Department of Computer Science

Learning nonparametric individualized treatment response curves

Andrea Cognolato

Aalto University

MASTER'S THESIS

Aalto University
MASTER'S THESIS 2022

Learning nonparametric individualized treatment response curves

Andrea Cognolato

Otaniemi, 21 June 2022

Supervisor: professor Pekka Marttinen
Advisor: doctoral candidate Çağlar Hızlı

Aalto University
Department of Computer Science
School of Science

Author
Andrea Cognolato

Title
Learning nonparametric individualized treatment response curves

School School of Science

Supervisor professor Pekka Marttinen

Cosupervisor professor Mauro Gasparini

Advisor doctoral candidate Çağlar Hızlı

Level Master's thesis **Date** 21 June 2022 **Pages** 70 **Language** English

Abstract

Thanks to modern medical devices, clinicians are able to obtain accurate and frequent measurements of the patient's physiological state. Precision medicine aims to use the vast amount of data, stored in Electronic Health Records (EHRs), to individualize the treatment for each patient and design optimal treatment regimes. Learning individualized treatment responses accurately is an essential step to achieve the goals of precision medicine.

In the literature, the majority of treatment response methods use parametric functions to model the response curves. The functions are designed using domain knowledge about the clinical behavior of the treatment and make strong assumptions about the response curve's shape. Our goal is to develop a nonparametric model for treatment response curves that achieves competitive performance against parametric models while allowing patient-specific customizations.

We analyze the differences between directly modeling the treatment responses with a Gaussian Process (GP) and modeling the treatment dynamics using a Latent Force Model (LFM). We evaluate our models on a challenging blood glucose prediction dataset. Additionally, we use the treatment's covariates to scale the response curve model. We run experiments comparing two GP regression models as well as several ways of sharing the treatment response and treatment covariate model between patients. Our code and data are public for reproducibility and as a building block for future work. We obtain State-Of-The-Art (SOTA) performance on our dataset and discover that modeling the treatment dynamics with a LFM does not significantly improve the predictive performance.

Our results support the case for nonparametric models in treatment response curve estimation, and lay a solid foundation more sophisticated, GP-based methods. By providing better estimation of physiological states, we hope to empower clinicians and provide better, faster, and cheaper healthcare.

Keywords healthcare, treatment response, gaussian process, time series, machine learning

Contents

Abstract ii

Contents iii

1. Introduction 1

2. Background 4

2.1 Gaussian Processes 4

2.1.1 Definition 4

2.1.2 Covariance functions 5

2.1.3 Prediction 6

2.1.4 Marginal Likelihood 8

2.2 Multi-Output Gaussian Processes 8

2.2.1 Introduction 8

2.2.2 Intrinsic Coregionalization Model 10

2.3 Ordinary Differential Equations 11

2.3.1 Definition 12

2.3.2 Linear ODEs 12

2.3.3 Exact solutions for 1^{st}-order linear ODEs 12

2.4 Latent Force Models 13

2.4.1 Definition 13

2.4.2 Output kernel 14

2.4.3 Output-latent kernel 15

2.5 LFMs for Treatment Response Estimation 17

2.5.1 Model definition 18

2.5.2 Time-marked Latent Forces 18

2.5.3 Output kernel 18

2.5.4 Limitations 20

3. Problem Formulation **22**
3.1 Model . . . 22
3.2 Data . . . 23
3.3 Task . . . 24
3.4 Multiple individuals . . . 24

4. Methods **26**
4.1 Time-Limited Treatment Responses . . . 26
4.1.1 Model Definition . . . 26
4.1.2 Non-independent treatments . . . 27
4.1.3 Time-Limited Squared Exponential Kernel . . . 27
4.1.4 Limitations . . . 28
4.2 Time-Limited Latent Forces . . . 29
4.2.1 Model Definition . . . 29
4.2.2 Limitations . . . 30
4.3 Treatment Covariates . . . 31
4.3.1 Linear Scaling . . . 31
4.4 Individual-level Treatment Sharing . . . 32
4.4.1 Model definition . . . 32
4.4.2 ICM for Individual-level Treatment Sharing . . . 33
4.4.3 Kernel . . . 33
4.4.4 Hierarchical Linear Scaling Coefficients . . . 34

5. Experiments **38**
5.1 Simulated Data . . . 38
5.1.1 Dataset Generation . . . 38
5.1.2 Experiments . . . 39
5.1.3 Results . . . 39
5.2 Glucose Data . . . 43
5.2.1 Dataset . . . 43
5.2.2 Evaluation Setup and Metrics . . . 43
5.2.3 Experiments . . . 44
5.2.4 Results . . . 44

6. Discussion **50**
6.1 Summary of results . . . 50
6.2 Directions for the future . . . 51
6.3 Possible impact . . . 51

1. Introduction

The right to health and well-being is one of the fundamental human rights. Clinicians determine how to treat each patient by combining knowledge collected from the general population with data relative to the specific individual. Thus, data is crucial in helping healthcare professionals to provide the best possible treatment to patients.

The advent of electronic health records (EHR) and modern medical devices has led to a rapid increment of the amounts of medical data available. Clinicians are now able to obtain accurate and frequent measurements of the patient's physiological state, both in real-time and historical.

Using this vast amount of data, precision medicine seeks to improve the treatment for each patient by providing individualized treatment strategies. Personalized treatment is possible thanks to the increase in availability of longitudinal data, where the health state of an individual is available for long periods of time.

A key task in providing personalized treatments is learning individualized treatment response (ITR) curves. That is, estimating the continuous response over time of treatments from a time series of the patient's state. One example application of ITR curves is to develop optimal dosing strategies for medications.

Previous work models ITRs using parametric curves. These parametric curves are designed using expert knowledge about the physiological response to the treatment. The parametric models presented in the literature show wide variety in their formulation, using models such as Gaussian PDFs, mixtures of sigmoid functions, analytical solutions to LTI systems, and to Dirichlet Process Mixtures.

Various techniques from Bayesian nonparametrics have been used to model the baseline evolution of physiological covariates and, in some cases, part of the treatment response curves. But there are only few recent

examples of modeling the treatment response curve using a fully nonparametric approach. We claim that nonparametric models are crucial in achieving the goals of precision medicine, since their flexibility can capture individual-specific variations much better than a parametric model.

Our approach to solving the ITR estimation problems is to reimplement current state of the art nonparametrics ITR methods from the literature and to improve them by incorporating more data and more constraints into their formulation.

Following existing work on nonparametric ITR, we choose Gaussian Processes (GPs) as the model for the treatment response curves. The probabilistic foundations of GPs are crucial to obtain credible intervals for our predictions, in order to correctly estimate uncertainty. Additionally, we can easily customize the models and incorporate constraints by designing new covariance functions, or kernels.

To incorporate domain-specific knowledge about the physiological dynamics of treatments, we turn to Latent Force Models (LFMs). By combining the mechanistic approach of Ordinary Differential Equation (ODE) modeling with nonparametric GPs, LFMs allow having flexible models that will not make unrealistic predictions.

Current nonparametric models do not try to model continuous treatment dosages and only allow a finite number of treatment variants. We develop a method to include any kind of data on the treatment, which we call treatment covariates, in the prediction. This allows us to predict the effect of treatments with dosages never before seen in the training dataset.

Our methods are evaluated on their predictive performance for future treatments. The first set of experiments uses simulated data, in order to verify the correctness of our implementations. We use a real-world dataset of blood glucose measurements to evaluate our models on the challenging task of predicting the impact of meals on the glucose levels.

Our findings show that Gaussian Process-based nonparametric models can achieve satisfactory performance in ITR estimation task. Additionally we find that on noisy dataset that are typical in the healthcare field, more sophisticated models often fail because of their lower noise robustness features. In our experiments we find that simpler GP models are superior in terms of predictive performance, training time, and inference time to the more complex LFM models. Finally, we find that using treatment covariates to estimate the effect of unseen dosages greatly improves the predictive performance, but must be used carefully as it is very sensitive

to noise in the data.

We hope that the results we have showed can provide useful insights for researchers interested in using nonparametric methods for ITR estimation. On a broader level, our goal is to have a positive impact in the field of precision medicine. We believe that providing healthcare practitioners with better decision-making tools is crucial for improving the quality of health care and the quality of life of those who need it.

This thesis is structured as follows: Section 2 presents the needed background to develop the new methods and for the results, starting with Gaussian Processes and their Multi-Ouput extensions in Sections 2.1 and 2.2. A short introduction to Ordinary Differential Equations is given in Section 2.3. We proceed to merge GPs an ODEs in Section 2.4, which develops the formulation of Latent Force Models. The application of Latent Force Models to Treatment Response Curve estimation is presented in Section 2.5. Section 3 introduces the task we will use to evaluate our methods, the notation, and describes the available data. Section 4 presents the methods we develop, starting with the Time-Limited Squared Exponential Kernel in Section 4.1. We proceed to present a new LFM model, the Time-Limited Latent Force model in Section 4.2. Then we discuss how to introduce treatment covariates in Section 4.3 and finally how to extend our treatment models to multiple patients in Section 4.4. Section 5 presents the experiments and their results, which are divided by dataset into simulated data and real world data. To conclude, in Section 6 we summarize the results, discuss their significance and consider directions for future research.

2. Background

2.1 Gaussian Processes

2.1.1 Definition

A Gaussian Process (GP) is a random function. The evaluations of a GP at a finite number of points form a joint Gaussian distribution [7].

Just like a multivariate normal distribution is completely determined by its mean vector and covariance matrix, a GP is determined by its mean function m and covariance function k. We use the following notation to indicate a real-valued Gaussian process f:

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

where the two functions are defined as follows:

$$\begin{aligned} m &: \mathbb{R} \to \mathbb{R} \\ m(x) &= \mathbb{E}(f(x)) \\ k &: \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\ k(x, x') &= \mathbb{E}((f(x) - m(x))(f(x') - m(x'))) \end{aligned}$$

Under this definition, the GP is an infinite-dimensional object. This property is necessary to represent arbitrary functions. In our applications, we are mainly interested in evaluating the GP at a finite number of points. This allows us to work with multivariate normal distributions, a much simpler finite-dimensional object. We use the following notation to denote a GP evaluated in a finite set of points $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{f} = f(\mathbf{x})$$
$$\mathbf{f} \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$$

2.1.2 Covariance functions

The covariance function, also known as the *kernel*, is the most important factor in GP modeling. The function takes two points x_i, x_j as inputs, and is equal to the covariance between $f(x_i), f(x_j)$, the random variables obtained by evaluating the GP at each one of the two points.

Let $x_i, x_j \in \mathbb{R}$, then:

$$\text{cov}(f(x_i), f(x_j)) = k(x_i, x_j)$$

The behaviour of sampled functions heavily depends on the kernel choice. Let us now see how different kernels result in different samples. To see this, we pick an arbitrary set of test points $\mathbf{x}_* \in \mathbb{R}^{n_*}$. The mean and covariance functions are evaluated at the test points to create a mean vector $\boldsymbol{\mu} \in \mathbb{R}^{n_*}$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n_* \times n_*}$, respectively. Finally, we sample random vectors from the multivariate normal distribution that we defined and plot them.

$$\boldsymbol{\mu} = m(\mathbf{x}_*)$$
$$\boldsymbol{\Sigma} = k(\mathbf{x}_*,\mathbf{x}_*)$$
$$\mathbf{f}_* \sim \mathcal{N}(\mu, \boldsymbol{\Sigma})$$

The three kernels we will use to illustrate the differences are: squared exponential, periodic, white noise.

$$k(x, x') = \sigma^2 \exp\left(-\frac{1}{2}\frac{(x - x')^2}{\ell^2}\right)$$
$$k(x, x') = \sigma^2 \exp\left(-\frac{2}{\ell^2}\sin^2\left(\pi\frac{|x - x'|}{p}\right)\right)$$
$$k(x, x') = \begin{cases} \sigma^2, & \text{if } x = x' \\ 0, & \text{otherwise} \end{cases}$$

Figure 2.1 shows the covariance matrix and samples from each kernel.

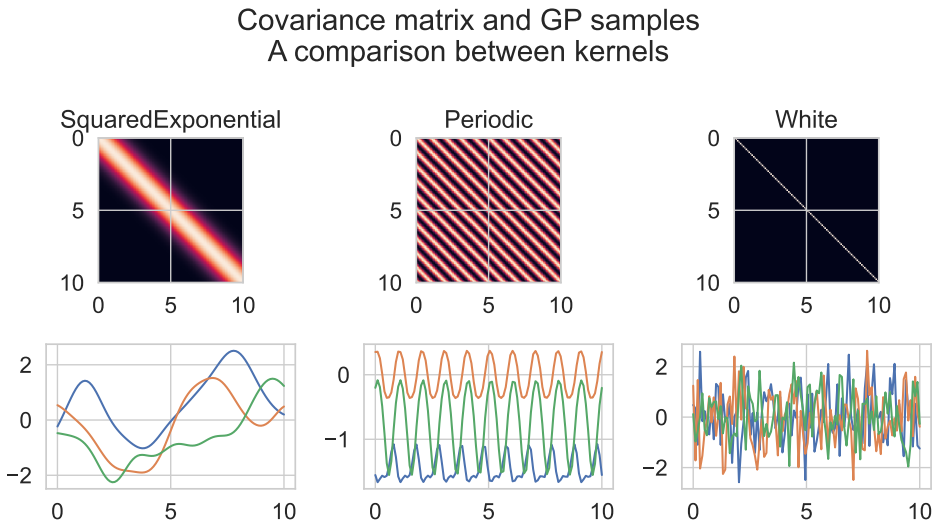


Figure 2.1. Comparison of the covariance matrix and samples from three zero-mean GPs with three different kernels. Left: A squared exponential kernel with lengthscale parameter $\ell = 1$. Center: A periodic squared-exponential kernel with lengthscale parameter $\ell = 1$ and period parameter $p = 1$. Right: White noise kernel. All kernels have scale parameter $\sigma = 1$.

2.1.3 Prediction

Having seen how a GP looks like *a priori*, that is without conditioning it on some data, let us now see how to incorporate observations. We start by writing the full joint distributions and then, using the conditioning property of multivariate Gaussians, we will obtain the GP posterior distribution.

The joint distribution over noiseless training outputs $\mathbf{f}$ and test outputs $\mathbf{f}_*$ is, when assuming zero-mean:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}, \mathbf{x}) & k(\mathbf{x}, \mathbf{x}_*) \\ k(\mathbf{x}_*, \mathbf{x}) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right),$$

where $\mathbf{x}$ are the training inputs, $\mathbf{x}_*$ the test inputs, and where by $k(\mathbf{x}, \mathbf{x}_*)$ we denote the $\mathbb{R}^{n \times n_*}$ matrix obtained by evaluating k on all pairs of training and test inputs.

