POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Learning Nonparametric Individualized Treatment Response Curves

Supervisors

Prof. Mauro GASPARINI

Prof. Pekka MARTTINEN

Eng. Çağlar HIZLI

Candidate

Andrea COGNOLATO

July 2022

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 individualize the treatment for each patient and design optimal treatment regimes, using the vast amount of data stored in EHRs. 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. Part of my work was to develop a new nonparametric model for treatment response curves that achieves competitive performance against parametric models while allowing patient-specific customizations.

I have analyzed the differences between directly modeling the treatment responses with a Gaussian Process (GP) and modeling the treatment dynamics using a Latent Force Model (LFM). I evaluated three models on a challenging blood glucose prediction dataset. Additionally, I have developed a method for using the treatment's covariates to scale the response curves: several experiments were run comparing two GP regression models as well as several ways of sharing the treatment response and treatment covariate model between patients. This code and data are now public for reproducibility and as a building block for future work. Finally, State-Of-The-Art (SOTA) performance on the dataset was obtained and it was discovered that modeling the treatment dynamics with a LFM does not significantly improve the predictive performance.

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

Table of Contents

Chapter 1

Introduction

In healthcare, 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 last few years have witnessed a huge growth in the amounts of medical data, thanks to the advent of electronic health records (EHRs) and modern medical devices. The goal of precision medicine is to improve the treatment for each patient by providing individualized treatment strategies developed using this large amount of data [1].

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 [2]. One example application of ITR curves is to develop optimal dosing strategies for medications [3].

Several studies have proposed models for ITRs that rely on parametric curves. These parametric curves are designed using expert knowledge about the physiological response to the treatment. The parametric models presented in the literature display a wide variety of formulations, using models such as bell-shaped curves [4], analytical solutions to LTI systems [5].

Only a few studies in literature, such as [6], propose modeling the treatment response curve using a fully nonparametric approach. The key motivation for this work is 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.

The approach I used to solve the ITR estimation problem is to reimplement current state of the art Gaussian Process (GP) based ITR methods from the literature and to improve them by incorporating more data and more flexibility into their formulation. The probabilistic foundations of GPs are crucial to obtain

credible intervals for our predictions, in order to correctly estimate uncertainty. Additionally it is possible, but not straightforward, to customize the models and incorporate constraints by designing new covariance functions, or kernels.

To incorporate domain-specific knowledge about the physiological dynamics of treatments, I turned to Latent Force Models (LFMs) [7, 8]. By combining the mechanistic approach of Ordinary Differential Equation (ODE) modeling with nonparametric GPs, LFMs allow having flexible models with the extrapolation abilities of mechanistic models.

Current nonparametric models do not try to model continuous treatment dosages and only allow a finite number of treatment variants [6]. I have developed a method to include treatment covariates to the response estimation. This allows to predict the effect of treatments with doses never before experienced in the training dataset.

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

Findings show that GP-based nonparametric models can achieve satisfactory performance in ITR estimation task. Additionally I have found that on noisy datasets typical of the healthcare field [9], more sophisticated models often fail because of their lower noise robustness. In these experiments, it was found that simpler GP models are superior in terms of predictive performance, training time, and inference time to the more complex LFM models. Finally, the results show 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.

I hope that these results can provide useful insights for researchers interested in using nonparametric methods for ITR estimation. On a broader level, the goal is to have a positive impact in the field of precision medicine. 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 and 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 I will use to evaluate the methods, the notation, and describes the available data. Section 4 presents the newly developed methods, starting with the Time-Limited Squared Exponential Kernel in Section 4.1. A new LFM model, the Time-Limited Latent Force model,

is presented in Section 4.2. Then I discuss how to introduce treatment covariates in Section 4.3 and finally how to extend these treatment models to multiple patients in Section 4.4. Section 5 presents the experiments and their results, on simulated and real-world data. To conclude, in Section 6 the results are summarized, their significance discussed and directions for future research are considered.

Chapter 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 [10].

