restored signals is presented in Fig. 2.

Due to the semi-symmetric nature of waves, the largest correlation of Signal #3 became negative, which implies that the extracted signal has the opposite phase of the original source signals. We use the plot of absolute correlation as Fig. 3 and find that correlation increase as an iteration number increases. For readability, we use absolute correlation for analyzing the effectiveness of the model in the following subchapters. Also, we can find that each source signal matches with the paired signals, which implies the ICA model successfully extracts original signals from mixed signals.

B. Batch size

The plot of absolute correlation at different batch sizes t is presented in Fig. 4. In the experiment, we used $n = 3$, $m = 3$, $\eta = 0.01$, and $k_{\max} = 5000$. From the results, we can find that the correlation increases fastest at the batch size $k = 32$. In a larger batch size, there is a negligible drop at the speed of correlation magnitude increases. On the other hand, a batch size causes a significant drop, which implies that the model suffers from a lack of sample information.

C. Learning rate

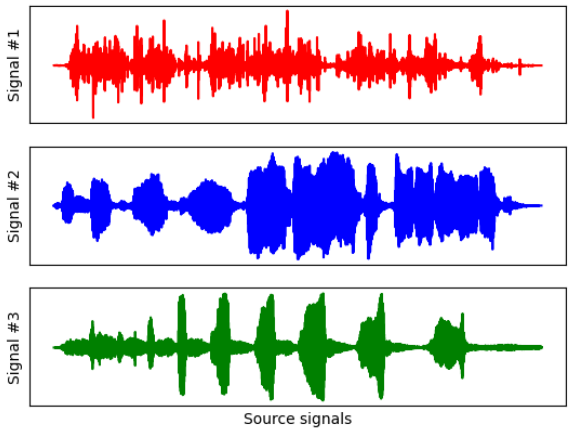
The plot of absolute correlation at different learning rates η is presented in Fig. 5. In the experiment, we used $n = 3$, $m = 3$, $t = 16$, and $k_{\max} = 5000$. From the results, we can find that a higher learning rate yields faster. However, an excessively high learning rate prevents the convergence at gradient descent, which causes unstable performance, such as perturbations in correlation remaining even after enough iterations.

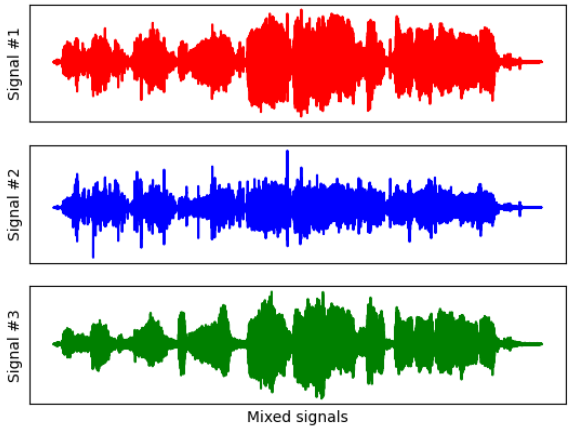
D. Sample size

The plot of absolute correlation at different sample signal numbers m is presented in Fig. 6. In the experiment, we used $n = 3$, $t = 16$, $\eta = 0.01$, and $k_{\max} = 5000$. From the result at $m = 2$, we can find that the model fails to restore Signal #1. This implies that a smaller sample number than the given source signal number, $m < n$, causes the dimension issue at the unmixing W, where the model cannot fully extract the original signals.

IV. Summary

The GP model and gradient descent for regressing the model from raw motion data are implemented in the assignment. For training, we mixed 5 human motion dataset of the same task demonstration and estimated the trajectory by regressing the





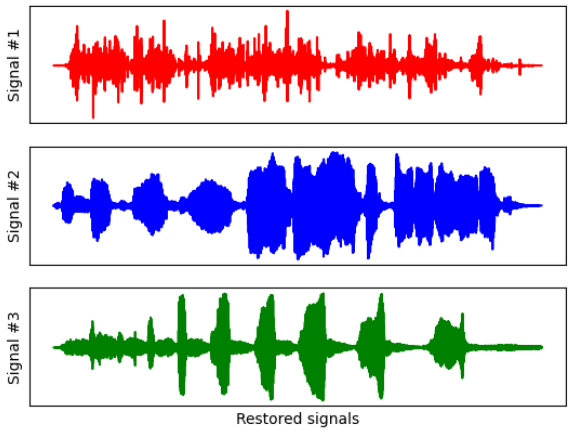


Fig. 1. Visualization of the signals: (upper) orignal signals, (middle) mixed signals, and (bottom) restored signals

GP model. By the regression, we optimized the optimal set of hyperparamters for the finite GP kenel from the given data. From the results of the experiments, we can confirmed that

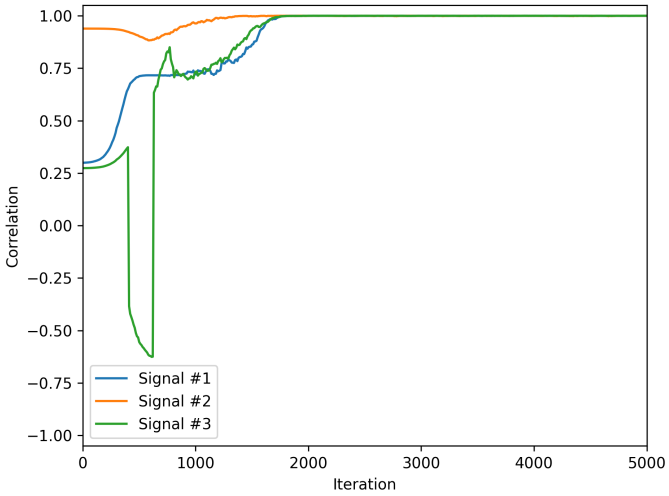


Fig. 2. Correlation between each source signal and the corresponding the restored signal.