Using the *conditioning* property of multivariate Gaussians [8], we can obtain a closed-form expression for the posterior distribution.

$$\begin{aligned} \boldsymbol{\mu_*} &= k(\mathbf{x}_*, \mathbf{x}) k(\mathbf{x}_*, \mathbf{x}_*)^{-1} \mathbf{f}, \\ \boldsymbol{\Sigma}_* &= k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{x}) k(\mathbf{x}_*, \mathbf{x}_*)^{-1} k(\mathbf{x}, \mathbf{x}_*), \\ \mathbf{f}_* \mid \mathbf{x}_*, \mathbf{f}, \mathbf{x} &\sim \mathcal{N}(\mu_*, \boldsymbol{\Sigma}_*). \end{aligned}$$

Using the same properties, we can obtain the posterior distribution given noisy observations $\mathbf{y} = \mathbf{f} + \epsilon$, where $\epsilon \in \mathbb{R}^n$. This assumes additive,

independent and identically distributed (i.i.d.) Gaussian noise. Thus, its covariance matrix will be, $\text{cov}(\epsilon, \epsilon) = \sigma_n^2 \mathbf{I}$. Rewriting the joint distributions of noisy observations and test outputs gives

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I} & k(\mathbf{x}, \mathbf{x}_*) \\ k(\mathbf{x}_*, \mathbf{x}) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right),$$

and the resulting posterior distribution is

$$\begin{aligned} \mu_* &= k(\mathbf{x}_*, \mathbf{x}) k(\mathbf{x}_*, \mathbf{x}_*)^{-1} \mathbf{y}, \\ \mathbf{\Sigma}_* &= k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{x})(k(\mathbf{x}_*, \mathbf{x}_*) + \sigma_n^2 \mathbf{I})^{-1} k(\mathbf{x}, \mathbf{x}_*), \\ \mathbf{f}_* \mid \mathbf{x}_*, \mathbf{y}, \mathbf{x} &\sim \mathcal{N}(\mu_*, \mathbf{\Sigma}_*). \end{aligned}$$

Finally, the posterior predictive distribution is obtained by simply adding $\sigma_n^2 \mathbf{I}$ to $\text{cov}(\mathbf{f}_*)$

$$\mathbf{y}_* \mid \mathbf{x}_*, \mathbf{y}, \mathbf{x} \sim \mathcal{N}(\mu, \mathbf{\Sigma}_* + \sigma_n^2 \mathbf{I})$$

In figure 2.2 we see an application of the concepts presented so far. We fit a GP model to some observations and show the posterior mean, some samples, as well as the 95% credible intervals.

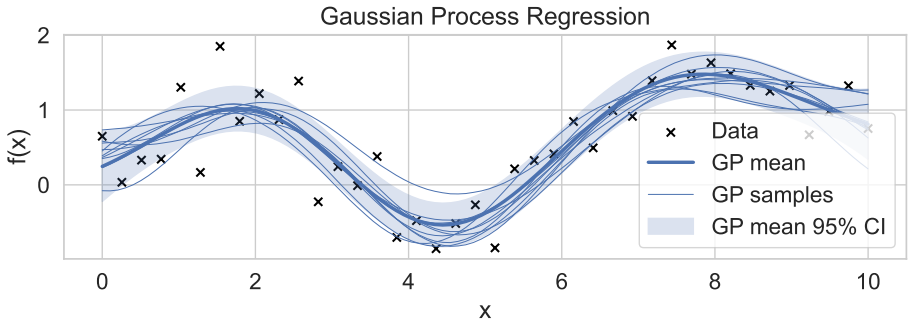


Figure 2.2. The chart shows an example of Gaussian Process Regression (GPR). The data, shown as black crosses, is generated by adding i.i.d. Gaussian noise with 0.5 standard deviation to the $\sin(x)$ function. A GP model with a squared exponential kernel is fitted and the three hyperparameters ℓ, σ, σ_n are estimated by MAP. We plot the posterior mean, posterior samples, and 2σ credible intervals with thick blue line, thin blue lines, and shaded blue regions, respectively.

2.1.4 Marginal Likelihood

Let us introduce the *marginal likelihood* $p(\mathbf{y})$. The marginal likelihood is likelihood $p(\mathbf{y} \mid \mathbf{f})$ integrated over the prior distribution $p(\mathbf{f})$.

$$p(\mathbf{y}) = \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f}) d\mathbf{f}$$

We call it marginal, since we are marginalizing or "integrating away" the function values of $\mathbf{f}$.

A closed-form expression for $p(\boldsymbol{y})$ can be derived by exploiting the fact that $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon}$, thus

$$\begin{aligned}
\mathbb{E}(\boldsymbol{y}) &= \mathbb{E}(\boldsymbol{f}) + \mathbb{E}(\boldsymbol{\epsilon}) = 0, \\
\mathbf{V}(\boldsymbol{y}) &= \mathbf{V}(\boldsymbol{f}) + \mathbf{V}(\boldsymbol{\epsilon}) = k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I}, \\
\boldsymbol{y} &\sim \mathcal{N}(0, k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I}),
\end{aligned}$$

Finally, we use the definition of log-likelihood and to obtain the formula

$$\log p(\mathbf{y}) = -\frac{1}{2}\mathbf{f}^T (k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I})^{-1} \mathbf{f} - \frac{1}{2} \log |k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}| - \frac{n}{2} \log 2\pi.$$

The marginal log likelihood can be efficiently computed by using the Cholesky decomposition instead of directly inverting the covariance matrix.

2.2 Multi-Output Gaussian Processes

2.2.1 Introduction

Until this point, our description of Gaussian Processes has focused on one-dimensional or real-valued GPs. Let us extend this definition to a larger class of models, *vector-valued* or multi-output GPs (MOGPs) [2].

Consider two independent GPs: $f_1(\cdot), f_2(\cdot)$. $f_1(\cdot)$ has zero mean and covariance function $k_1(\cdot, \cdot)$. $f_2(\cdot)$ has zero mean and covariance function $k_2(\cdot, \cdot)$.

$$\begin{aligned}
f_1(\cdot) &\sim \mathcal{GP}(0, k_1(\cdot, \cdot)) \\
f_2(\cdot) &\sim \mathcal{GP}(0, k_2(\cdot, \cdot))
\end{aligned}$$

Assume that we have an observation model with additive i.i.d. Gaussian

errors.

$$
\begin{aligned}
y_1 &= f_1 + \epsilon_1 \\
\epsilon_1 &\overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{n1}^2) \\
y_2 &= f_2 + \epsilon_2 \\
\epsilon_2 &\overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{n2}^2)
\end{aligned}
$$

and that we have two datasets of training input and observation pairs.

$$
\begin{aligned}
\mathbf{x}_1, \mathbf{y}_1 &\in \mathbb{R}^{N_1} \\
\mathbf{x}_2, \mathbf{y}_2 &\in \mathbb{R}^{N_2}
\end{aligned}
$$

We can then write the joint distribution

$$
\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k_1(\mathbf{x}_1, \mathbf{x}_1) + \sigma_{n1}^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_2(\mathbf{x}_2, \mathbf{x}_2) + \sigma_{n2}^2 \mathbf{I} \end{bmatrix} \right)
$$

Because the two GPs are independent, the covariance matrix is block-diagonal. In the general non-independent case, the matrix has nonzero upper right and lower left blocks.

To reinforce our intuition, see figure 2.3. In the left plot we display the joint covariance matrix for two independent GPs. In the right plot, we take one sample from the distribution defined by the covariance matrix on the left. That sample is a vector $\boldsymbol{f} \in \mathbb{R}^{N_1+N_2}$. We split the vector in its two components $\boldsymbol{f_1}, \boldsymbol{f_2}$ and plot them on top of each other.

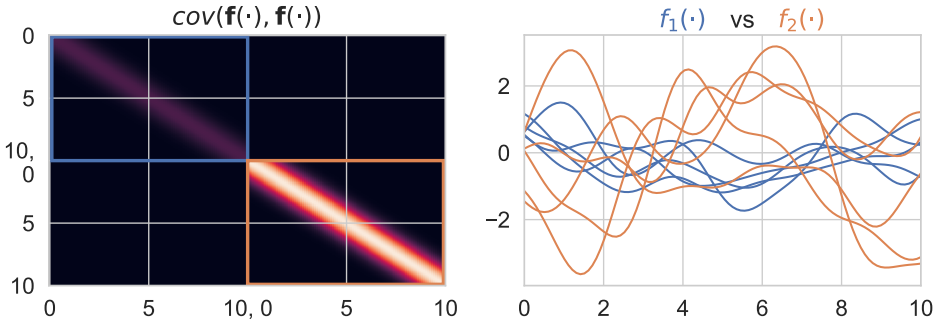


Figure 2.3. Left: the covariance matrix for the joint distribution of two independent one-dimensional GPs. For this plot, the two GPs use the same squared exponential kernel with the only difference being the scale parameters: $\sigma_1 = 0.5, \sigma_2 = 2.5$. Observe that this is a block-diagonal matrix. Right: samples from the zero-mean GPs using this covariance matrix. Notice how they samples show no signs of correlation.

2.2.2 Intrinsic Coregionalization Model

Instead of directly trying to define a covariance function for MOGPs, we are going to pick a generative model for our outputs and derive its corresponding covariance function.

We are going to keep the same assumptions as earlier but with one crucial modification. Let $\mathbf{a} \in \mathbb{R}^d$.

$$
\begin{aligned}
u(\cdot) &\sim \mathcal{GP}(0, k(\cdot, \cdot)) \\
\mathbf{f}(\cdot) = \mathbf{a}u(\cdot) &= \begin{bmatrix} f_1(\cdot) \\ \vdots \\ f_d(\cdot) \end{bmatrix} = \begin{bmatrix} a_1 u(\cdot) \\ \vdots \\ a_d u(\cdot) \end{bmatrix}
\end{aligned}
$$

This model is called *Intrinsic Coregionalization Model* ICM. In this model, we assume that all outputs are generated by linearly transforming an underlying GP $u(\cdot)$.

We now compute the multi-output kernel function for this multi-output GP. This is a function that, given the indices i, j of two GPs in $\boldsymbol{f}$ and given two locations x, x', will be equal to the covariance between $f_i(x)$, the i-th element of $\boldsymbol{f}$ evaluated at x, and $f_j(x')$, the j-th element of $\boldsymbol{f}$ evaluated at x'. Let $i, j \in 1, ..., d$, then

$$
\begin{aligned}
\mathbf{cov}(f_i(\cdot), f_j(\cdot)) &: \mathbb{N} \times \mathbb{R} \times \mathbb{N} \times \mathbb{R} \to \mathbb{R} \\
\mathbf{cov}(f_i(\cdot), f_j(\cdot)) &= a_i a_j \mathbf{cov}(u(\cdot), u(\cdot)) \\
&= a_i a_j k(\cdot, \cdot)
\end{aligned}
$$

Now that we defined the covariance between two arbitrary output, we can write in a single matrix expression the full multi-output covariance

between all pairs of outputs.

$$
\begin{aligned}
\text{cov}(\mathbf{f}(\cdot), \mathbf{f}(\cdot)) &= \begin{bmatrix} \text{cov}(f_1(\cdot), f_1(\cdot)) & \dots & \text{cov}(f_1(\cdot), f_d(\cdot)) \\ \vdots & \ddots & \vdots \\ \text{cov}(f_d(\cdot), f_1(\cdot)) & \dots & \text{cov}(f_d(\cdot), f_d(\cdot)) \end{bmatrix} k(\cdot, \cdot) \\
&= \begin{bmatrix} a_1 a_1 & \dots & a_1 a_d \\ \vdots & \ddots & \vdots \\ a_d a_1 & \dots & a_d a_d \end{bmatrix} k(\cdot, \cdot) \\
&= \begin{bmatrix} b_{11} & \dots & b_{1d} \\ \vdots & \ddots & \vdots \\ b_{d1} & \dots & b_{dd} \end{bmatrix} k(\cdot, \cdot) \\
&= \mathbf{a}\mathbf{a}^T k(\cdot, \cdot) \\
&= B k(\cdot, \cdot)
\end{aligned}
$$

where $B \in \mathbb{R}^{d \times d}$ is a matrix of rank 1.

We plot the covariance matrix and samples from it in figure 2.4, proceeding in a similar fashion as we did in figure 2.3.

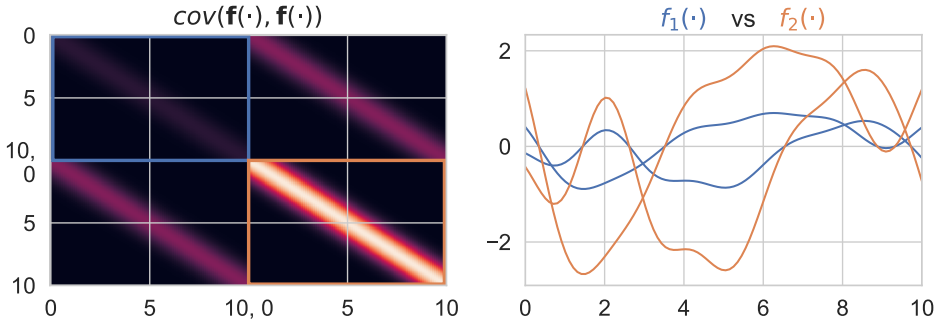


Figure 2.4. Left: the covariance matrix for the joint distribution of a Multi-Output Gaussian Process (MOGP) generated using the Intrinsic Coregionalization Model (ICM). For this plot, the two GPs use squared exponential kernels with scale parameters: $\sigma_1 = 0.5, \sigma_2 = 1.5$ and a common lengthscale $\ell = 1$. Observe that this matrix is has a block structure, but it is not block-diagonal. Right: samples from the zero-mean GPs using this covariance matrix. Notice how the samples from the 2$^{\text{nd}}$ GP $f_2(\cdot)$ are, as expected, simply the scaled version of the 1$^{\text{st}}$ GP $f_1(\cdot)$.

2.3 Ordinary Differential Equations

The majority of natural phenomenons involve change. To mathematically describe change, we must be able to write equations that relate varying quantities.

The derivative $y'(x)$ describes the rate of change of y with respect to x.

Hence, we will naturally want to write equations where a function and its derivative are related. We will call these differential equations.

In our treatment, we will only consider 1-dimensional, single-argument functions. This allows us to only consider *Ordinary Differential Equations* (ODEs).

2.3.1 Definition

Given G, a function of x, y, and the derivatives of y. Then an expression of the form

$$y^{(n)} = G(x, y, y', ..., y^{(n-1)})$$

is what we call an explicit ordinary differential equation of order n [1].

2.3.2 Linear ODEs

Let $a_i(x), f(x)$ be continuous functions of x. If the function G can be written in the following form:

$$y^{(n)} = G(x, y, y', ..., y^{(n-1)}) = \sum_{i=0}^{n-1} a_i(x)y^{(i)}(x) + f(x)$$

then we say that it is a *linear* ordinary differential equation.

The $f(x)$ term is called *forcing* or *source* term. If $f(x) = 0$ then we say that the ODE is *homogeneous*, otherwise we call it *inhomogeneous*.

2.3.3 Exact solutions for 1^{st}-order linear ODEs

Consider an equation of the form

$$y'(x) + ay(x) = f(x)$$

where $f(x)$ is a continuous functions of x, a a constant. We have a formula for the general solution

$$y(x) = \exp(-ax) \int \exp(ax) f(x) dx + c \exp(-at)$$

where c is an arbitrary real number.