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 call these n-dimensional Gaussian distributions the finite-dimensional distributions of the process at x. 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 an important ingredient 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 relative finite-dimensional multivariate normal distribution 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.

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

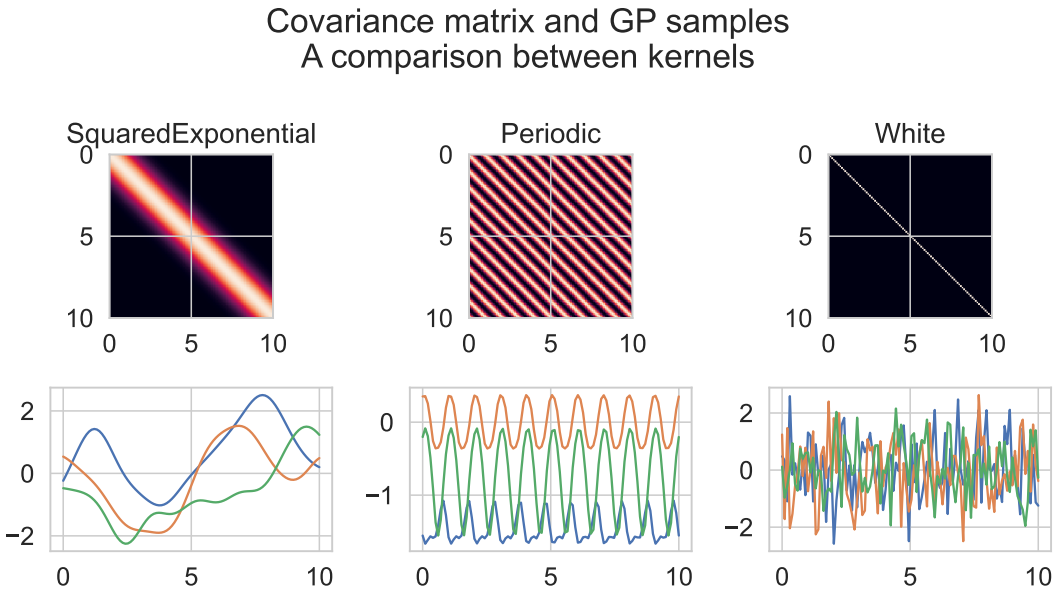


Figure 2.1: Comparison of the covariance matrix and samples from three zero-mean GPs with three different kernels. Left: A squared exponential kernel with lengthscale parameter $\ell = 1$. Center: A periodic squared-exponential kernel with lengthscale parameter $\ell = 1$ and period parameter $p = 1$.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

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 [11], 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 $\epsilon \in \mathbb{R}^n$. The ϵ are independent and identically

distributed (i.i.d.) Gaussian noise with known variance σ_n^2. 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.

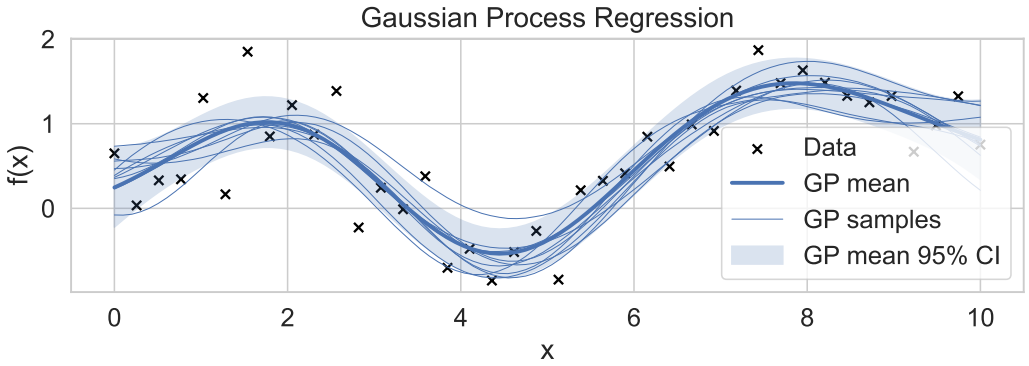


Figure 2.2: The chart shows an example of Gaussian Process Regression (GPR). The data, shown as black crosses, is generated by adding i.i.d. Gaussian noise with 0.5 standard deviation to the $\sin(x)$ function. A GP model with a squared exponential kernel is fitted and the three hyperparameters ℓ, σ, 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) [12].

