Gaussian Process Regression, A Brief Study

Saeed Mohseni seh deh
The Bradley Department of Electrical and Computer Engineering at Virginia Tech

1 Introduction

In this report, an investigation into Gaussian Process Regression (GPR) is conducted. GPR represents a form of regression that presupposes the Gaussian distribution assumption for both parameters and data, thereby inferring that the response output of a regression model adheres to a Gaussian distribution. This inference stems from the intrinsic properties of the Gaussian distribution, which persist through various operations. To elaborate, let us consider the following equation:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{D}) = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{w},\boldsymbol{D})p(\boldsymbol{w}|\boldsymbol{D},\boldsymbol{x}^*)d\boldsymbol{w} = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{w},\boldsymbol{D})p(\boldsymbol{w}|\boldsymbol{D})d\boldsymbol{w} = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{w},\boldsymbol{D})\frac{p(|\boldsymbol{D}|\boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{D})}d\boldsymbol{w}, \tag{1}$$

In this equation, $\boldsymbol{y}^*$ and $\boldsymbol{x}^*$ represent the test response and input, respectively, while $\boldsymbol{w}$ and $\boldsymbol{D}$ denote the model parameters and the training data. The last equality is derived from the application of Bayes' rule. Assuming a Gaussian distribution for both the data and the model, it becomes apparent that the distribution of the response follows a Gaussian distribution. This finding is noteworthy as it indicates that the model parameters are integrated out of the equation, resulting in the final output distribution being independent of the model parameters.

This report is dedicated to the simulation study of Gaussian Process Regression (GPR), with its data sourced from a problem involving robotic arm control [1]. Section 2 elucidates the underlying data model, including the input and output variables. Section 3 explores the simulation results and accompanying discussion, while Section 4 presents the concluding remarks.

2 Model and Data

The data utilized for Gaussian Process Regression (GPR) in this investigation are sourced from a robotic arm control scenario [1]. This robotic arm, with its shoulder fixed at the origin, comprises four segments. Each segment is capable of assuming an angle with respect to the horizontal axis. The variables pertinent to the problem encompass the lengths of each segment and the angles formed with the horizontal axis. The output or response variable denotes the distance of the arm tip from the origin within a (u, v) plane. As delineated, this problem entails eight-dimensional input and one-dimensional output. To elaborate, $\theta_i \in [0, 2\pi]$ and $L_i \in [0, 1]; \forall i \in 1, 2, 3, 4$. The response y is then computed as:

$$y = f(\boldsymbol{x}) = \sqrt{u^2 + v^2}, \tag{2}$$

$$\boldsymbol{x} = [\theta_1, \theta_2, \theta_3, \theta_4, L_1, L_2, L_3, L_4], \tag{3}$$

$$u = \sum_{i=1}^{4} L_i \cos(\sum_{j=1}^{i} \theta_j),$$

$$v = \sum_{i=1}^{4} L_i \sin(\sum_{j=1}^{i} \theta_j).$$

The subsequent section will elaborate on the specifics of the dataset generated using these equations for the purpose of the GPR simulation study.

3 Simulation

The simulation results are presented in this section, wherein various scenarios concerning parameters and methodologies are examined. Like many machine learning problems, the dataset in this study comprises both training and testing sets. The training data is represented as $\mathbf{X} \in \mathbb{R}^{N_{\text{train}} \times 8}$ for the input and $\mathbf{y} \in \mathbb{R}^{N_{\text{train}}}$ for the output. Similarly, the testing dataset is modeled as $\mathbf{X}' \in \mathbb{R}^{N_{\text{test}} \times 8}$ and $\mathbf{y}' \in \mathbb{R}^{N_{\text{test}}}$ for the input and output, respectively.

As discussed previously, the GPR method does not entail explicit parameters (except for certain internal hyperparameters in some instances). The sole requirement is the covariance matrix of the output variables to derive the distribution. A notable approach to obtaining this covariance matrix involves employing covariance functions, among which the squared exponential and linear functions stand out. Once this covariance matrix is constructed, determining the distribution of the output involves only matrix multiplication to compute the mean and covariance matrix of the output distribution.