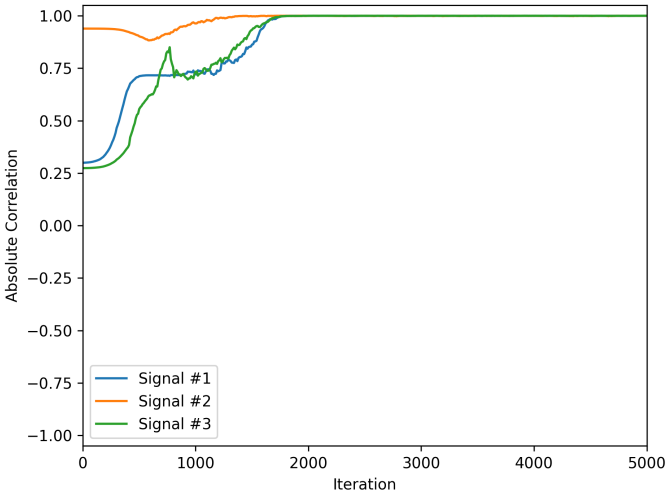


Fig. 3. Correlation between each source signal and the corresponding the restored signal.

a sliding window kernel is able to fit the given data more accurately than a global kernel function. This is verified by taking the mean of the log-likelihood and square errors over all the points it spans.

References

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

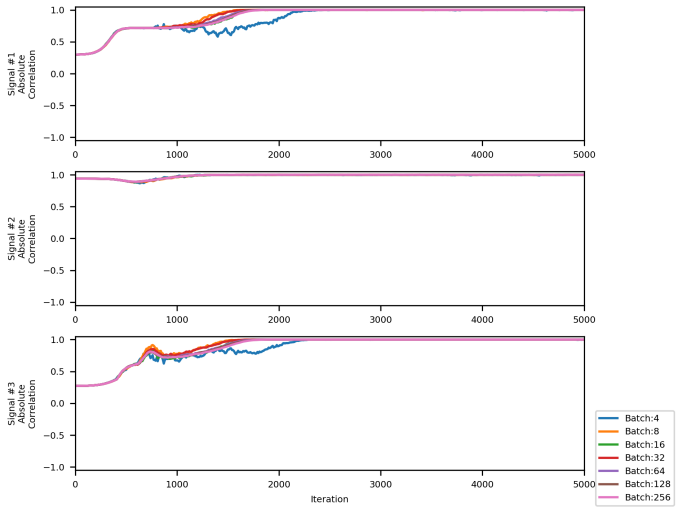


Fig. 4. Absolute correlation changes on batch size: batch sample sizes are chosen in [4, 8, 16, 32, 64, 128, 256].

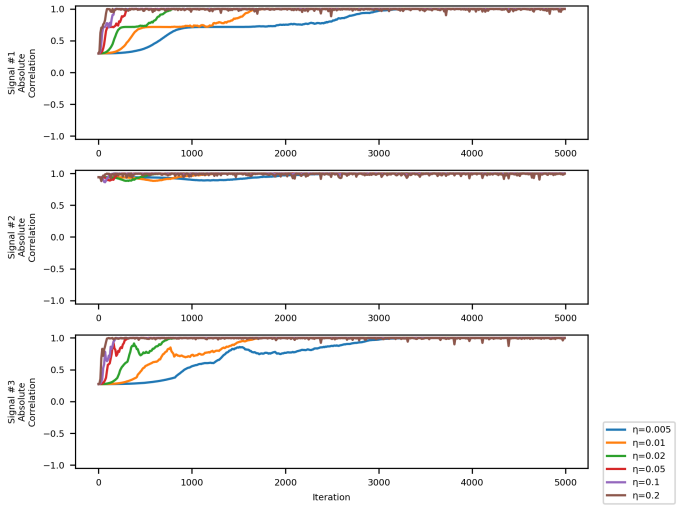


Fig. 5. Absolute correlation changes on learning rate: learning rates are chosen in [0.005, 0.01, 0.02, 0.05, 0.1, 0.2].

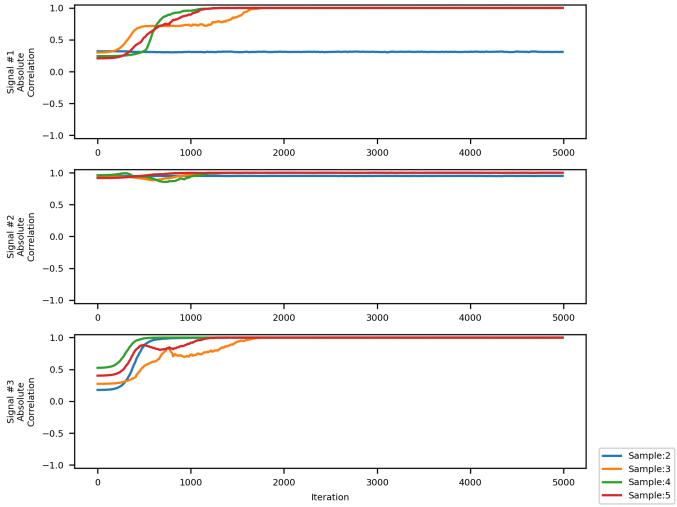


Fig. 6. Absolute correlation changes on sample sizes: sample sizes are chosen in [2, 3, 4, 5].