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.

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

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 two panels on the right, we plot each one of the two independent Gaussian Processes. As expected, since the off-diagonal blocks in the covariance matrix are zeros, the two GPs are completely uncorrelated.

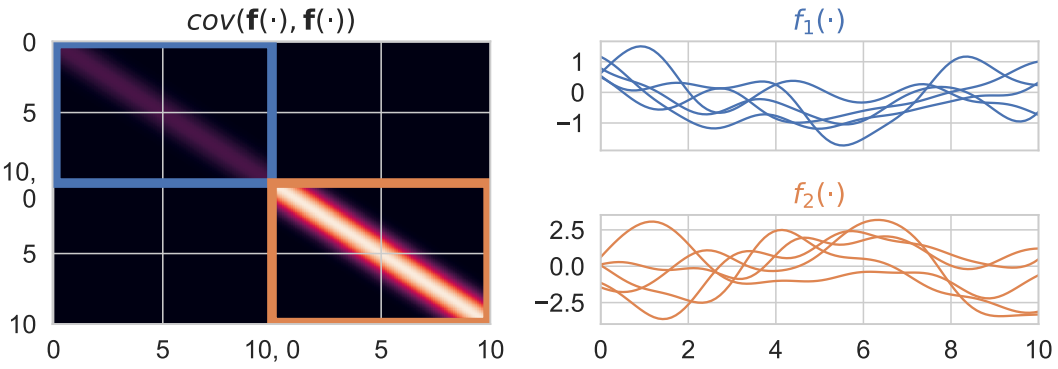


Figure 2.3: Left: the covariance matrix for the joint distribution of two independent one-dimensional GPs. For this plot, the two GPs use the same squared exponential kernel with the only difference being the scale parameters: $\sigma_1 = 0.5, \sigma_2 = 2.5$. Observe that this is a block-diagonal matrix. 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 $\mathbf{a} \in \mathbb{R}^d$ and define:

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

This model is called *Intrinsic Coregionalization Model* ICM [12]. 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

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

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.

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

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. Unlike the previous plot, this time the off-diagonal blocks of the covariance matrix are not zeros. These blocks, which we

call cross-covariance matrices, define the dependence structure between the two GPs. In this case, the two GPs are completely dependent on each other, with the only difference being a scaling factor.

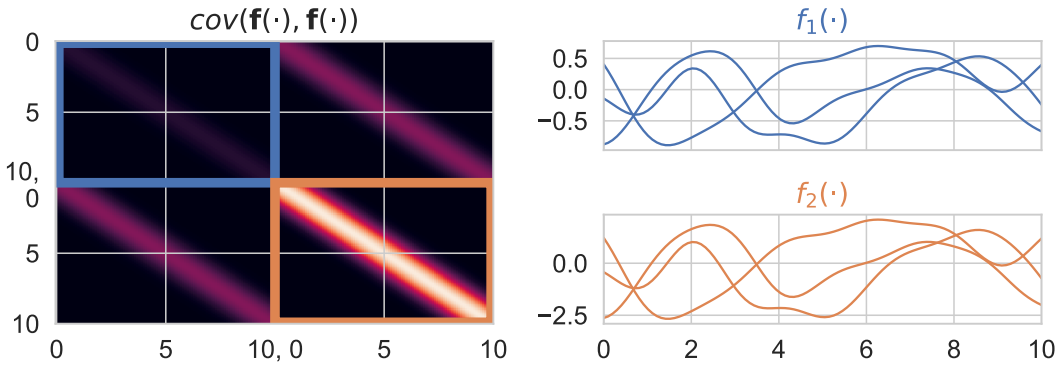


Figure 2.4: Left: the covariance matrix for the joint distribution of a Multi-Output Gaussian Process (MOGP) generated using the Intrinsic Coregionalization Model (ICM). For this plot, the two GPs use squared exponential kernels with scale parameters: $\sigma_1 = 0.5, \sigma_2 = 1.5$ and a common lengthscale $\ell = 1$. 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^{nd} GP $f_2(\cdot)$ are, as expected, simply the scaled version of the 1^{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 [13].