In these simulations, two scenarios for the number of training and testing datasets are examined, namely $(N_{\text{train}}, N_{\text{test}}) \in \{(100, 100), (500, 500)\}$. For the covariance function, three kernels are investigated:

1. Squared Exponential Kernel: $k(x_i, x_j) = -\exp(\theta||x_i - x_j||^2)$, where θ is the scale parameter, which is optimized by minimizing the negative log-likelihood function of the training data,
2. Linear Kernel: $k(x_i, x_j) = x_i^T x_j$, and
3. Indicator Kernel: $k(x_i, x_j) = I(x_i = x_j)$.

For reporting purposes, graphs of real values and the mean of the GPR along with uncertainty intervals are displayed for each parameter. However, due to the eight-dimensional input, graphing poses challenges. Slice plots are utilized for demonstration purposes to provide an overall view of the model's performance.

As mentioned earlier, upon completion of GPR, a distribution is provided:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{D}).$$

Note that this distribution is conditional, thus its mean and variance are also conditional. One approach to reduce parameter dimensionality is through marginalization. Suppose $\mathbf{X} = [X_1, X_2, \ldots, X_N]$ are a set of independent random variables, and assume $E\{y|\mathbf{X}\}$ and $Var(y|\mathbf{X})$ are known. Then, to find marginalized quantities such as $E\{y|X_1\}$ and $Var(y|X_1)$, the following calculations must be performed:

$$E\{y|X_1\} = \int E\{y|\mathbf{X}\} f_{X_2}(x_2) f_{X_3}(x_3) \ldots f_{X_N}(x_N) dx_2 dx_3 \ldots dx_n, \tag{4}$$

$$Var(y|X_1) = -(E\{y|X_1\})^2 + \int (Var(y|\mathbf{X}) + (E\{y|\mathbf{X}\})^2) f_{X_2}(x_2) f_{X_3}(x_3) \ldots f_{X_N}(x_N) dx_2 dx_3 \ldots dx_n. \tag{5}$$

These integrals can be very challenging to solve. In the simple case of a uniform prior for the input parameters, the summation approximation can become the sum of values. However, since the space is eight-dimensional, the number of points at which the integrand should be computed is very large, N^8 (where N is the number of intervals each variable is divided into), making it computationally inefficient. Another solution is to directly infer $E\{y|\mathbf{X}\}$ and $Var(y|\mathbf{X})$ while keeping only one of the parameters as variable and others fixed at an arbitrary point. This approach is computationally efficient and provides a comprehensive overview of how the method performs in estimating each variable.

Additionally, the Mean Square Error (MSE) is reported as a measure of how well the method performs in each scenario. The MSE of different kernels will be compared to each other, and the different MSE resulting from varying length scale parameters of the squared exponential kernel is depicted to illustrate the effects of hyperparameters on the method's performance.

Figures 1 to 8 demonstrate the performance of the GPR on each variable. As discussed, each of these graphs represents slice plots, indicating the specific point at which they are calculated. Each figure comprises six subfigures. In each row, the kernel of the GPR is altered, and in each column, the number of training and testing datasets is varied. It is evident that only the squared exponential kernel satisfactorily performs for parameter estimation, while the other two kernels perform poorly. The flexibility of the squared exponential kernel allows it to effectively capture underlying patterns. While the linear kernel can adequately capture linear trends in the data, it struggles when confronted with nonlinear trends. The indicator kernel exhibits behavior akin to the nearest neighbor algorithm; however, it performs poorly in capturing complex data trends. In general, the squared exponential kernel outperforms the linear and indicator kernels, with the linear and indicator kernels exhibiting comparable performance, albeit with a slight advantage to the linear kernel.

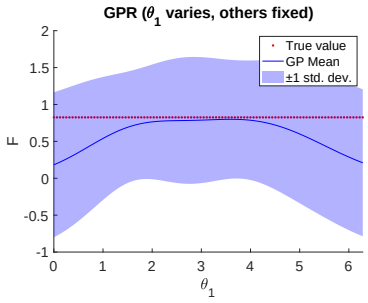
Another observation is that as the number of training data increases, the method can more accurately estimate the true value, resulting in decreased uncertainty intervals.

Figure 9 illustrates the effects of the length scale parameter on the final MSE of the model on the test data. It is evident that this hyperparameter significantly influences the output results of the algorithm. Table 1 summarizes the MSE for each kernel choice and for each data size.