To get a better intuition about the behaviour of this class of equations, figure 2.5 shows the forcing term and its effect on the solution of an inhomogeneous linear ODE with constant coefficients.

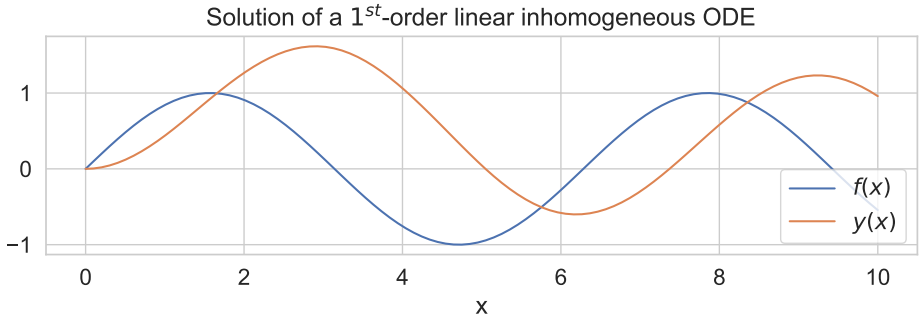


Figure 2.5. Solution of a 1st-order linear inhomogeneous ODE. The ODE parameters are $a = 0.15$, which we shall call the *decay* parameter, and $f(x) = \sin(x)$ which we call *forcing* function. We numerically solve the initial value problem with initial conditions $y(0) = 0$ and plot the solution $y(x)$ in orange and the forcing function $f(x)$ in blue.

2.4 Latent Force Models

Latent Force Models (LFMs) [5, 3] were introduced to bridge the gap between pure data-driven modeling and purely mechanistic modeling.

Data-driven techniques such as GPs and neural networks make weak assumptions about the underlying data generating process, thus "letting the data speak". In mechanistic modeling, the typical paradigm of physics, the models rely on existing physical laws combined with strong knowledge driven constraints, often expressed as differential equations.

It is natural to expect a range of models which vary in the strength of their mechanistic assumptions. Latent Force Models enrich GPs, a data-driven statistical model, with physics-inspired mechanistic ideas. To do so, LFMs incorporate differential equations into latent variable GP models. In our treatment, we will only consider first-order ordinary linear differential equations. However, in the literature, we see LFMs that use second-order linear ODEs [3], nonlinear ODEs [5, 6] and Partial Differential Equations (PDEs) [6].

2.4.1 Definition

We start by considering our mechanistic ODE model.

$$y'(x) + Dy(x) = B + Sf(x)$$

We have a first-order linear ordinary differential equation with constant coefficients and a nonzero forcing function.

The data-driven modeling aspect comes from the fact that we model the forcing function $f(x)$ using GPs. In this instance, we assume that the

latent force comes from a GP with zero-mean and squared exponential kernel with a length scale parameter ℓ.

$$k_{ff}(x, x') = \exp\left(-\frac{1}{2}\frac{(x - x')^2}{\ell^2}\right)$$
$$f(\cdot) \sim \mathcal{GP}(0, k_{ff}(\cdot, \cdot))$$

To become more familiar with this model, let us see in figure 2.6 some examples of its behaviour with different parameters B, D, S. Because this model was originally used to model gene transcription processes, the parameters names are: B basal rate, D decay rate, S sensitivity.

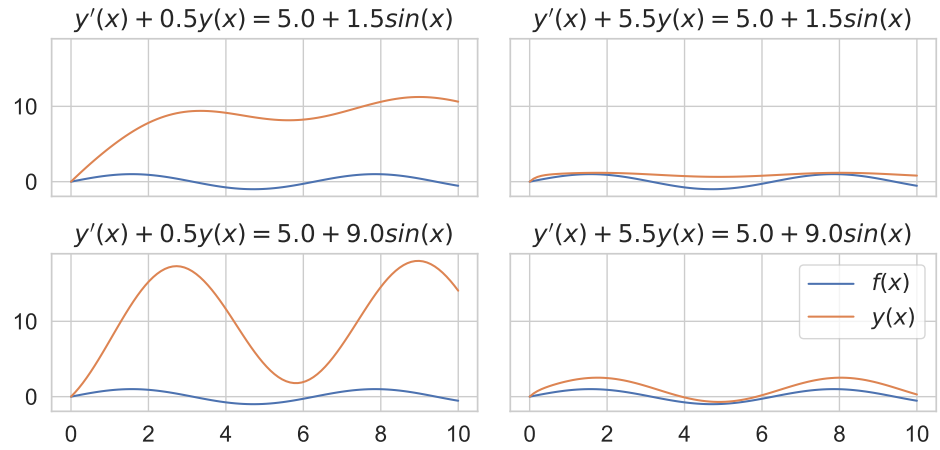


Figure 2.6. The four panels show the latent force/forcing term of the LFM (blue line) as well as the numerical solution of the ODE problem (orange line). We give the solver boundary conditions $y(0) = 0$. Each panel uses a different combination of parameters. Notice how increasing the decay rate D greatly reduces the average value of $y(x)$ while increasing the sensitivity S increases such value.

2.4.2 Output kernel

Using the results from the ODE chapter, we can write a closed-form expression for $y(x)$. To get rid of the arbitrary c factor, we must assume that $y(0) = B/D$.

$$y(x) = \frac{B}{D} + S\exp(-Dx)\int_0^x f(u)\exp(Du)du$$

In this model, the ODE's forcing function is a GP. Because the ODE is linear, its solution is a linear operator of the forcing function. The normal distribution is closed under linear operations, this also applies to GPs [7].

From this we can conclude that the ODE solution $y(\cdot)$ is also a GP.

$$y(\cdot) \sim \mathcal{GP}(0, k_{yy}(\cdot, \cdot))$$

Let us now compute the output kernel k_{yy} analytically.

$$\begin{aligned}
k_{yy}(x, x') &= \mathbf{cov}(y(x), y(x')) \\
&= \mathbb{E}((y(x) - B/D)(y(x') - B/D)) \\
&= \mathbb{E}((S\exp(-Dx)\int_0^x f(u)\exp(Du)du) \\
&\cdot (S\exp(-Dx')\int_0^{x'} f(u')\exp(Du')du')) \\
&= \mathbb{E}(S^2\exp(-D(x+x'))\int_0^x\int_0^{x'} f(u)f(u')\exp(D(u+u'))dudu') \\
&= S^2\exp(-D(x+x'))\int_0^x\int_0^{x'} \mathbb{E}(f(u)f(u'))\exp(D(u+u'))dudu' \\
&= S^2\exp(-D(x+x'))\int_0^x\int_0^{x'} k_{ff}(u,u'))\exp(D(u+u'))dudu'.
\end{aligned}$$

Substituting the definition of k_{ff} inside the double integral and using the properties of the error function allows us to obtain a closed-form expression.

$$k_{yy}(x, x') = S^2\frac{\sqrt{\pi}\ell}{2}[h(x, x') + h(x', x)]$$

where

$$\begin{aligned}
h(x', x') =& \frac{\exp(\gamma^2)}{2D}\{\exp\left[-D(x'-x)\right]\left[\mathbf{erf}\left(\frac{x'-x}{\ell} - \gamma\right) + \mathbf{erf}\left(\frac{x}{\ell} + \gamma\right)\right] \\
&- \exp\left[-D(x'-x)\right]\left[\mathbf{erf}\left(\frac{x'}{\ell} - \gamma\right) + \mathbf{erf}\left(\gamma\right)\right]\}.
\end{aligned}$$

Here $\mathbf{erf}(x) = \int_0^x \exp(-u^2)du$ and $\gamma = D\ell/2$.

Looking at figure 2.7, we see how the covariance matrix of this kernel looks like. In this instance, the kernel uses parameters $B = 0, D = 0.5, S = 1.5, \ell = 1.5$. Additionally, we take samples from a GP using this kernel and can indeed verify that they look similar the ODE solutions we see in figure 2.6.

2.4.3 Output-latent kernel

To infer the latent forces that are responsible for the output's behaviour we also need an "output-latent" kernel that computes the cross-covariance between the output $y(x)$ and the latent force $f(x')$.

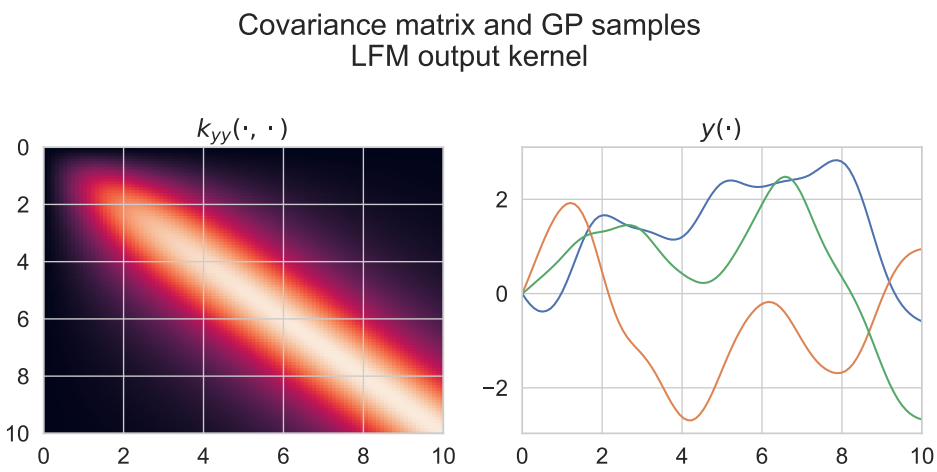


Figure 2.7. Left: Covariance matrix of a GP with a LFM output kernel. In this chart, the kernel uses parameters $B = 0, D = 0.5, S = 1.5, \ell = 1.5$. Observe how, in the top-left corner, all values are very close to zero. Right: samples from the zero-mean GPs using this covariance matrix. Notice how, near zero, all of the samples have small values. This is because of the initial condition $y(0) = B/D = 0$ which we have used to derive the kernel's formula.

The derivation for the output-latent kernel follows the sames steps as the output kernel:

$$\begin{aligned}
k_{yf}(x, x') &= \text{cov}(y(x), f(x')) \\
&= \mathbb{E}((y(x) - B/D)f(x')) \\
&= \mathbb{E}((S\exp(-Dx)\int_0^x f(u)\exp(Du)du)f(x')) \\
&= \mathbb{E}((S\exp(-Dx)\int_0^x f(u)f(x')\exp(Du)du)) \\
&= S\exp(-Dx)\int_0^x \mathbb{E}(f(u)f(x'))\exp(Du)du \\
&= S\exp(-Dx)\int_0^x k_{ff}(u, x')\exp(Du)du.
\end{aligned}$$

Again, for squared exponential kernels this can be obtained explicitly leading to

$$k_{yf}(x, x') = \frac{\sqrt{\pi}\ell S}{2}\exp(\gamma^2)\exp(-D(x' - x))\left[\text{erf}\left(\frac{x' - x}{\ell} - \gamma\right) + \text{erf}\left(\frac{x}{\ell} + \gamma\right)\right].$$

In figure 2.8 we see Gaussian Process Regression (GPR) with a GP that used the covariance function that we have derived. In the left panels we plot the data using black crosses as well as the posterior mean, and 2σ credible intervals with thick blue line, and shaded blue regions, respectively.

The right panel uses the cross-covariance function to estimate the latent force posterior from data. The latent force posterior is also a GP with the

following mean and covariance:

$$\mu_* = k_{fy}(\mathbf{x}_*, \mathbf{x}) k_{yy}(\mathbf{x}_*, \mathbf{x}_*)^{-1} \mathbf{y},$$
$$\mathbf{\Sigma}_* = k_{ff}(\mathbf{x}_*, \mathbf{x}_*) - k_{fy}(\mathbf{x}_*, \mathbf{x})(k_{yy}(\mathbf{x}_*, \mathbf{x}_*) + \sigma_n^2 \mathbf{I})^{-1} k_{yf}(\mathbf{x}, \mathbf{x}_*),$$
$$\mathbf{f}_* \mid \mathbf{x}_*, \mathbf{y}, \mathbf{x} \sim \mathcal{N}(\mu_*, \mathbf{\Sigma}_*).$$

The posterior mean (solid orange line) and its 2σ credible interval (shaded orange region) are plotted. Additionally, we plot the true latent force $\sin(x)$ and verify that it is always inside the shaded interval.

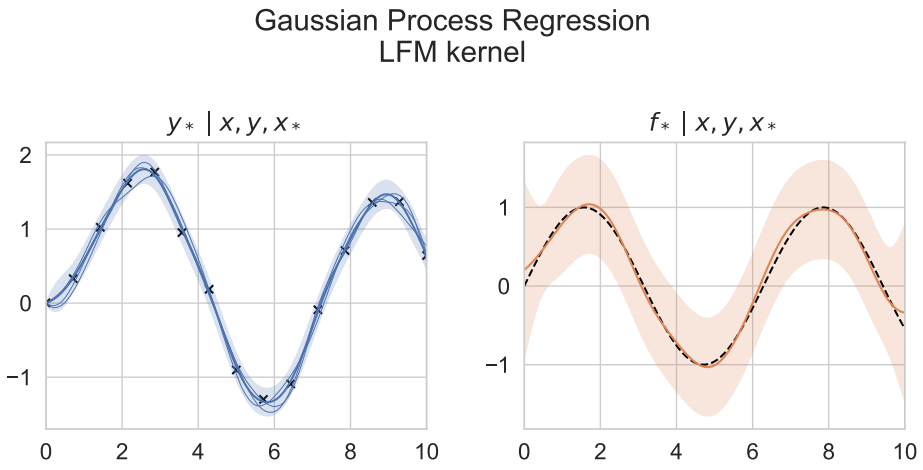


Figure 2.8. The chart shows an example of Gaussian Process Regression (GPR) using a GP kernel based on a Latent Force Model. The data, shown as black crosses, is generated by giving the $\sin(x)$ function to an ODE solver that solves the LFM ODE problem. Then, to the resulting ODE solution, we add i.i.d. Gaussian noise with 0.5 standard deviation. A GP model with a LFM kernel is fitted. Left: The data is plotted using black crosses. We plot the posterior mean, and 2σ credible intervals with thick blue line, and shaded blue regions, respectively. Right: We plot the estimated latent force (solid orange line) and its 2σ credible interval (shaded orange region). We plot the true latent force $\sin(x)$ and verify that it is always inside the shaded interval.

2.5 LFMs for Treatment Response Estimation

Up to this point we have introduced Gaussian Processes and Linear Ordinary Differential Equations. These two concepts were then merged together to develop Latent Force Models. We have motivated LFMs as a way to mix data-driven, weakly mechanistic models like GPs and knowledge-drive, stronly mechanistic models like ODEs. We will now see a real-world application of LFMs, as we will use them to model treatment response curves.

We will present a simplified version of [4]. We limit ourselves to discussing the model for treatment responses, without introducing the extra complexity required to model the baseline signal and without discussing

how a hierarchical structure can be used to improve the prediction across multiple individuals.