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.

2.4 Latent Force Models

Latent Force Models (LFMs) [8, 7] 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 [7], nonlinear ODEs [8, 14] and Partial Differential Equations (PDEs) [14].

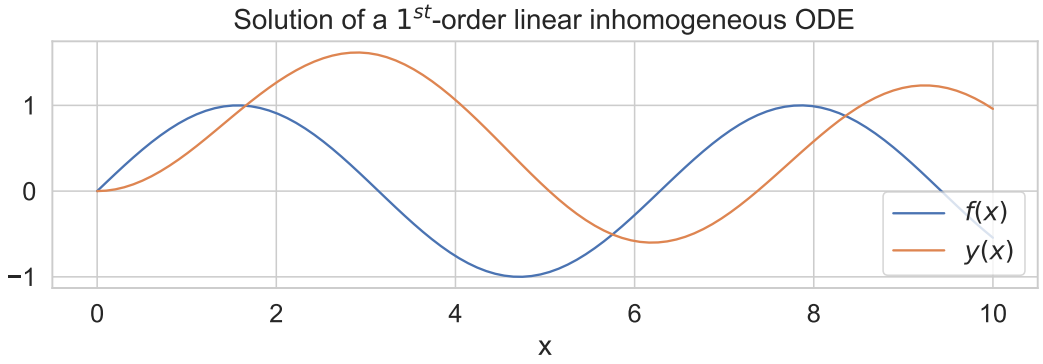


Figure 2.5: Solution of a 1st-order linear inhomogeneous ODE. The ODE parameters are $a = 0.15$, which we shall call the *decay* parameter, and $f(x) = \sin(x)$ which we call *forcing* function. We numerically solve the initial value problem with initial conditions $y(0) = 0$ and plot the solution $y(x)$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

2.4.1 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.

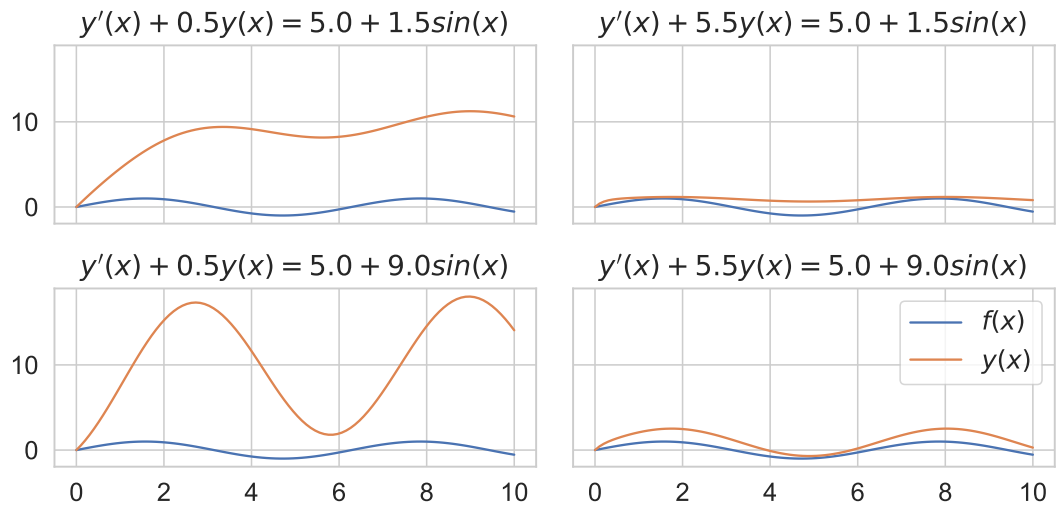


Figure 2.6: The four panels show the latent force/forcing term of the LFM (blue line) as well as the numerical solution of the ODE problem (orange line). We give the solver boundary conditions $y(0) = 0$. Each panel uses a different combination of parameters. Notice how increasing the decay rate D greatly 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 [10]. 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.