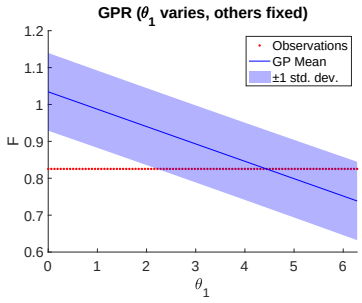
4 Conclusion

In this simulation study, Gaussian Process Regression (GPR) was investigated within the context of a robotic arm control problem. As demonstrated, the GPR model exhibits favorable performance for the aforementioned regression task with

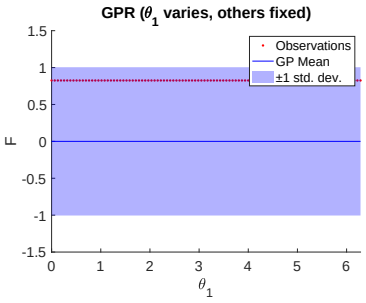
minimal need for model parameterization, save for a few hyperparameters. It was observed that the efficacy of the method is largely contingent upon the selection of kernel functions, where a judicious choice can yield substantial enhancements. One such favorable option is the employment of the squared exponential kernel. This kernel adeptly captures the nonlinear structures inherent in the data owing to its inherent flexibility. Furthermore, it was demonstrated that augmenting the size of the training dataset can lead to more precise model predictions, a characteristic common to many data-driven regression methodologies.



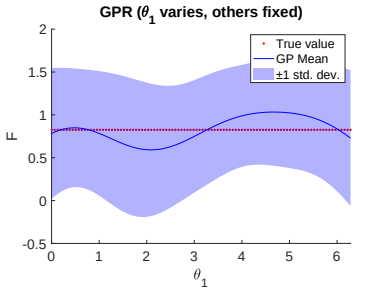
(a) θ_1, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



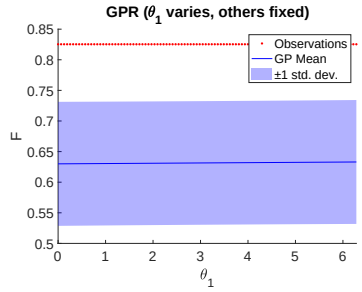
(b) θ_1, Kernel = Linear, $N_{train} = N_{test} = 100$



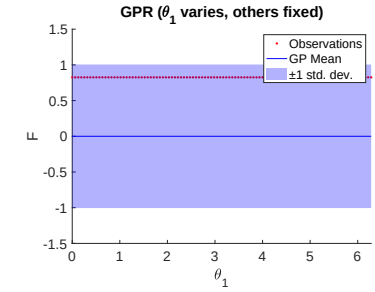
(c) θ_1, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) θ_1, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$



(e) θ_1, Kernel = Linear, $N_{train} = N_{test} = 500$

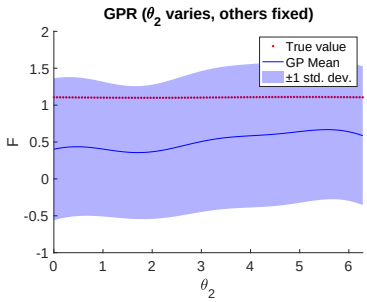


(f) θ_1, Kernel = Indicator, $N_{train} = N_{test} = 500$

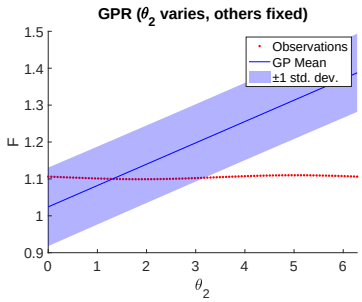
Figure 1: GPR estimation and uncertinity intervals of θ_1 at $[\theta_2, \theta_3, \theta_4, L_1, L_2, L_3, L_4] = [4.4650, 6.1768, 2.3771, 0.0319, 0.7935, 0.1825, 0.2635]$

References

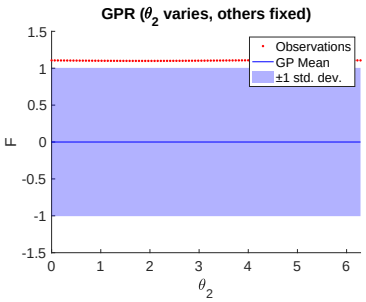
[1] J. An and A. Owen, "Quasi-regression," *Journal of complexity*, vol. 17, no. 4, pp. 588–607, 2001.