2.5.1 Model definition

Let τ be the time. Let $y(\tau)$ be the physiological quantity we are interested in modeling. Let $\mathbf{t} = \{t_m \mid m = 1, \dots, M\}$ be a set of *treatment times* i.e. the time at which a specific treatment was administered to the patient.

We write the ODE part of the LFM describing the time evolution of the physiological quantity

$$y'(\tau) = B - Dy(\tau) + S \sum_{m=1}^{M} f(\tau; t_m)$$

2.5.2 Time-marked Latent Forces

Again, the forcing function $f(\tau; t_m)$ is a GP. But, since our goal is to model treatments, we must add one crucial condition. The effect of the treatment must be constant before the treatment time t_m. To model this we turn to *time-marked* or *causal* GPs.

$$k_{ff}(\tau, \tau'; t_m) = \exp\left\{-\frac{[h(\tau - t_m) - h(\tau' - t_m)]^2}{\ell^2}\right\}$$
$$f(\tau; t_m) \sim \mathcal{GP}(0, k_{ff}(\cdot, \cdot; t_m)$$

where $h(\tau) = \tau \mathcal{I}(\tau > 0)$ is the clipping function that enforces causality by cancelling any effect before the treatment time.

For clarity, let us now compare in figure 2.9 the covariance matrices and samples of two GPs using the standard squared exponential kernel and the time-marked squared exponential kernel, respectively.

2.5.3 Output kernel

Just like we did in the LFM section, we can write the analytical solution to the ODE. Again, we must assume that $y(0) = B/D$.

$$y(\tau) = \frac{B}{D} + S \sum_{m=1}^{M} \exp(-D\tau) \int_0^\tau f(u; t_m) \exp(Du) du$$

We can now proceed and compute the analytical expression for the output

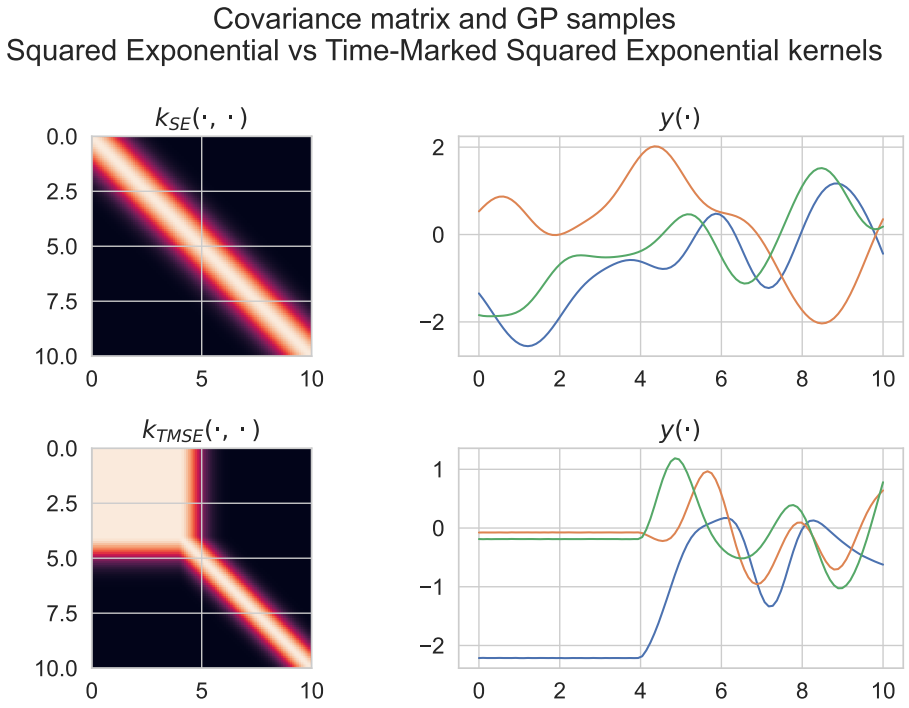


Figure 2.9. Top left: covariance matrix for a squared exponential (SE) kernel with length scale $\ell = 1$. Top right: samples from a GP with zero mean and SE kernel. Bottom left: covariance matrix for a time-marked squared exponential (TMSE) kernel with length scale $\ell = 1$ and treatment time $t_m = 4$. Top right: samples from a GP with zero mean and TMSE kernel.

kernel Let us now compute the output kernel k_{yy} analytically

$$
\begin{aligned}
k_{yy}(\tau,\tau') =& \mathbf{cov}(y(\tau), y(\tau')) \\
=& \mathbb{E}((y(\tau) - B/D)(y(\tau') - B/D)) \\
=& \mathbb{E}((S\sum_{m=1}^{M}\exp(-D\tau)\int_0^{\tau} f(u;t_m)\exp(Du)du) \\
& \cdot (S\sum_{m'=1}^{M}\exp(-D\tau')\int_0^{\tau'} f(u';t'_m)\exp(Du')du')) \\
=& \mathbb{E}(S^2\exp(-D(\tau+\tau')) \\
& \cdot \sum_{m=1}^{M}\sum_{m'=1}^{M}\int_0^{\tau}\int_0^{\tau'} f(u;t_m)f(u';t_m)\exp(D(u+u'))dudu') \\
=& S^2\exp(-D(\tau+\tau')) \\
& \cdot \int_0^{\tau}\int_0^{\tau'} \mathbb{E}(\sum_{m=1}^{M}\sum_{m'=1}^{M} f(u;t_m)f(u';t'_m))\exp(D(u+u'))dudu' \\
=& S^2\exp(-D(\tau+\tau')) \\
& \cdot \int_0^{\tau}\int_0^{\tau'} \sum_{m=1}^{M} k_{ff}(u,u';t_m)\exp(D(u+u'))dudu'
\end{aligned}
$$

We have obtained a formulation similar to the single-force LFM. There are

two crucial differences that prohibit us from re-using the same analytical expression as before. First, k_{ff} is now a time-marked squared-exponential. Second, and most crucially, when simplifying the double summation over m, m' to a single one over m we have implicitly assumed independence between two treatments $m \neq m'$. We will revisit this assumption is our methods section.

While it is possible to obtain a closed-form solution for this kernel, we will neither present the steps nor the final formulas here. The same goes for the "cross-covariance" output-latent kernel. For more details check [4]'s supplementary material or the code samples provided with this document.

In figure 2.10 we can see the result of our efforts. The treatment response shows exponential growth/decay before the treatment time t_m, due to the constant value of the latent force, and shows SE-like behaviour after.

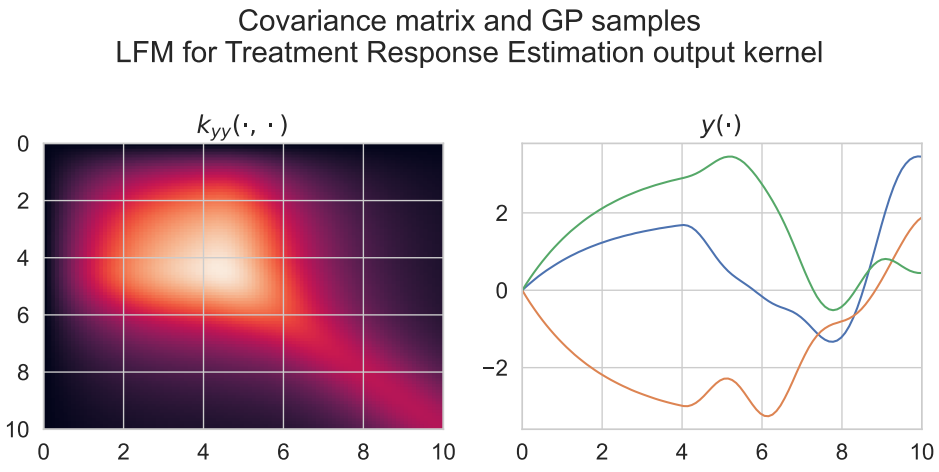


Figure 2.10. Left: covariance matrix for the k_{yy} output kernel with parameters $B = 0, D = 0.5, S = 1.5, t_m = 4, \ell = 1$. Right: samples from the output kernel.

2.5.4 Limitations

Let us now discuss some limitations of this model, considering our goals of using it to predict the effect of treatments on physiological quantities. The main issues that we have found, when trying to apply this to a real-world dataset are four

- Treatments are nonzero (but constant) before the treatment time.

- Treatments have infinite duration. The effect persists long after the treatment time. This is not a realistic assumption if our goal is to model real-world drug effects.

- Treatments are independent. This is not realistic as it is reasonable to assume that the same drug, taken with the same dosage will have very similar effects regardless of the administration time.

- The dosage or, more generally, the treatment's covariates have no effect on the treatment response.

3. Problem Formulation

We address the problem of treatment response curve estimation. Our goal is to estimate the effects of a treatment or intervention on a physiological quantity. This effect is modeled as a continuous function of time and, optionally, of the treatment's covariates. Estimating a treatment response curve is used to predict the state of a patient under the administration of drugs and other therapeutic interventions.

In this section, we will introduce the high-level components of the models used in the methods and results sections, describe the dataset used to train the models, and the details of the task.

3.1 Model

Let $\hat{y}(\tau)$ be a physiological quantity of an individual.
Let $f_b(\tau)$ be a continuous function that models the baseline state, that is, the state without any treatments or interventions. In the literature, this is also called *counterfactual trend*.
Let $f_t(\tau; t_m)$ be a continuous function that models the treatment response to a treatment administered at time t_m. This function is the main focus of this dissertation and it is known in the literature as the *treatment response curve*. Different treatment response curve models will have different f_t formulations.
The generative model we have chosen to model the physiological quantity is

$$\hat{y}(\tau) = f_b(\tau) + \sum_{t_m} f_t(\tau; t_m)$$

The treatment effects are additive both within themselves and with the baseline function.
Let $y(\tau)$ be the noisy physiological quantity. The noise model is zero-mean

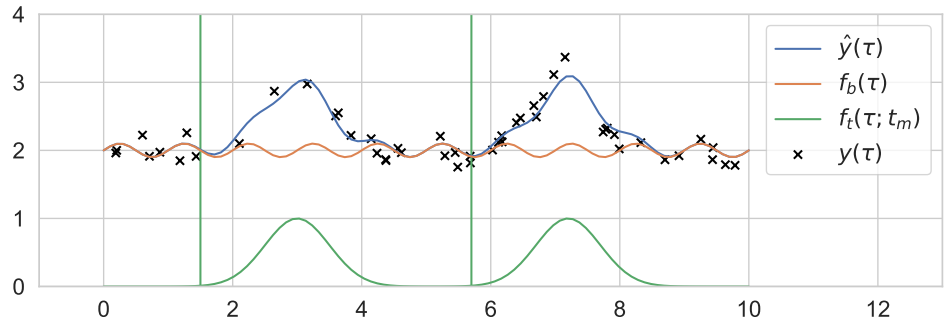


Figure 3.1. Illustration of the distinct functions in the model and how they are combined to form the final prediction. In this example, the physiological quantity $\hat{y}(\tau)$ is the sum of a sinusoidal baseline $f_b(\tau) = \sin(2\pi\tau)$ and treatment response curves $f_t(\tau; t_m) = \exp\{-0.5(\tau - t_m - 2)^2/0.5^2\}$. Two treatments are applied at $t_1 = 1.5$ and $t_2 = 5.7$. Finally, $y(\tau)$ is sampled in 50 points uniformly distributed in $[0, 40]$, after having added i.i.d. Gaussian noise with standard deviation $\sigma = 0.15$.

independent and identically distributed Gaussian noise.

$$y(\tau) = \hat{y}(\tau) + \epsilon(\tau)$$
$$\epsilon(\tau) \sim \mathcal{N}(0, \sigma^2_{\text{obs}})$$

There might be extra data associated with the treatments. We shall call it *treatment covariates* and denote it with $\mathbf{x}_m$.

Figure 3.1 illustrates how all of the aforementioned functions come together to form the model for treatment effects.

3.2 Data

Let $\tau = \{\tau_i \mid i = 1, 2, \ldots, N\}$ be irregularly-sampled times, sorted temporally.

Let $\mathbf{y} = \{y_i \mid \tau = 1, 2, \ldots, N\}$ be noisy observations of the physiological quantity on we wish to estimate the effect of a treatment on. Every observation y_i was performed at a time τ_i.

Let $\mathbf{t} = \{t_m \mid m = 1, 2, \ldots, M\}$ be irregularly-sampled times at which a treatment is administered. Notice that generally $N \neq M$, i.e. the observations and treatments are not necessarily aligned.

Let $\mathbf{x} = \{\mathbf{x}_m \mid m = 1, 2, \ldots, M\}$ be treatment covariates associated with every treatment, e.g. a drug's dosage, the amount of carbohydrates, proteins, and fats in a meal.

Our complete dataset $\mathcal{D}$ is thus a 4-tuple $\mathcal{D} = \{\tau, \mathbf{y}, \mathbf{t}, \mathbf{x}\}$.

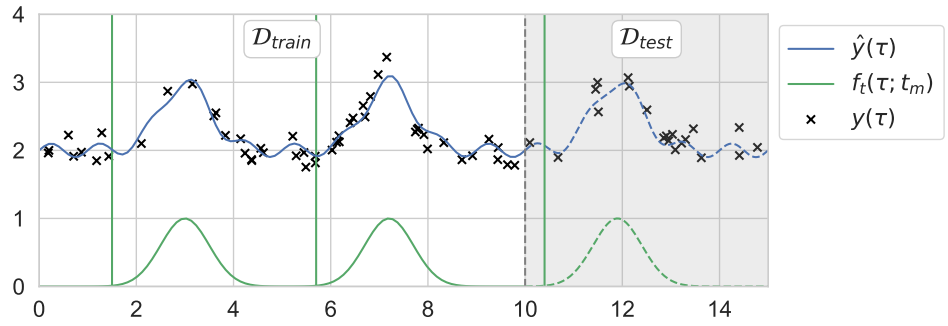


Figure 3.2. Illustration of how the model is trained on the training dataset $\mathcal{D}_{\text{train}}$, and then tested using the test dataset $\mathcal{D}_{\text{test}}$. The only difference between the two datasets is that in the test dataset we do not have access to $\mathbf{y}_{\text{test}}$. All of the other values: time, treatment times, and treatment covariates are available.

3.3 Task

Our goal is to estimate the values of $\hat{y}(\tau)$, the noiseless physiological quantity, for any future time point τ_{test}, given new treatment times $\mathbf{t}_{\text{test}}$ and treatment covariates $\mathbf{x}_{\text{test}}$. We have at our disposal an historical dataset $\mathcal{D}_{\text{train}} = \{\tau_{,\text{train}}\, \mathbf{y}_{\text{train}}, \mathbf{t}_{\text{train}}, \mathbf{x}_{\text{train}}\}$. The estimate $\hat{y}(\tau)$ is then evaluated at τ_{test} and compared with $\mathbf{y}_{\text{test}}$, the true noisy observations at τ_{test}. In figure 3.2 we show the data used to train the model as well as the data used in the prediction task the model is evaluated on.

3.4 Multiple individuals

So far, we have formulated the problem as if we had data about a single individual. This formulation could also have worked with multiple individuals, as long as they are treated as completely independent between each other, in a so-called *unpooled* fashion.