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

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

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

Here $\operatorname{erf}(x) = \int_0^x \exp(-u^2)du$ 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 $f(x')$.

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

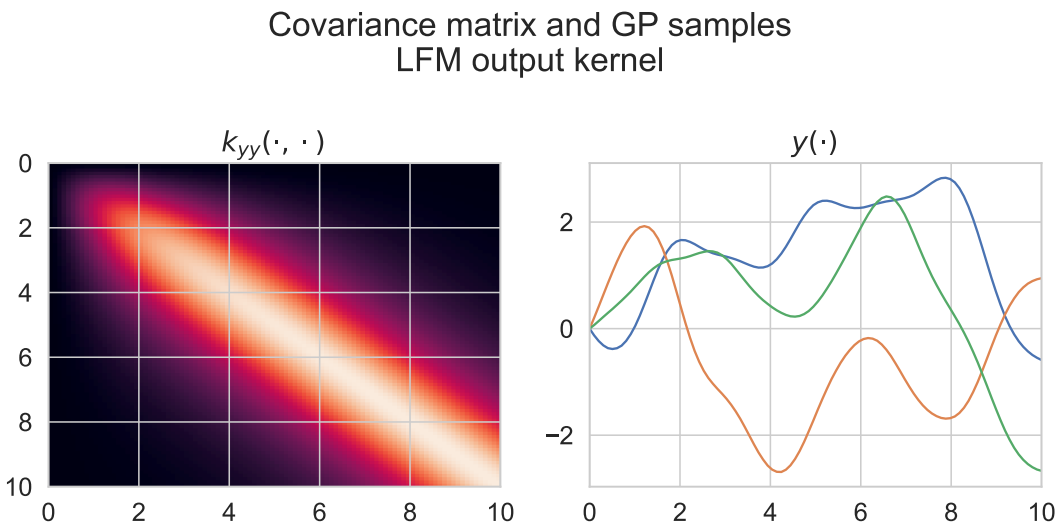


Figure 2.7: Left: Covariance matrix of a GP with a LFM output kernel. In this chart, the kernel uses parameters $B = 0, D = 0.5, S = 1.5, \ell = 1.5$. Observe how, in the top-left corner, all values are very close to zero. Right: samples from the zero-mean GPs using this covariance matrix. Notice how, 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.

kernel:

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

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

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

In figure 2.8 we see Gaussian Process Regression (GPR) with a GP that used the covariance function that we have derived. The latent force function is chosen

to be $\sin(x)$. Then we use an ODE solver that solves the LFM ODE problem to generate the data. To the resulting ODE solution, we add i.i.d. Gaussian noise with 0.5 standard deviation. 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:

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

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.

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-driven, 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 [6]. 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 $\mathbf{t} = \{t_m \mid m = 1, \ldots, M\}$ be a set of *treatment times* i.e. the time at which a specific treatment was administered to the patient.

We write the differential equation of the LFM describing the time evolution of the physiological quantity under the effect of M independent treatments f:

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

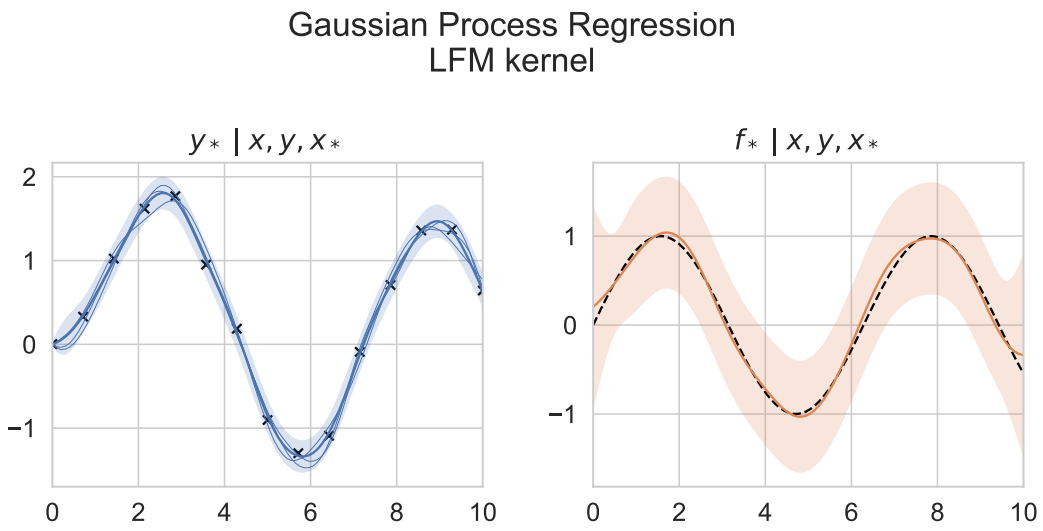


Figure 2.8: The chart shows an example of Gaussian Process Regression (GPR) using a GP kernel based on a Latent Force Model. The data, shown as black crosses, is generated by giving the $\sin(x)$ function to an ODE solver that solves the LFM ODE problem. Then, to the resulting ODE solution, we add i.i.d. 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.

2.5.2 Time-marked Latent Forces

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 [6]:

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

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.