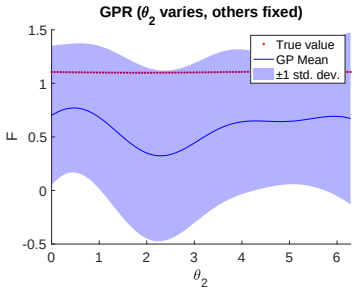
(a) θ_2, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



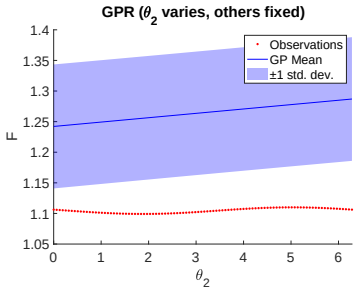
(b) θ_2, Kernel = Linear, $N_{train} = N_{test} = 100$



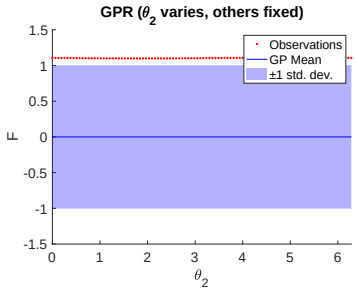
(c) θ_2, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) θ_2, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$



(e) θ_2, Kernel = Linear, $N_{train} = N_{test} = 500$

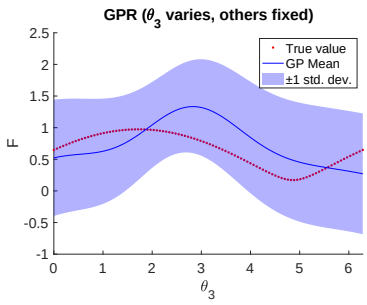


(f) θ_2, Kernel = Indicator, $N_{train} = N_{test} = 500$

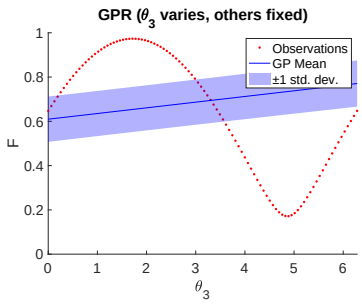
Figure 2: GPR estimation and uncertinity intervals of θ_2 at $[\theta_1, \theta_3, \theta_4, L_1, L_2, L_3, L_4] = [2.4292, 2.6999, 4.6007, 0.0054, 0.7779, 0.9328, 0.7674]$

Table 1: MSE values for various data sizes and different kernels.

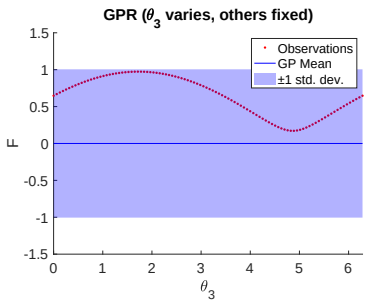
N	**Squared exponential**	**Linear**	**Indicator**
100	0.166	0.211	1.214
500	0.089	0.215	1.350



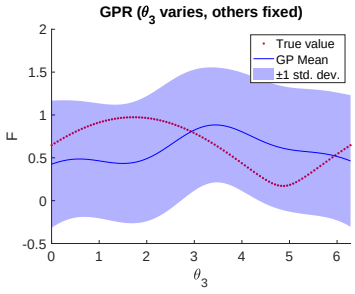
(a) θ_3, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



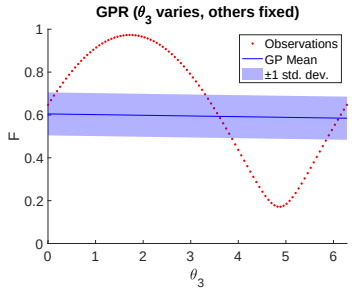
(b) θ_3, Kernel = Linear, $N_{train} = N_{test} = 100$