Motivated by the fact that we will want to share data across multiple individuals, we now introduce the notation for multiple individuals
Let $P \in \mathbb{N}$ be the number of individuals in our complete dataset.
Let $p = 1, 2, \ldots, P$ be the index of a specific individual.
Let $N^{(p)}, M^{(p)} \in \mathbb{N}$ be number of observations and the number of treatments for a specific individuals, respectively.
Then $\tau^{(p)} \in \mathbb{R}^{N^{(p)}}$ are the times for individual p's observations $y^{(p)} \in \mathbb{R}^{N^{(p)}}$ the noisy observations. $t^{(p)} \in \mathbb{R}^{M^{(p)}}$ be the treatment times. $\mathbf{x}^{(p)} \in \mathbb{R}^{M^{(p)}}$ be the treatment covariates.

Notice how, in general $N^{(p)} \neq N^{(p')}$. This means that the number of observations can vary vastly between two different individuals and that

they are not necessarily temporally aligned.

We also define the generative model for the p-th individual, assuming no sharing of functions.

$$\hat{y}^{(p)}(\tau) = f_b^{(p)}(\tau)^{(p)} + \sum_{t_m} f_t^{(p)}(\tau; t_m)$$

This could also be rewritten using the formalism of Multi-Output Gaussian Processes, as we will see in later chapters.

4. Methods

4.1 Time-Limited Treatment Responses

Here, we propose our first model for learning nonparametric treatment response curves. Unlike the subsequent models, this model does not use ODEs to model the dynamics of physiological quantities. Instead, the treatment responses are modelled directly using GPs with a modified squared exponential kernel.

4.1.1 Model Definition

We start from the generative model defined in chapter 3.

$$\hat{y}(\tau) = f_b(\tau) + \sum_{t_m} f_t(\tau; t_m)$$

The baseline function f_b is a constant, motivated by the fact that in the dataset we are interested in modeling we see no clear patterns outside of the ones explained by treatments.

$$f_b(\tau) = k$$

The treatment response function f_t is a zero-mean GP with a custom kernel k_{ff}.

$$f_t(\tau; t_m) \sim \mathcal{GP}(0, k_{ff}(\cdot, \cdot; t_m))$$

4.1.2 Non-independent treatments

Unlike the approaches we presented so far, we do not assume that the treatment are independent between each other. On the other hand, we assume that for all $t_m \in \mathbf{t}$, the treatment response is the same sample from the GP. Mathematically speaking, if we assume that treatments are independent

$$\text{cov}(f_t(\tau; t_m), f_t(\tau'; t_{m'})) = \begin{cases} k(\tau, \tau'; t_m), & \text{if } m = m' \\ 0, & \text{otherwise} \end{cases}$$

whereas, in our non-independent formulation

$$\text{cov}(f_t(\tau; t_m), f_t(\tau; t_{m'})) = k(\tau, \tau'; t_m, t_{m'}).$$

4.1.3 Time-Limited Squared Exponential Kernel

Because the goal of the treatment response f_t is to model the effects of some drug, a regular squared exponential (SE) kernel is not enough. The main issue is that a SE kernel generates functions that are not time-limited. On the other hand, we expect the treatment effect to have a starting time and a finite duration. To model this, we modify the SE kernel to generate functions that are zero before t_m and zero again after $t_m + T$, where T is the treatment's duration.

To design a kernel that produces zero-valued functions we take a regular SE kernel and then set its value to zero whenever $\tau, \tau' < t_m$ or $\tau, \tau' > t_m + T$. We shall call this *Time-Limited Squared Exponential* kernel (TLSE).

$$k_{\text{SE}}(\tau, \tau') = \sigma^2 \exp\left\{-\frac{1}{2}\frac{(\tau - \tau')^2}{\ell^2}\right\}$$

$$k_{\text{TLSE}}(\tau, \tau'; t_m) \overset{\text{def}}{=} \begin{cases} k_{\text{SE}}(\tau - t_m, \tau' - t_m) & \text{if } t_m < \tau, \tau' < t_m + T \\ 0 & \text{otherwise} \end{cases}$$

The resulting covariance matrices and GP samples can be compared in figure 4.1.

This "trick" works for any base kernel and relies on the definition of a constant kernel.

$$k_{\text{CONST}}(\tau, \tau') = \sigma^2$$

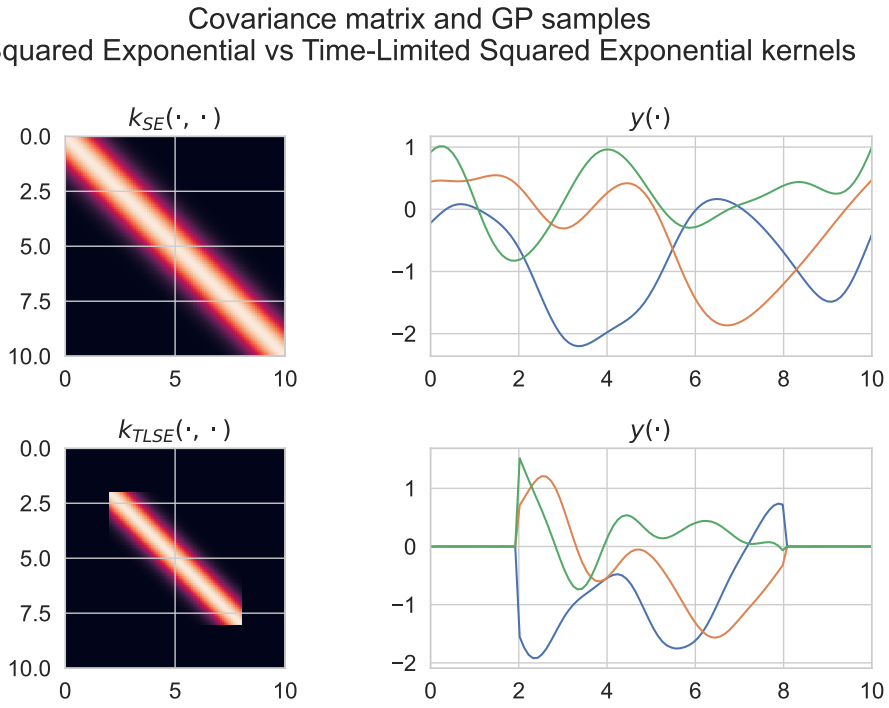


Figure 4.1. Comparing the covariance matrices and 3 GP samples for a Squared Exponential (SE) kernel and Time-Limited Squared Exponential (TLSE) kernel. Both kernels share the same length scale $\ell = 1$. This TLSE kernel is using a treatment time $t_m = 2$ and treatment duration $T = 6$, so we expect its effect to end at $t_m + T = 2 + 6 = 8$. Notice the discontinuities near the times where the treatment starts and stops its effects. In practice they do not appear to be problematic.

This kernel will produce random constant functions (i.e. horizontal lines) with values $k \sim \mathcal{N}(0, \sigma^2)$. By setting $\sigma^2 = 0$ the normal distribution collapses and we will always get functions with the value 0.

4.1.4 Limitations

This is the simplest model that has been developed as part of this dissertation. While its simplicity is an attractive factor, mainly because of how it benefits interpretability and training performance, it is also a source of several downsides. The main limitations we have observed are twofold:

- No knowledge of treatment covariates. Since this model does not use any data from the treatment covariates, it will predict the same effect for two vastly different dosages. This is especially an issue when modeling the effect of meals, which can vary greatly in their caloric content.

- No explicit model for treatment dynamics. On the spectrum of weakly mechanistic to strongly mechanistic models, this one clearly is closer to the former. The only knowledge-driven modeling aspect we have used is

the kernel design. We would like to understand whether it is reasonable to introduce additional knowledge of the treatment dynamics through a Latent Force Model.

4.2 Time-Limited Latent Forces

The goal of our second model is use our knowledge about the treatment dynamics for better treatment response estimation. We build on previous work on Latent Force Models (LFMs) for Treatment Response Estimation, with some crucial modifications that we believe are required for adequate modeling of treatment.

Our main contribution is to update the existing SOTA [4] model to include the two constraints developed in our first model.

4.2.1 Model Definition

Starting from our common generative model

$$\hat{y}(\tau) = f_b(\tau) + \sum_{t_m} f_t(\tau; t_m)$$

we decide to use a constant baseline function $f_b(\tau) = k$.

The treatment response function f_t comes directly from the LFM model. The notation is slightly different, as the physiological quantity y is now f_t, and the latent force f is now f_l. Additionally, we consider the basal rate term B to be zero, as it does not contribute to the output kernel and can be modelled through the mean function.

$$f_t'(\tau; t_m) = -Df_t(\tau) + Sf_l(\tau; t_m);$$

The latent function f_l is modelled with a GP. Since this function describes the underlying "effect" of a treatment, our previous arguments about causality and time-limitedness still apply. For these reasons, we do not use a squared exponential kernel like [3] or a time-marked squared exponential kernel [4], but instead our newly developed time-limited squared exponential kernel (TLSE).

$$f_l(\cdot; t_m) \sim \mathcal{GP}(0, k_{\text{TLSE}}(\cdot, \cdot; t_m))$$

Again, since we are using a linear ODE its solution is a linear functional L of the forcing function. And since the forcing function is a GP, any linear combination will still be a GP, albeit with a different covariance function.

We can write the analytical solution to the ODE

$$\begin{aligned} f_t(\tau; t_m) &= S \exp(-D\tau) \int_0^\tau f_l(u; t_m) \exp(Du) du \\ &= L[f_l(\cdot; t_m)](\tau) \end{aligned}$$

Finally, similar to our first model and unlike the LFM models found in the literature, we assume that all treatments are not independent between each other. In practice, this implies that we will have to compute the covariance matrix for all pairs of treatment times, instead of just its diagonal.

In figure 4.2 we can compare the covariance matrix and samples for the latent forces and outputs between two models. The first model is a LFM for Treatment Response Estimation discussed in the background section, the second one is the one we have just described. We can notice that in the Time-limited LFM output kernel, after the treatment duration is over and the treatment's latent force is 0, the output physiological quantity shows exponential decay behaviour.

4.2.2 Limitations

Having now included some knowledge-driven inductive bias inside our model, thanks to the LFM formulation, our model should, in theory, be able to fit more complex datasets and provide better extrapolation performance with less data. This, however comes with two big associated costs

- Knowledge-driven ODE model is wrong. This is the Achilles' heel of all mechanistic models. When using a purely nonparametric universal function approximator, like a GP or a Neural Network, we do not have to worry about biasing the system towards learning an unrealistic model. On the other hand, by using LFMs, we are forcing the model to learn latent functions inside a linear, inhomogeneous, constant coefficient ODE. This is almost surely not how the underlying process we are trying to model really works, and there is a big possibility that it is not a good approximation either.

- Performance. The LFM output kernels, both time-marked and time-

limited involve heavy use of nontrivial mathematical operations such as division, the complementary error function, and exponentiation. Additionally, since we treat all treatments as dependent, we have to evaluate M^2 times the covariance function on all N^2 input points, as opposed to the M evaluations needed to compute the independent version. Because of these two factors fitting these models becomes very compute and memory intensive, even when using hardware accelerators such as GPUs.

- No knowledge of treatment covariates. Like the first model, we have not used data about the treatment covariates.

4.3 Treatment Covariates

One of the main goals of this work is to improve existing LFM for treatment response estimation by using the information contained in treatment covariates. We study the behaviour of scaling the treatment response curves by a factor obtained as a function of the treatment covariates.

We extend our existing generative model by applying a scaling factor $S(\mathbf{x}_m)$ to the treatments

$$\hat{y}(\tau) = f_b(\tau) + \sum_{t_m} f_t(\tau; t_m)$$
$$\hat{y}(\tau) = f_b(\tau) + \sum_{t_m} S(x_m) f_t(\tau; t_m)$$

4.3.1 Linear Scaling

Let x be the treatment covariates.
Let $x_m \in \mathbf{x} = \{x_1, x_2, ..., x_m\}$ be the covariates associated to one specific treatment m. For our analysis, we will consider $x_m \in \mathbb{R}^K$. Each component x_{im} can be, for example, the dosage of one of the active ingredients of the drugs, or the amount of macro-nutrients in a meal.

We define the scaling function S to be a linear combination of the individual covariate components plus an intercept γ.

$$S \stackrel{\text{def}}{=} \beta^T x_m + \gamma$$

where $\beta \in \mathbb{R}^K, \gamma \in \mathbb{R}$.

4.4 Individual-level Treatment Sharing

Up to this point, all of our models have considered a single individual at a time. We believe that the treatment responses of two separate individuals are not completely independent between each other. Because of this, the predictions for one individual can be improved by adding data from multiple other individuals. We discuss how to implement a model that incorporates information from multiple individuals using the formalism of Multiple-Output Gaussian Processes (MOGPs).

4.4.1 Model definition

We start from the multiple-individual generative model

$$\hat{y}^{(p)}(\tau) = f_b^{(p)}(\tau) + \sum_{t_m} f_t^{(p)}(\tau; t_m)$$

and start deriving an expression for the multi-output covariance function. That is, a function whose arguments include both a pair of times (τ, τ') but also a pair of individual indices (p, p').

$$\begin{aligned}
&\text{cov}(\hat{y}^{(p)}(\tau), \hat{y}^{(p')}(\tau')) \\
=&\text{cov}(f_b^{(p)}(\tau) + \sum_{t_m} f_t^{(p)}(\tau; t_m), f_b^{(p')}(\tau') + \sum_{t_{m'}} f_t^{(p')}(\tau'; t_{m'})) \\
=&\mathbb{E}\left[\left(\sum_{t_m} f_t^{(p)}(\tau; t_m)\right)\left(\sum_{t_{m'}} f_t^{(p')}(\tau'; t_{m'})\right)\right] \\
=&\sum_{t_m}\sum_{t_{m'}} \mathbb{E}\left[f_t^{(p)}(\tau; t_m) f_t^{(p')}(\tau'; t_{m'})\right]
\end{aligned}$$

The next steps of the derivation depend on the choices we make about two aspects of the model. First of all, whether the treatment for the same individuals but for two different times are independent or not. Second, whether two treatments for two different individuals are independent or not.

For both cases, we decide to consider all of them to be dependent, thus

$$\forall p, p' \in 1, 2, ..., P \quad f_t^{(p)} = f_t^{(p')}$$

4.4.2 ICM for Individual-level Treatment Sharing

We can interpret the covariance matrix that we have just build using the framework of MOGPs.