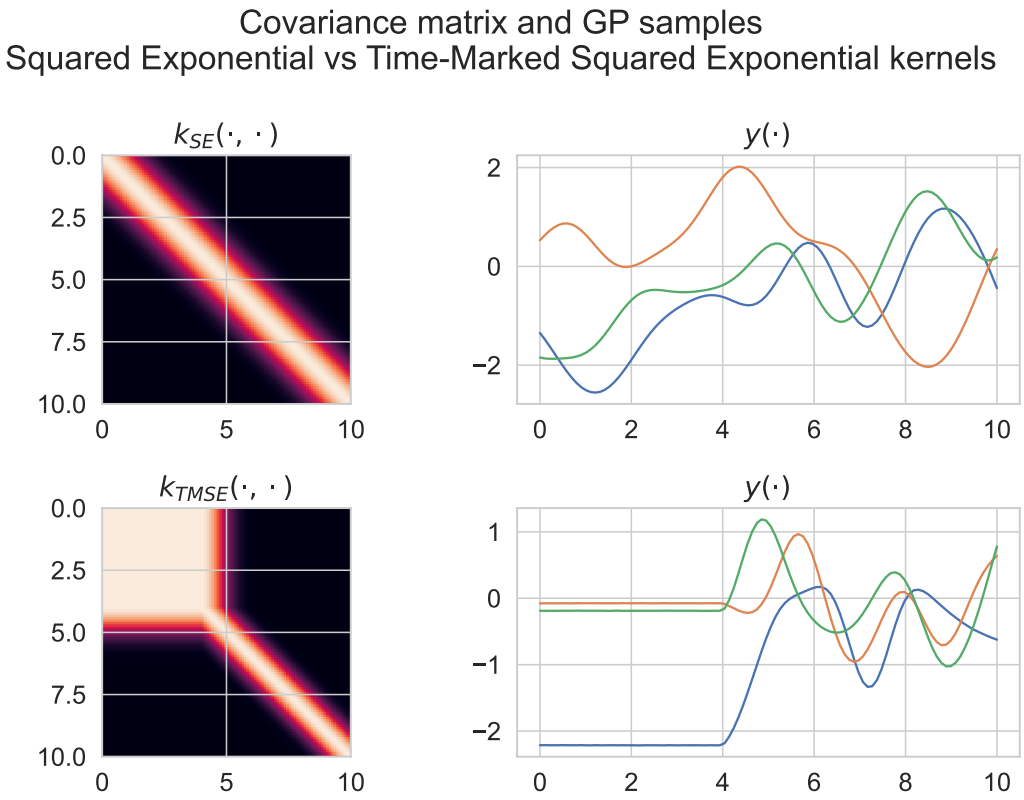


Figure 2.9: Top left: covariance matrix for a squared exponential (SE) kernel with length scale $\ell = 1$. Top right: samples from a GP with zero mean and SE kernel. Bottom left: covariance matrix for a time-marked squared exponential (TMSE) kernel with length scale $\ell = 1$ and treatment time $t_m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

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

kernel.

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

We have obtained a formulation similar to the single-force LFM. The formula derived in Section 2.4 cannot be reused here for two reasons: The latent force kernel k_{ff} is a time-marked squared exponential instead of a squared exponential. There are M latent forces instead of a single one.

While it is possible to obtain a closed-form solution, we will neither present the steps nor the final formulas here. The same goes for the "cross-covariance" [6] output-latent kernel. For more details check the 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 then it behaves like a squared exponential GP convolved with an exponential function.

2.5.4 Limitations

Let us now discuss some limitations of this model. While applying to a real-world data set, we have found four main issues:

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

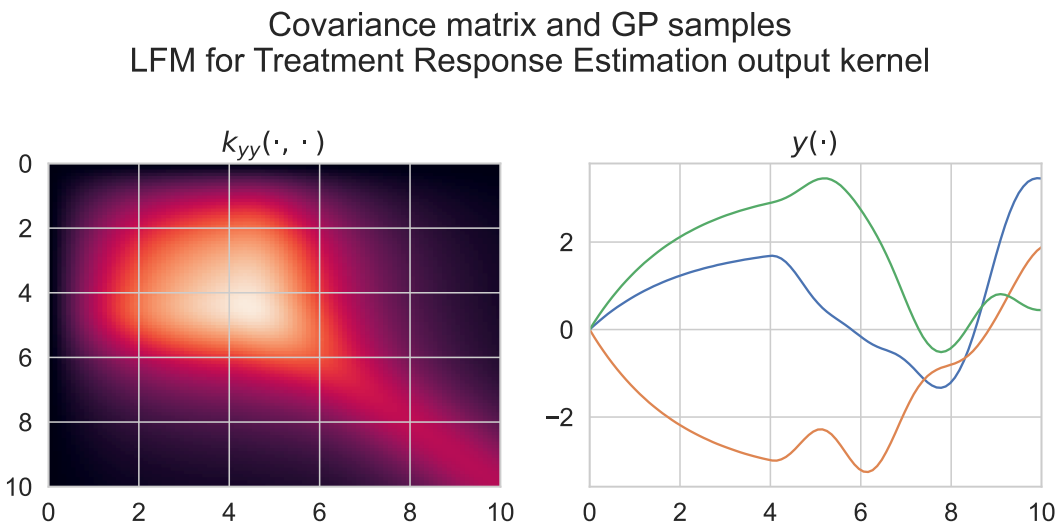


Figure 2.10: Left: covariance matrix for the k_{yy} output kernel with parameters $B = 0, D = 0.5, S = 1.5, t_m = 4, \ell = 1$. Right: samples from the output kernel.

- 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 [15].
- 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.

Chapter 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 $f(\tau)$ 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* [4, 16]. Let $f_t(\tau)$ be a continuous function that models the treatment response to a treatment. This function is the main focus of this dissertation and it is known in the literature as the *treatment response curve* [2]. Different treatment response curve models will have different f_t formulations.

The generative model we have chosen for the physiological quantity is

$$f(\tau) = f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m),$$

The treatment response functions f_t are additive both within themselves and with the baseline function.