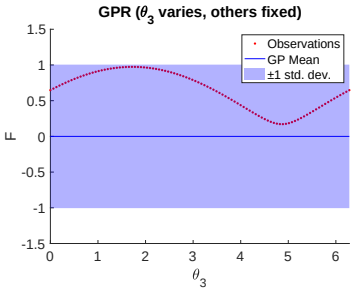
(c) θ_3, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) θ_3, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$

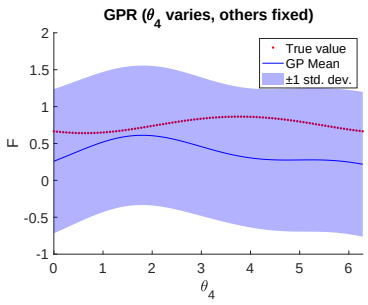


(e) θ_3, Kernel = Linear, $N_{train} = N_{test} = 500$

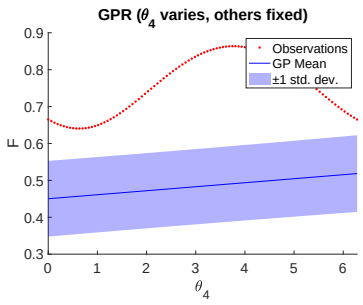


(f) θ_3, Kernel = Indicator, $N_{train} = N_{test} = 500$

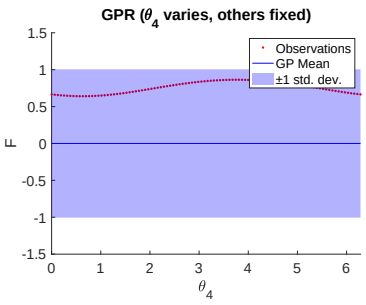
Figure 3: GPR estimation and uncertinity intervals of θ_3 at $[\theta_1, \theta_2, \theta_4, L_1, L_2, L_3, L_4] = [1.9219, 2.3591, 2.3797, 0.6500, 0.1181, 0.0189, 0.4146]$



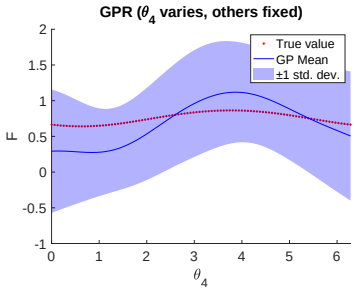
(a) θ_4, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



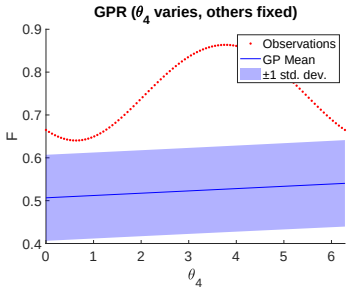
(b) θ_4, Kernel = Linear, $N_{train} = N_{test} = 100$



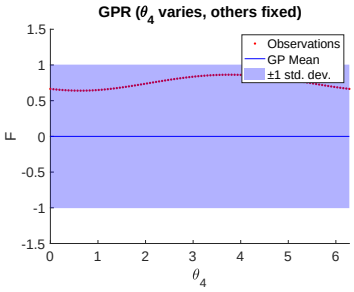
(c) θ_4, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) θ_4, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$

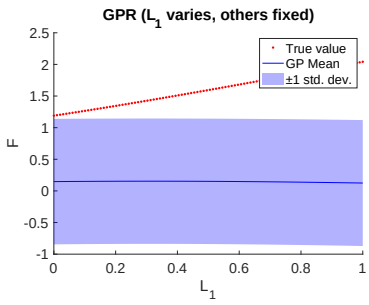


(e) θ_4, Kernel = Linear, $N_{train} = N_{test} = 500$

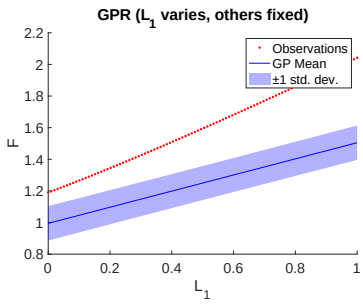


(f) θ_4, Kernel = Indicator, $N_{train} = N_{test} = 500$

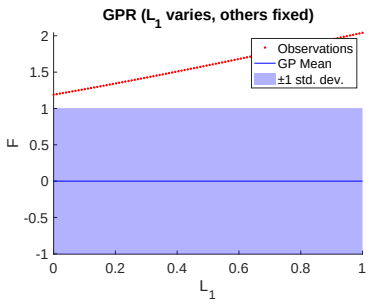
Figure 4: GPR estimation and uncertinity intervals of θ_4 at $[\theta_1, \theta_2, \theta_3, L_1, L_2, L_3, L_4] = [3.5464, 0.2228, 2.4702, 0.5355, 0.3378, 0.1388, 0.1116]$