Consider the MOGP $\hat{\mathbf{y}}(\tau)$ defined as

$$\hat{\mathbf{y}}(\tau) = \begin{bmatrix} \hat{y}^{(1)}(\tau) \\ \hat{y}^{(2)}(\tau) \\ \vdots \\ \hat{y}^{(P)}(\tau) \end{bmatrix}$$

Then its covariance function is a block-matrix with the following structure

$$\text{cov}(\hat{\mathbf{y}}(\tau), \hat{\mathbf{y}}(\tau')) = \begin{bmatrix} \text{cov}(\hat{y}^{(1)}(\tau), \hat{y}^{(1)}(\tau')) & \dots & \text{cov}(\hat{y}^{(1)}(\tau), \hat{y}^{(P)}(\tau')) \\ \vdots & \ddots & \vdots \\ \text{cov}(\hat{y}^{(P)}(\tau), \hat{y}^{(1)}(\tau')) & \dots & \text{cov}(\hat{y}^{(P)}(\tau), \hat{y}^{(P)}(\tau')) \end{bmatrix}$$

By assuming that all treatments have the same underlying treatment response function, i.e. the same single sample from a GP then

$$\text{cov}(\hat{\mathbf{y}}(\tau), \hat{\mathbf{y}}(\tau')) =$$
$$\begin{bmatrix} \sum_{t_m^{(1)}} \sum_{t_{m'}^{(1)}} k(\tau^{(1)}, \tau'^{(1)}; t_m^{(1)}, t_{m'}^{(1)}) & & \\ & \ddots & \\ & & \sum_{t_m^{(P)}} \sum_{t_{m'}^{(P)}} k(\tau^{(P)}, \tau'^{(P)}; t_m^{(P)}, t_{m'}^{(P)}) \end{bmatrix}$$

Notice that since, in general $\tau^{(p)} \neq \tau'^{(p')}$, that is the observations are not *homotopic*, but rather *heterotopic* i.e. not aligned, we cannot simply take the kernel out of the matrix via a Hadamard product.

4.4.3 Kernel

We need to extend the definition of our the kernel, since we now need to evaluate the "cross-covariance" between treatments done on two different individuals with different observations time $\tau^{(p)}, \tau'^{(p')}$ and treatment times $\mathbf{t}^{(p)}, \mathbf{t}^{(p')}$.

Thus, we redefine the Time-Limited Squared Exponential (TLSE) kernel

$$k_{\text{TLSE}}(\tau, \tau'; t_m, t_{m'}) \overset{\text{def}}{=} \begin{cases} k_{\text{SE}}(\tau - t_m, \tau' - t_{m'}) & \text{if } t_m < \tau < t_m + T \text{ and} \\ & t_{m'} < \tau' < t_{m'} + T \\ 0 & \text{otherwise} \end{cases}$$

and with an identical argument we could redefine other base kernels too.

In figure 4.3 we can view and compare the covariance matrices generated by the multi-output kernel. The kernel is evaluated on two individuals, each one receiving one treatment at times 1 and 5, respectively. Hence the treatment time vectors will be $\mathbf{t}^{(1)} = [1], \mathbf{t}^{(2)} = [5]$. The same procedure is repeated with two different base kernels, a time-limited SE Kernel and a time-limited LFM kernel. Then samples from the two individual's GPs are plotted on the same panel. Notice how the treatment response functions have the same shape for the two individuals, even though they have different starting times.

4.4.4 Hierarchical Linear Scaling Coefficients

Let us now introduce the treatment covariates in this model. We do this by following the linear scaling approach described earlier into this chapter. This means that our generative model will be

$$\hat{y}^{(p)}(\tau) = f_b^{(p)}(\tau) + \sum_{t_m, x_m} S^{(p)}(\mathbf{x_m}) f_t^{(p)}(\tau; t_m)$$
$$S^{(p)}(x_m) = (\beta^{(p)})^T x_m + \gamma^{(p)}$$

Deciding how much information about the coefficients $\beta^{(p)}, \gamma^{(p)}$ should be shared across individuals is the differentiating factor for the next three models.

- *Unpooled*. In the unpooled model, we learn a separate set of coefficients for every individual. While allowing for the largest flexibility, this model is also very sensitive to noise in the dataset, which could lead it to learn unreasonable coefficients in the training phase, leading to poor predictive performance.

  Using formulas, we would write the Bayesian model as

$$\beta^{(p)} \sim \mathcal{N}(\mu_\beta^{(p)}, \sigma_\beta^{(p)})$$
$$\gamma^{(p)} \sim \mathcal{N}(\mu_\gamma^{(p)}, \sigma_\gamma^{(p)})$$

- *Pooled* The pooled model is on the opposite side of the spectrum compared to the unpooled one. Rather than learning a separate set of parameters for each individual, we learn a single set shared by all individuals. This allows us to learn a very small number of parameters with a very large number of samples, protecting us from the risk of overfitting.

$$\beta^{(p)} \sim \mathcal{N}(\mu_\beta, \sigma_\beta)$$
$$\gamma^{(p)} \sim \mathcal{N}(\mu_\gamma, \sigma_\gamma)$$

- *Hierarchical* We wish to get the best of both words with an hierarchical model. Here, we model the coefficients as the sum of a shared component plus individual-specific corrections. Ideally this will allow us to both learn a robust baseline and give us enough flexibility to model patient-specific reactions.

$$\mu_\beta^{(p)} \sim \mathcal{N}(\nu_\beta, \tau_\beta)$$
$$\mu_\gamma^{(p)} \sim \mathcal{N}(\nu_\gamma, \tau_\gamma)$$
$$\beta^{(p)} \sim \mathcal{N}(\mu_\beta^{(p)}, \sigma_\beta^{(p)})$$
$$\gamma^{(p)} \sim \mathcal{N}(\mu_\gamma^{(p)}, \sigma_\gamma^{(p)})$$

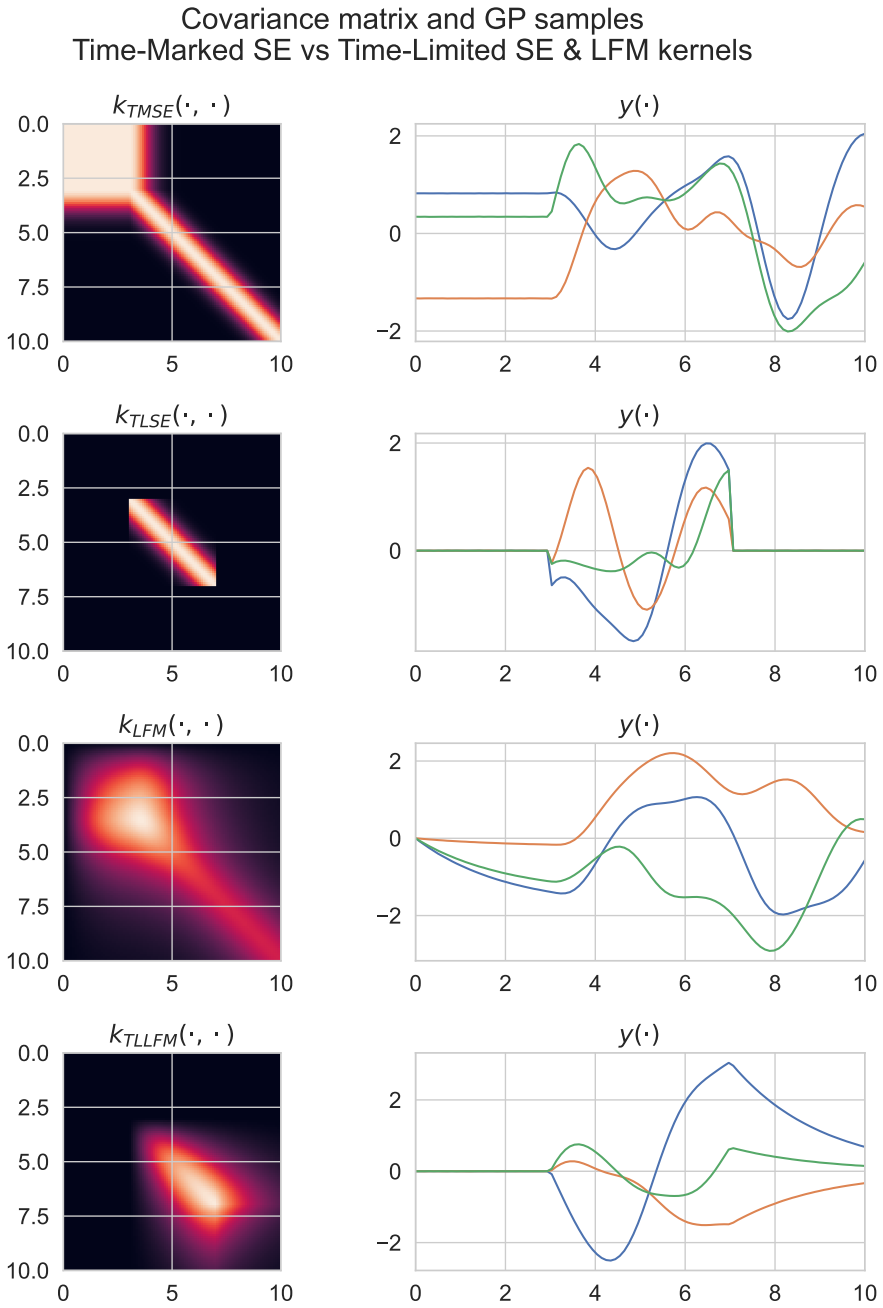


Figure 4.2. The four rows display the covariance matrix and GP samples from: Time-marked Squared Exponential kernel, Time-limited Squared Exponential kernel, Time-marked LFM Output Kernel, Time-limited LFM Output Kernel. All kernels share the same length scale $\ell = 1$ and treatment time $t_m = 3$. The time-limited kernels use a treatment duration $T = 4$. All LFM kernels use decay rate $D = 0.5$ and sensitivity $S = 1.5$.

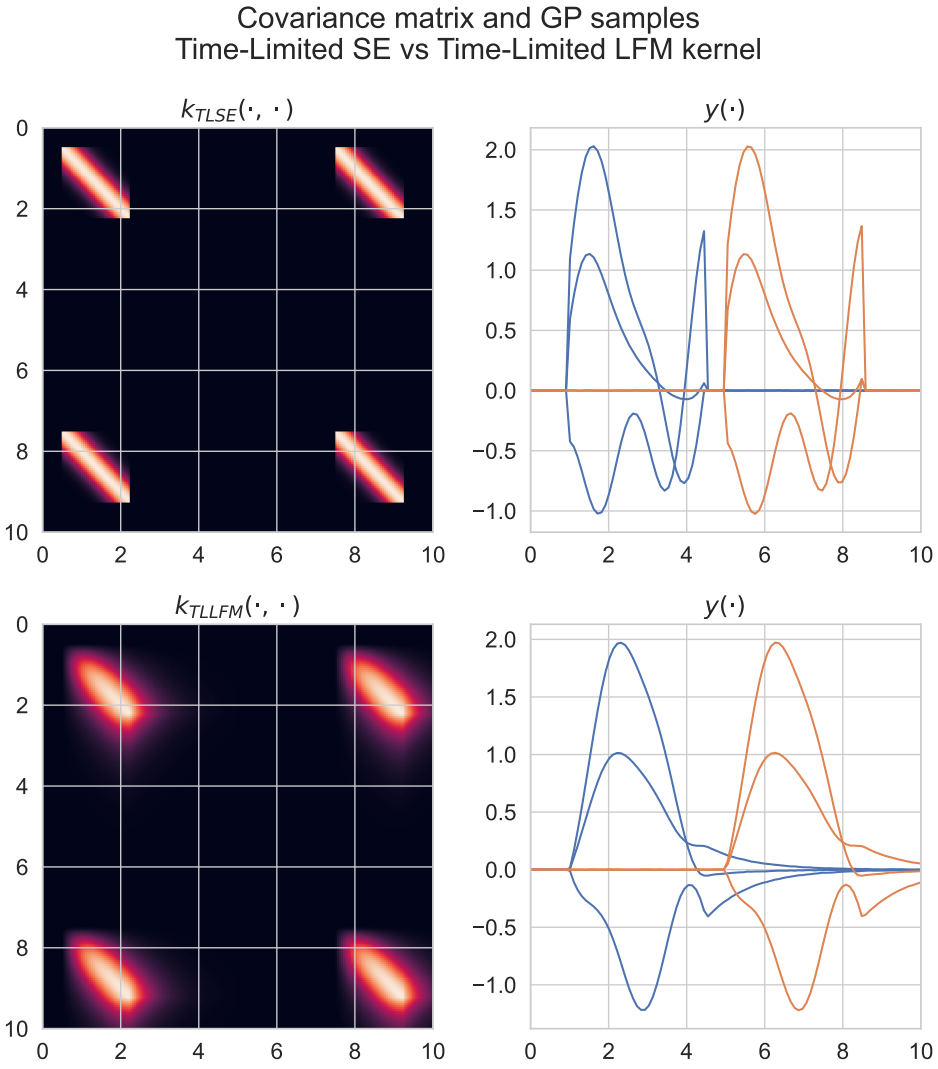


Figure 4.3. Comparison of the covariance matrices and GP samples generated by the multi-output kernel. The kernel is evaluated on two individuals, each one receiving one treatment at times 1 and 5, respectively. Two base kernels are used: a time-limited SE Kernel and a time-limited LFM kernel. Both kernels share the same length scale $\ell = 1$ and duration $T = 3.5$. The LFM kernel has decay rate $D = 0.9$ and sensitivity $S = 1.5$. Samples from the GP of each individual are plotted in the same panel, in blue and orange for individual 1 and 2, respectively.

5. Experiments

In this section, we demonstrate the efficacy of our methods in modeling multiple treatment effects on two datasets, an artificial simulated dataset and a real dataset using data from the Helsinki University Hospital.

We describe the generation procedure of the simulated dataset and the data and preprocessing steps for the real dataset.

First, we use simulated data to discuss the shortcomings of methods from previous works. Additionally, we show that our newly presented methods can fit the artificial dataset successfully, achieving satisfactory performance.

Finally, we train our newly developed methods on the real dataset and we show empirical performance results for our method using the metrics of prediction accuracy.

5.1 Simulated Data

5.1.1 Dataset Generation

We simulate artificial data using a Latent Force Model.

$$\hat{y}(\tau) = f_b(\tau) + \sum_{t_m} f_t(\tau; t_m)$$

$$f_b(\tau) = 0$$

$$f_t'(\tau; t_m) = B - D f_t(\tau; t_m) + S f_l(\tau; t_m)$$

$$f_l(\tau) = \exp\left(-\frac{1}{2}\frac{(\tau - t_m)^2}{1^2}\right)$$

where we have chosen the basal rate parameter $B = 0$, the decay rate $D = 0.2$, the sensitivity $S = 0.1$. The treatment times are $\{15, 25\}$.

The ODE is numerically solved through SciPy's `solve_ivp` routine. The

numerical solution is evaluated at 100 points, which form the full dataset.

Finally, independent and identically distributed Gaussian noise is added to every point in the dataset, with zero mean and standard deviation $\sigma = 0.05$.

$$y(\tau) = \hat{y}(\tau) + \epsilon(\tau)$$
$$\epsilon(\tau) \sim \mathcal{N}(0, \sigma^2)$$

5.1.2 Experiments

The goal of these experiments is twofold. First to verify the correctness of model implementations. Second, to verify that the LFMs can recover the underlying dynamics correctly.