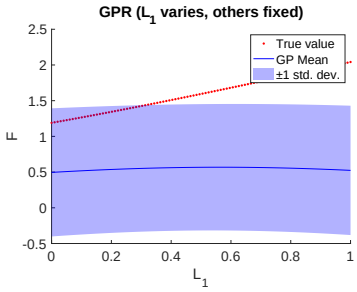
(a) L_1, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



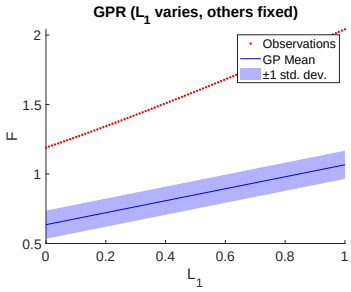
(b) L_1, Kernel = Linear, $N_{train} = N_{test} = 100$



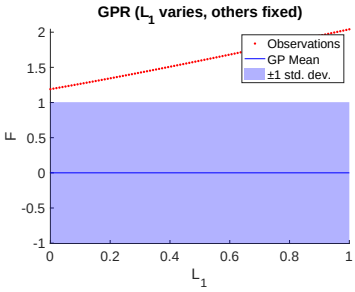
(c) L_1, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) L_1, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$

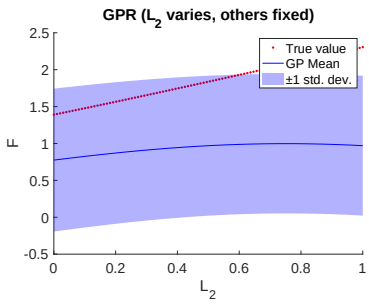


(e) L_1, Kernel = Linear, $N_{train} = N_{test} = 500$

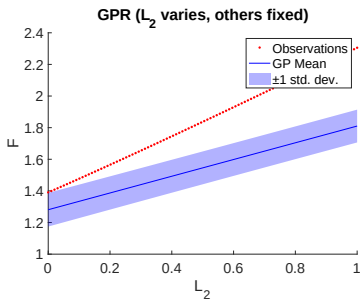


(f) L_1, Kernel = Indicator, $N_{train} = N_{test} = 500$

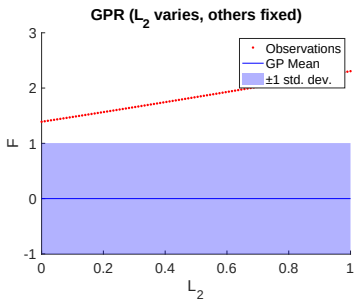
Figure 5: GPR estimation and uncertinity intervals of L_1 at $[\theta_1, \theta_2, \theta_3, \theta_4, L_2, L_3, L_4] = [0.7563, 5.7595, 6.0312, 6.2007, 0.2212, 0.7729, 0.2029]$



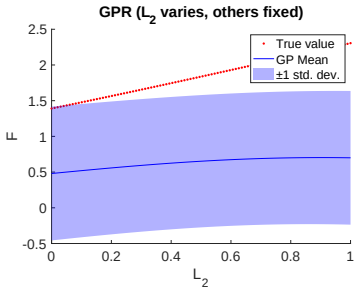
(a) L_2, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



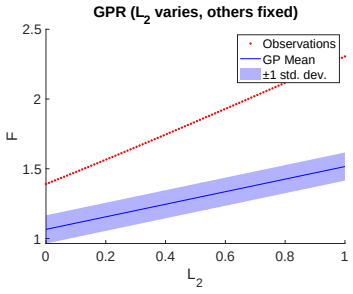
(b) L_2, Kernel = Linear, $N_{train} = N_{test} = 100$



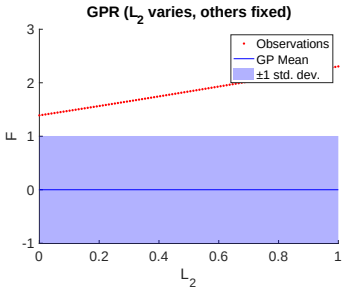
(c) L_2, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) L_2, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$

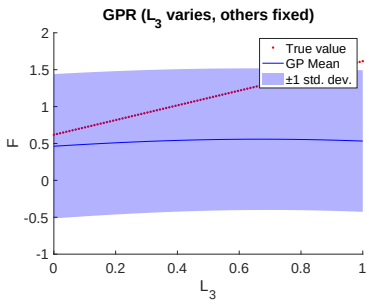


(e) L_2, Kernel = Linear, $N_{train} = N_{test} = 500$

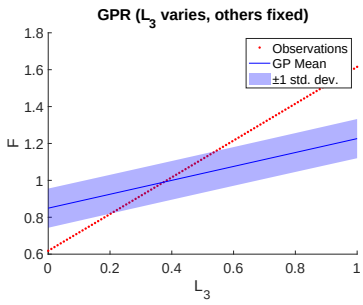


(f) L_2, Kernel = Indicator, $N_{train} = N_{test} = 500$

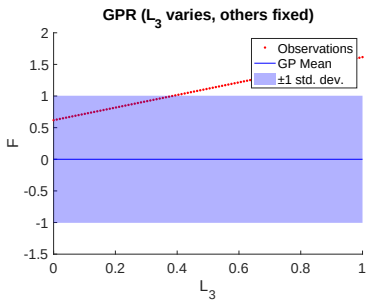
Figure 6: GPR estimation and uncertinity intervals of L_2 at $[\theta_1, \theta_2, \theta_3, \theta_4, L_1, L_3, L_4] = [2.9316, 6.1954, 6.1299, 1.9715, 0.8905, 0.4900, 0.7374]$



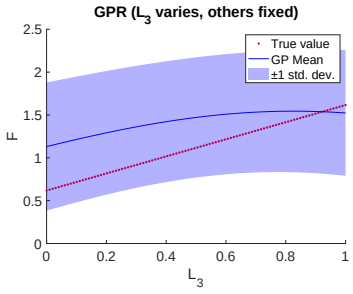
(a) L_3, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



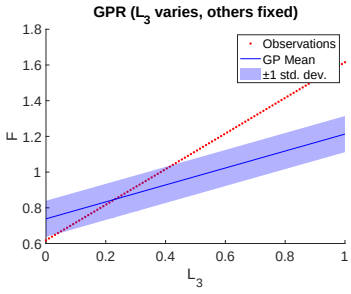
(b) L_3, Kernel = Linear, $N_{train} = N_{test} = 100$



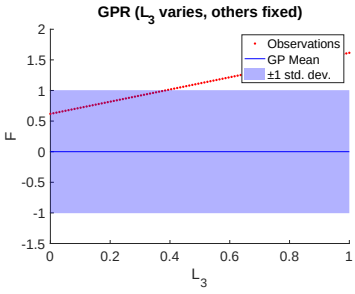
(c) L_3, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) L_3, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$

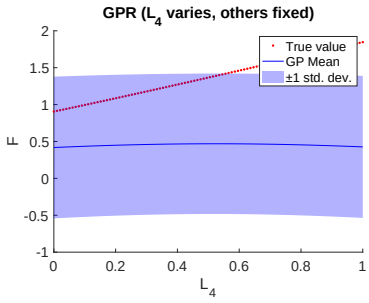


(e) L_3, Kernel = Linear, $N_{train} = N_{test} = 500$

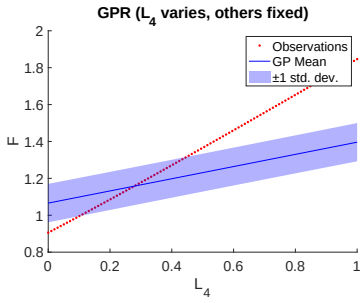


(f) L_3, Kernel = Indicator, $N_{train} = N_{test} = 500$

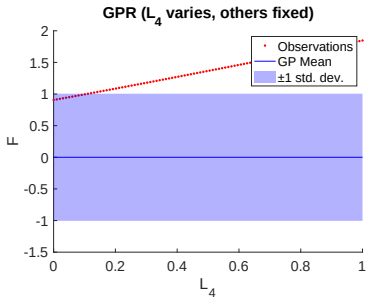
Figure 7: GPR estimation and uncertinity intervals of L_3 at $[\theta_1, \theta_2, \theta_3, \theta_4, L_1, L_2, L_4] = [5.3573, 3.9298, 0.7315, 5.1511, 0.6646, 0.8626, 0.0132]$