The dataset is used to evaluate three models:

- *Time-Marked Latent Force*

  A simplified version of model described in [4]. We choose to implement the model described in the paper with three simplifications. We assume that there is only one single individual. We only model one single output. Our baseline model is the zero function, instead of a GP with a squared exponential plus periodic kernel

- *Time-Limited Treatment Response*

  The first model described in the methods section. The treatment responses are modeled with the newly-introduced Time-Limited Squared Exponential (TLSE) kernel.

- *Time-Limited Latent Force*

  The second model described in the methods section. The latent forces are GPs using the TLSE kernel, and thus the treatment responses use a custom Time-Limited LFM kernel.

5.1.3 Results

We plot the results of our simulated data experiments in figure 5.1.

- *Time-Marked Latent Force*

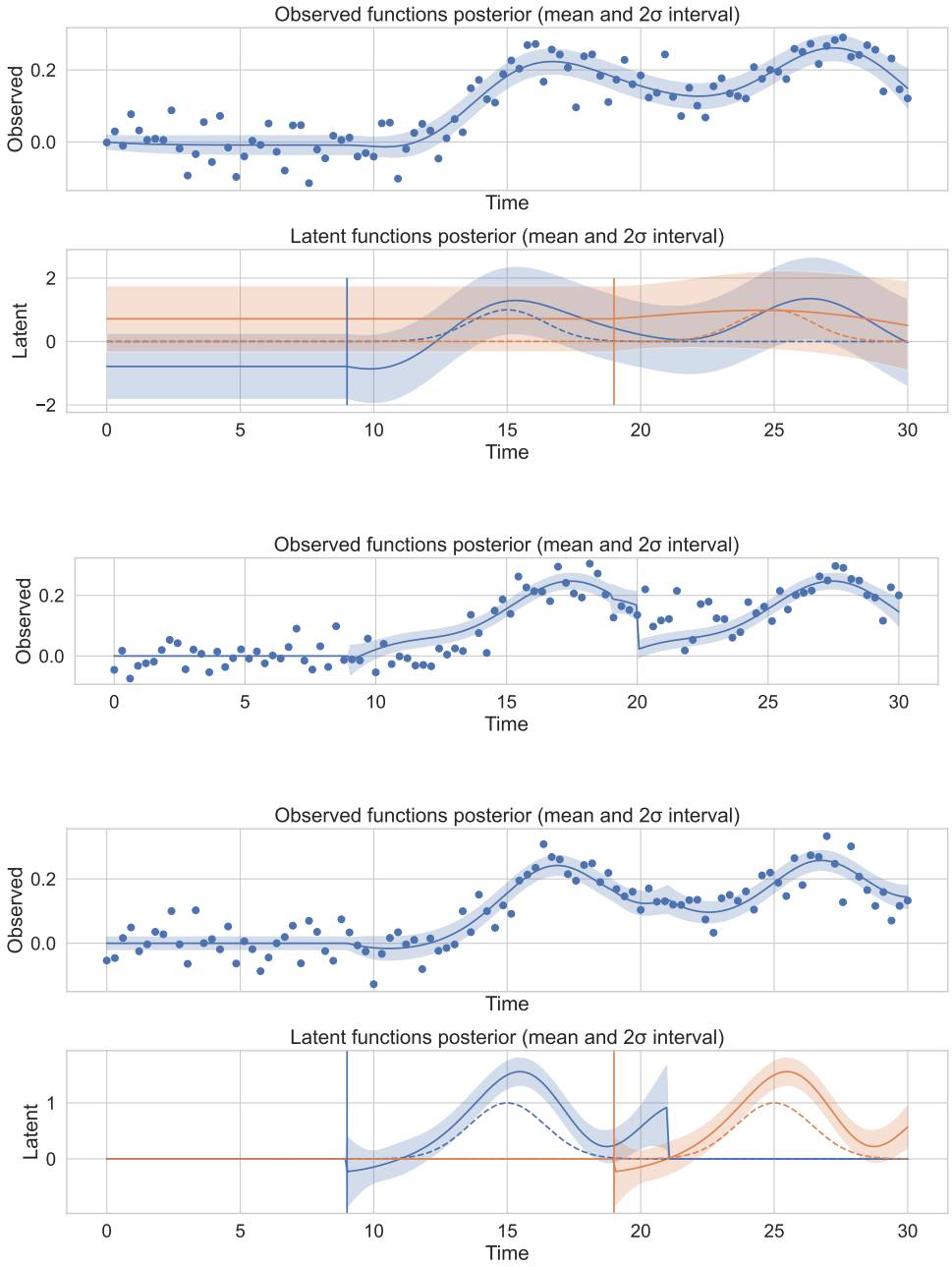


Figure 5.1. Top group: *Time-Marked Latent Force*.
Top panel: the observed function's posterior mean and 95% credible interval is plotted on top of the observations. Bottom panel: the latent functions' posterior mean and 95% credible interval are plotted using solid lines and shaded areas. The true latent functions used to generate the dataset are plotted with dashed lines.
Center group: *Time-Limited Treatment Response*.
The observed function's posterior mean and 95% credible interval is plotted on top of the observations.
Bottom group: *Time-Limited Latent Force*.
Top panel: the observed function's posterior mean and 95% credible interval is plotted on top of the observations. Bottom panel: the latent functions' posterior mean and 95% credible interval are plotted using solid lines and shaded areas. The true latent functions used to generate the dataset are plotted with dashed lines.

The model successfully successfully fits the dataset. Which, is expected, as the dataset itself was generated using a latent force model.

Looking at the bottom panel we see one of the main limitations of this methods: the first latent force (blue solid line) captures the effect of the two treatments, while the second latent force (orange solid line) does not. Additionally, we see that the first latent force and the second latent forces are nonzero before the first treatment. The first latent force is negative and the second is positive so the cumulative effect is zero. While this fact is not an issue for fitting the data, it does not make sense when trying to interpret the latent forces from a clinical perspective.

Furthermore, we stress the fact that since all latent forces are assumed to be independent, it is not clear what the prediction setup would look like.

Finally, the uncertainty of the latent forces' means is really high. This is not an issue per se, but we expected less uncertainty when training on such a large amount of data. The uncertainty, again, can be explained by the fact that forces are not limited to being zero before the effect. Thus, the two latent forces are "fighting" each other and we see the full spectrum of possible forces whose sum is zero.

- *Time-Limited Treatment Response*

  The model fits the dataset, albeit the fit is worse than the other two.

  Introducing a constraint on the support of the treatment responses shows its benefits immediately with this model. The two latent forces are not "fighting" each other anymore, which introduced a lot of variance in the last model.

  The main issue observed with the this model is due to its kernel. The Time-Limited kernels generate discontinuous latent functions which, in this case, make it hard to fit a dataset generated by smooth functions.

  As we are directly modeling the treatment response curve with a GP, rather than the latent forces, we do not have a second panel.

- *Time-Limited Latent Force Model*

  The model successfully fits the dataset. Since this kernel is derived from the LFM formulation and the dataset is generated using a LFM, we expect the fit to be great.

  Again, making the latent forces a time-limited signal is paying off significantly. Compared to *Time-Marked Latent Force*, the uncertainty of

the latent force posterior mean is much smaller. The smaller uncertainty can be explained by the fact the the two latent forces almost never overlap each other, thus the possible number of curves that explain the dataset is much smaller than in the prior model. This hypothesis can be confirmed by looking at the region of the plot where $\tau \in [19, 21]$, where the two latent forces overlap. There the variance is much higher, resembling our first plot.

In addition, since the two latent forces are fully dependent, rather than being completely independent, the hypothesis space shrinks significantly and thus the variance of the estimates does as well.

On a final note, we can observe in the bottom panel that the latent force's mean and the true latent force have similar shapes but different amplitudes. There is a simple explanation for this phenomenon. Both the latent forces' kernel and the sensitivity parameter S control the amplitude of the final latent force. Since there are two parameters that influence in the same way the same quantity we end up having a non-identifiable system. This means that, to compare the amplitude between two different fits we may want to normalize the final sensitivities or avoid training them altogether, and just relying on the flexibility of the GP kernel.

From this set of experiment we can draw the following conclusions.

First, the model described in [4], while being a great foundation for treatment response curve estimation, has two critical flaws. The first flaw being latent forces having infinite duration and the second one is the independence of latent forces. For these two reasons, we will not use this model on real data.

Second, introducing a time-limited kernel and dependence between forces greatly reduces the uncertainty of the model. For these reasons, we will continue using these models in the second set of experiments on the real-world glucose dataset.

5.2 Glucose Data

5.2.1 Dataset

We evaluated our methods using clinical data collected at Helsinki University Hospital and provided by the Obesity Research Unit at the University of Helsinki.

The dataset contains blood glucose measurements and meal macronutrient data of 14 non-diabetic individuals observed across three days. The blood glucose measurements are collected by a portable continuous glucose monitoring system at approximately every fifteen minutes. In total, there are around 300 real-valued observations per individual.

The meal times (treatment times) and meal macronutrient contents (treatment covariates) have been collected for all meals during the study period. The macronutrients are five: starch, sugar, fiber, fat, and protein.

The goal is to learn the response curve associated with every meal and to predict as accurately as possible the effects of an arbitrary meal on the blood glucose levels.

The data is preprocessed by selecting all meals where the sum of starch and sugar is above the threshold of 10. In our experiments, we only include starch and sugar as treatment covariates.

Since this is a real-world dataset, there are several sources of errors, both systematic and random. The blood glucose measurements are noisy, due to the limitations of the sensors. Since the treatment times and covariates are reported by the users, there are frequent reporting errors both in the meal timing and the amount of macronutrients consumed in each meal.

Figure 5.2 displays the blood glucose trajectories as well as the associated treatment times and covariates for the first four individuals of the dataset.

5.2.2 Evaluation Setup and Metrics

To evaluate the predictive performance the dataset is split in two folds using a time-series holdout scheme. The training set consists of the first two days and the test set is the third day. The models are trained on the training set and then must predict on the test set.

The metric used for evaluation is Mean Squared Error (MSE). For every individual p in the dataset, we compute the MSE using the true values from the test set and the model's predictions. All the MSEs are then

averaged to obtain the mean MSE (mMSE), the metric we use to compare models.

$$\text{MSE}^{(p)} = \frac{1}{N}\sum_{i=1}^{n}(y_i^{(p)} - \hat{y}^{(p)}(\tau_i))^2$$

$$\text{mMSE} = \frac{1}{P}\sum_{p=1}^{P}\text{MSE}^{(p)}$$

5.2.3 Experiments

In our experiments, we train and compare the predictive accuracy metrics for 8 different models. We evaluate several different combinations of kernels, treatment response curve sharing, and scaling coefficients sharing.

We begin by comparing the Time-Limited Treatment Response (TR) model against the Time-Limited Latent Force (LF) model. The models are trained one individual at at time, learning a separate treatment response curve for every individual. Finally, the treatments are scaled with a linear combination of the treatment's covariates. The scaling coefficients can either be fixed or a separate set is trained for every individual (unpooled).

For a visual comparison of the TR and LR models with separate curves and unpooled coefficients see the first two panel groups of figure 5.4.

After having identified the best performing kernel, we compare four possible methods for determining the scaling coefficients. Non-trainable fixed coefficients, separate set of trainable coefficients for every individual (unpooled), one set of trainable coefficients shared across all individuals (pooled) and the finally hierarchical coefficients.

The last panel group of figure 5.4 shows the TR model with pooled treatment response curves and pooled coefficients trained on individuals 0 and 1 at the same time.

5.2.4 Results

The goal of the first set of experiments is to identify the best performing kernel. We report our results in table 5.1. Our results found evidence that, for this dataset, the Time-Limited Treatment Response model's errors are lower than the Time-Limited Latent Force model. Additionally, we have found that learning the hyperparameters of the Latent Force model is challenging and that the optimization's results heavily depend on the initial conditions. This is because even small changes in the ODE parame-

ters such as the decay rate or the sensitivity cause big variations in the response curve's shape and magnitude.

In light of the kernel comparison results, we run a second set of experiments using the best performing kernel, i.e. the TR kernel. From this experiment, we find that sharing the same treatment response curve across individuals performs better than learning a separate curve for every individual.

We claim that the shared models perform better than the separate ones because of the superior robustness to noise of the shared models. This claim is supported by our comparison of per-individual MSEs. We compare two models with identical kernel and scaling coefficients but the first model learns separate response curves while the second uses shared ones.

In figure 5.3 we see a comparison of the Mean Squared Errors (MSEs) for every individual. The top panel compares the MSEs of two TR models with fixed scaling coefficients. The bottom panel compares the MSEs of two TR models with unpooled scaling coefficients. For fixed scaling coefficients, the performance of the two models is comparable for all individuals except individuals 4 and 5. For unpooled scaling coefficients, the performance is also comparable for most individuals but there is more variance in the performance differences, attributable to the higher flexibility of the unpooled model than the fixed one.

Finally, after having determined the best-performing kernel and treatment response curve sharing method, we focus our attention on how to share the scaling coefficients. Our experiments show that all scaling coefficients sharing methods have similar performance, with the pooled model, which learns one set of coefficients for all individuals having the lowest error. Again, we claim that the the pooled model works better because of its superior noise robustness compared to the alternatives.

Kernel	Treatment Response Curve	Scaling Coefficients	mMSE
TR	separate	fixed	0.640
TR	separate	unpooled	0.641
LF	separate	fixed	0.670
LF	separate	unpooled	1.221
TR	shared	fixed	0.572
TR	shared	unpooled	0.573
TR	shared	pooled	**0.564**
TR	shared	hierarchical	0.568

Table 5.1. Prediction results on test data. The mean Mean Squared Error (mMSE) is computed for the models described in our method. The best performing model uses a Time-Limited Treatment Response kernel, shared treatment response curve, and a single set of scaling coefficients for all patients (unpooled scaling coefficients).