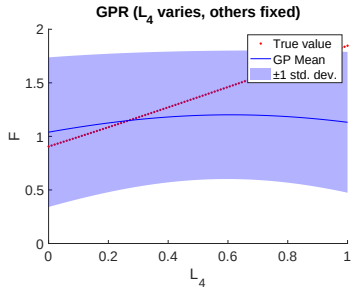
(a) L_4, Kernel = Squared Exponential, $N_{train} = N_{test} = 100$



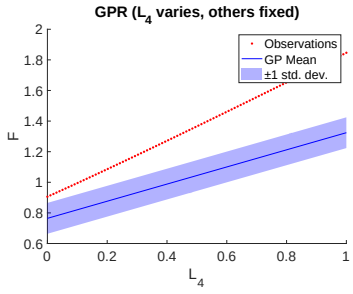
(b) L_4, Kernel = Linear, $N_{train} = N_{test} = 100$



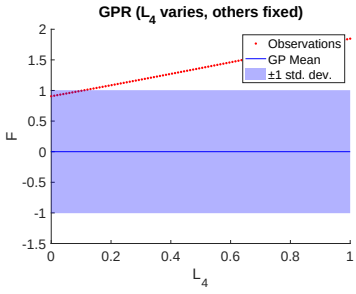
(c) L_4, Kernel = Indicator, $N_{train} = N_{test} = 100$



(d) L_4, Kernel = Squared Exponential, $N_{train} = N_{test} = 500$l

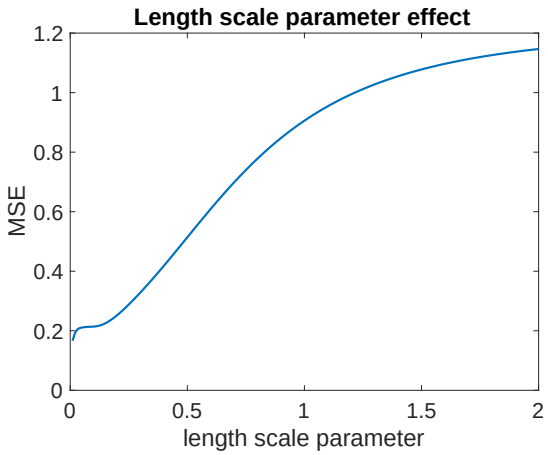


(e) L_4, Kernel = Linear, $N_{train} = N_{test} = 500$

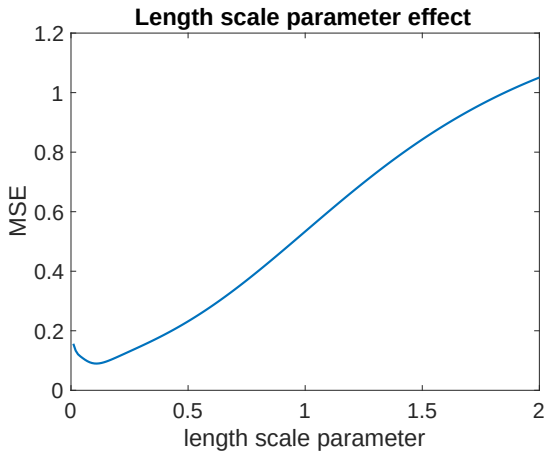


(f) L_4, Kernel = Indicator, $N_{train} = N_{test} = 500$

Figure 8: GPR estimation and uncertinity intervals of L_4 at $[\theta_1, \theta_2, \theta_3, \theta_4, L_1, L_2, L_3] = [3.7999, 5.0441, 4.5565, 2.9303, 0.9145, 0.4377, 0.2829]$



(a) $N_{train} = N_{test} = 100$



(b) $N_{train} = N_{test} = 500$

Figure 9: Effect of length scale parameter on the MSE of test data.