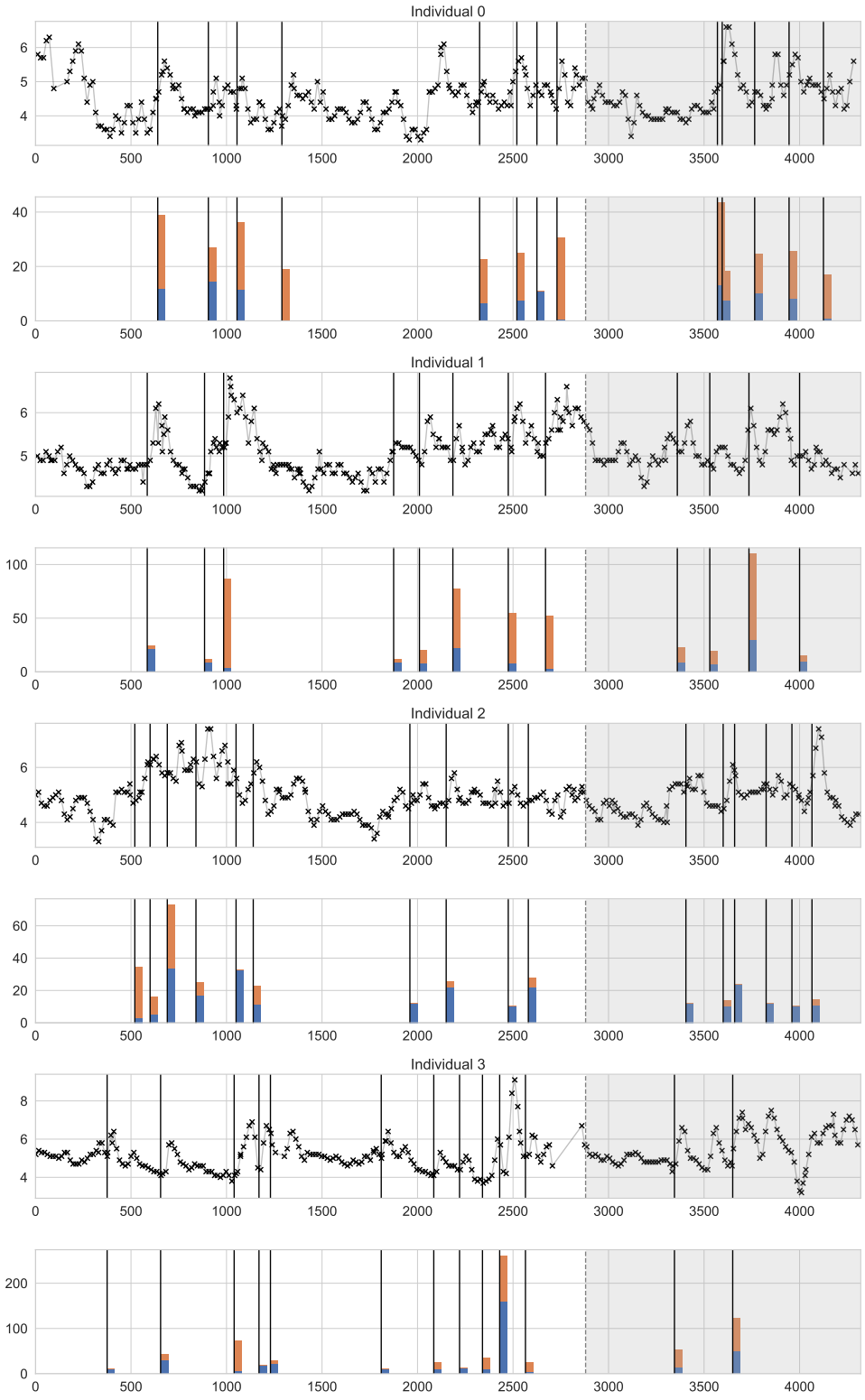


Figure 5.2. Visualization of the 3-days of blood glucose dataset. For each pair of panels, the top panel displays the blood glucose observation time-series with black signs joined by solid gray lines. The treatment times appear on all panels as solid black vertical lines. The bottom panels displays the treatment covariates associated with every treatment. Areas overlaid in gray are the testing set where the model is evaluated, the remainder is the training set.

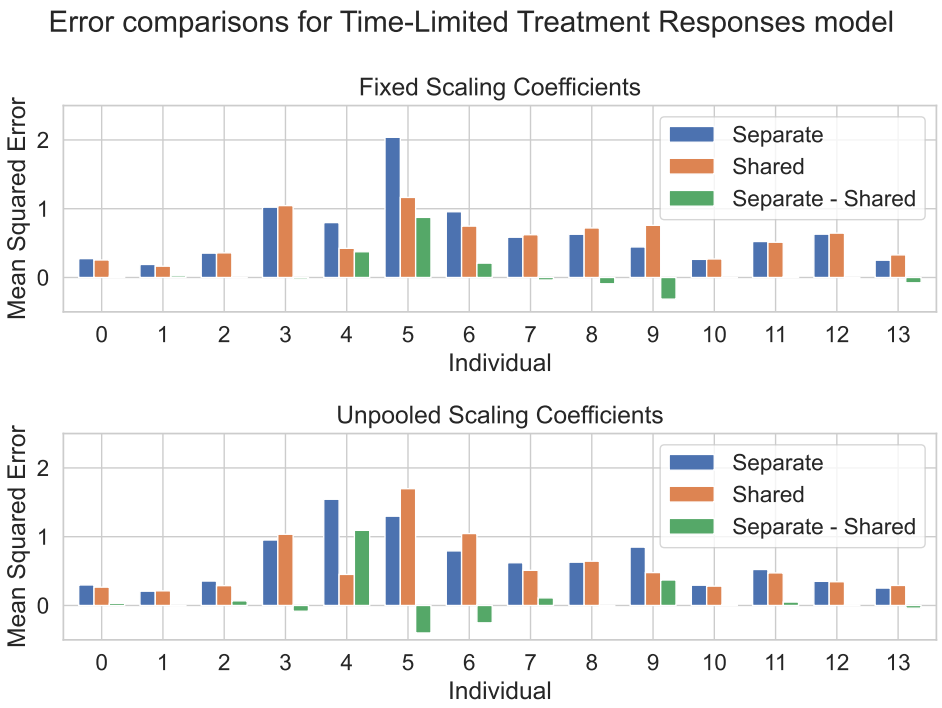
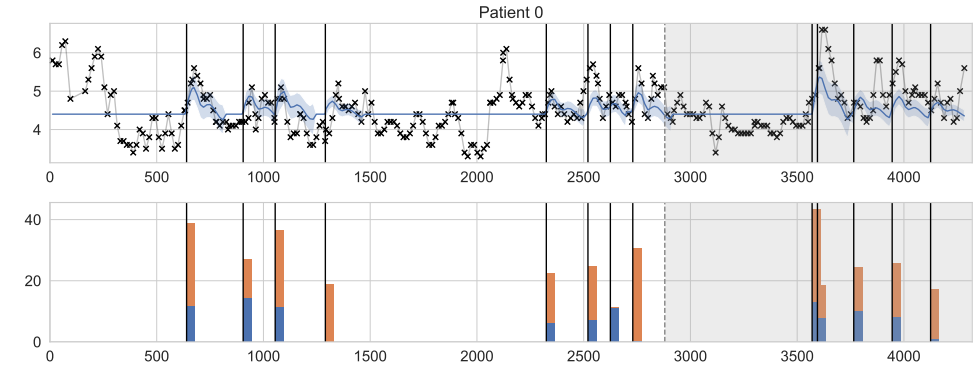
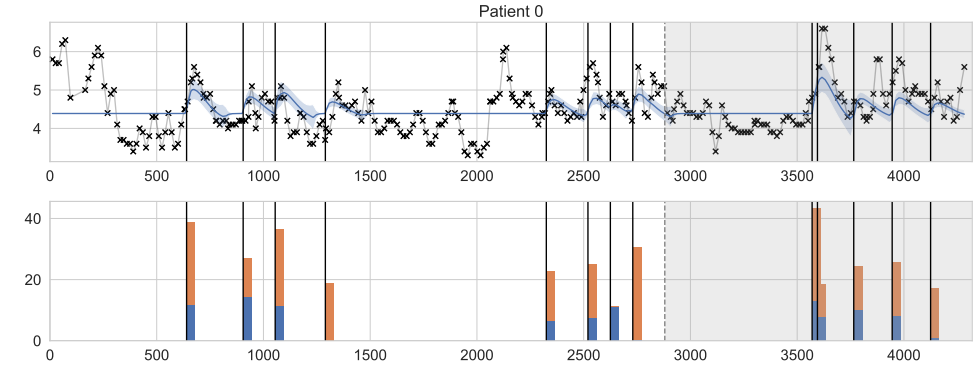


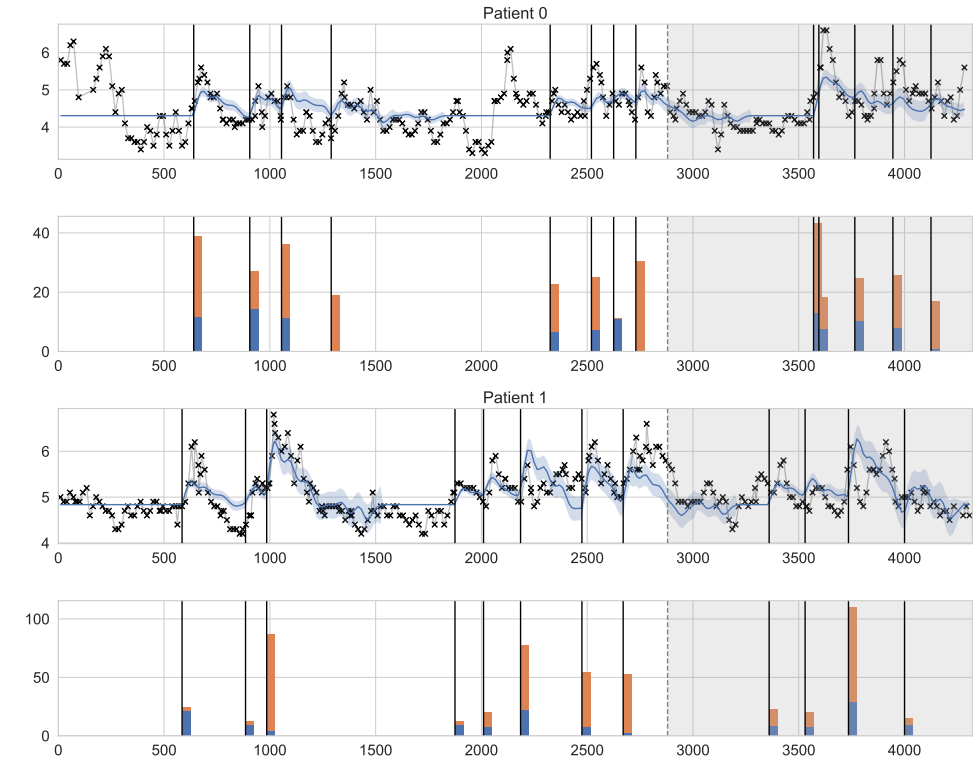
Figure 5.3. Comparison of the Mean Squared Errors (MSEs) for every individual. The top panel compares the MSEs of two TR models with fixed scaling coefficients and the bottom panel compares the MSEs of two TR models with unpooled scaling coefficients.
The difference Separate − Shared is plotted and, as expected, it is positive in most individuals, consistent with the lower mMSE of the shared model.



(a)



(b)



(c)

Figure 5.4. Prediction of blood glucose level after a meal. (a): results from the Time-Limited Treatment Response method trained separately on patient 0 and using unpooled scaling coefficients. (b): results from the Time-Limited Latent Force method trained separately on patient 0 and using unpooled scaling coefficients. (c): results from the Time-Limited Treatment Response method trained at the same time on patients 0, 1 and using pooled scaling coefficients.

6. Discussion

This final section contains a summary of the results and the conclusions we have drawn from the experiments. Then, we propose some directions for future research and finally, we consider the potential impact of our work.

6.1 Summary of results

We have verified the correctness of our implementations of methods from related works and of our proposed methods on a simulated dataset. The dataset is simulated using a LFM, which allows us to verify that the Time-Marked Latent Force (TMLF) and Time-Limited Latent Force (TLLF) models can indeed recover the original latent forces.

Through this experiment we identify two limitations of the TMLF model: The latent forces are not limited in time, which causes the latent forces of two different treatment to overlap each other. The latent forces are independent, which forbids us from using the model for prediction forward in time.

On the other hand, the results obtained by the Time-Limited Treatment Response (TLTR) and Time-Limited Latent Force models are satisfactory and thus we select these models for the next stage.

We evaluate the models on a blood glucose prediction task using real data collected by the Helsinki University Hospital. Our experiments show that the TLTR model has better predictive performance, faster training time, and higher noise robustness than the TLLF model.

The experiments on real data continue by evaluating the impact of sharing the treatment responses and scaling coefficients across multiple individuals for the TLTR model. We find that sharing the response curve improves the predictive performance and that using a single set of scaling

coefficients for the whole group of individuals results in the best performing model. We claim that superior performance of the shared model is due to the higher robustness of the model to noise and errors in the data.

6.2 Directions for the future

The overarching conclusion from our experiment is that, on dataset with large amounts of noise and errors, simpler models work better than more sophisticated one due to more noise robustness. We hope to repeat these experiments on additional datasets with smaller amounts of data but also smaller noise, to see if the knowledge-driven inductive bias of LFMs can help in those situations.

On a more technical note, we are interested in evaluating new methods for scaling the treatment response curve using treatment covariates. For example, using nonlinear models such as logistic-regression or neural networks. Additionally, given recent advancements in variational inference techniques and automatic differentiation, we believe that it is possible to successfully extend our model to nonlinear ODEs.

Finally, our experiments relied on Maximum a Posteriori (MAP) optimization to find the kernel hyperparameters. Ideally, we would use Markov Chain Monte Carlo (MCMC) techniques for estimating them, in order to obtain credible intervals on the hyperparameters and, in general, to achieve more robust predictions. That will probably require improving the performance of our log-likelihood evaluations significantly, which we believe is possible by using sparse GP techniques such as inducing points.

6.3 Possible impact

The results achieved in our experiments provide useful insights for future research in individualized treatment response estimation. While showing that highly flexible models like GPs can indeed estimate plausible treatment response curves, our results also come with a word of caution on using such high capacity methods on datasets with such high amounts of noise.

We believe that the methods we developed, even if not used directly for treatment response estimation, can be used as a guiding tool for developing more physiologically accurate parametric treatment curves.

On a broader level, our goal is to participate in the advancement of the field of precision medicine. By providing clinicians with better tools to estimate the future state of their patients, we hope that they can make decision that improve the quality of health care and the quality of life of those who need it.

Bibliography

[1] Boyce et al. *Elementary Differential Equations and Boundary Value Problems*. Wiley, 2017. ISBN: 9781119443766. URL: https://books.google.fi/books?id=SyaVDwAAQBAJ.

[2] Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. *Kernels for Vector-Valued Functions: a Review*. 2011. DOI: 10.48550/ARXIV.1106.6251. URL: https://arxiv.org/abs/1106.6251.

[3] Mauricio Álvarez, David Luengo, and Neil D. Lawrence. "Latent Force Models". In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 2009, pp. 9–16. URL: https://proceedings.mlr.press/v5/alvarez09a.html.

[4] Li-Fang Cheng et al. "Patient-Specific Effects of Medication Using Latent Force Models with Gaussian Processes". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4045–4055. URL: https://proceedings.mlr.press/v108/cheng20c.html.

[5] Neil Lawrence, Guido Sanguinetti, and Magnus Rattray. "Modelling transcriptional regulation using Gaussian Processes". In: *Advances in Neural Information Processing Systems*. Ed. by B. Schölkopf, J. Platt, and T. Hoffman. Vol. 19. MIT Press, 2006. URL: https://proceedings.neurips.cc/paper/2006/file/f42c7f9c8aeab0fc412031e192e2119d-Paper.pdf.

[6] Jacob D. Moss et al. *Approximate Latent Force Model Inference*. 2021. DOI: 10.48550/ARXIV.2109.11851. URL: https://arxiv.org/abs/2109.11851.

[7] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006, pp. I–XVIII, 1–248. ISBN: 026218253X.

[8] S.M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists, Student Solutions Manual*. Fourth. Elsevier Science, 2009. ISBN: 9780080919423. URL: https://books.google.co.in/books?id=p3zNCgAAQBAJ.