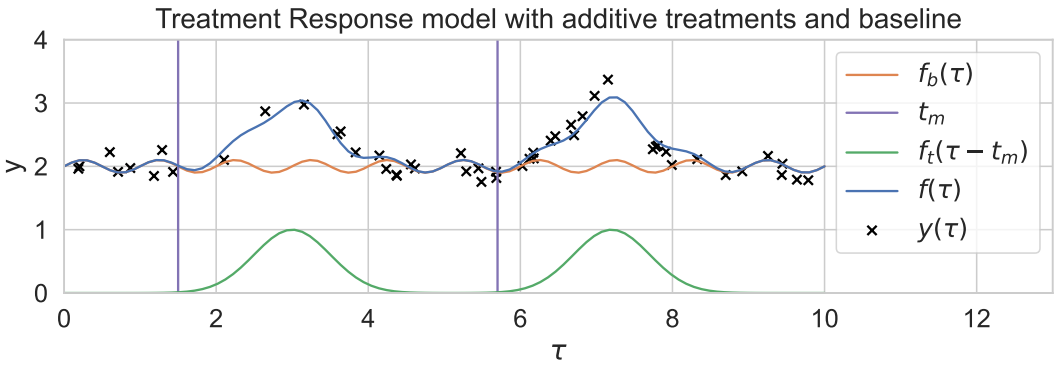


Figure 3.1: Illustration of the distinct functions in the model and how they are combined to form the final prediction. In this example, the physiological quantity $f(\tau)$ is the sum of a sinusoidal baseline $f_b(\tau) = \sin(2\pi\tau)$ and treatment response curves $f_t(\tau; t_m) = \exp\{-0.5(\tau - t_m - 2)^2/0.5^2\}$. 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 $\sigma = 0.15$.

Let $y(\tau)$ be the noisy physiological quantity. The noise model is zero-mean independent and identically distributed Gaussian noise.

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

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 $\boldsymbol{\tau} = \{\tau_i \mid i = 1, 2, \ldots, N\}$ be irregularly-sampled times, sorted temporally. Let $\mathbf{y} = \{y_i \mid \tau = 1, 2, \ldots, N\}$ be noisy observations of the physiological quantity that we want to estimate the effect of a treatment on. Every observation y_i occurs 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 $\mathcal{D} = \{\boldsymbol{\tau}, \mathbf{y}, \mathbf{t}, \mathbf{x}\}$.

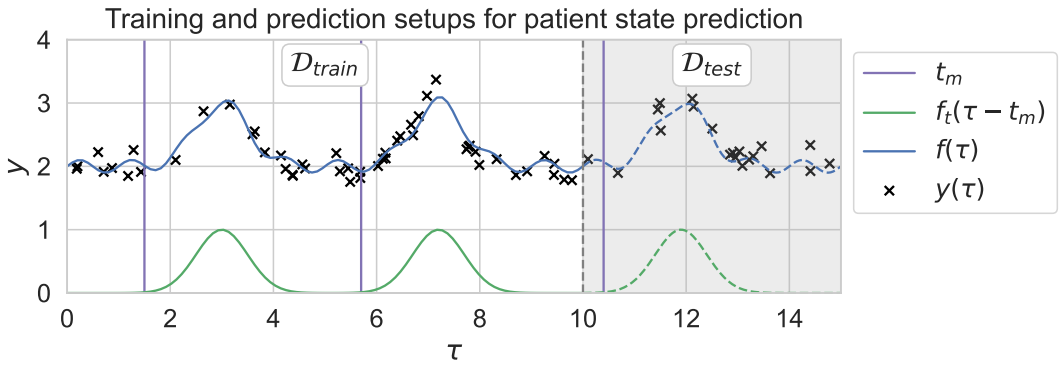


Figure 3.2: Illustration of how the model is trained on the training dataset $\mathcal{D}_{\text{train}}$, and then tested using the test dataset $\mathcal{D}_{\text{test}}$. The only difference between the two datasets is that in the test dataset we do not have access to $\mathbf{y}_{\text{test}}$.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

3.3 Task

Our goal is to estimate the values of $f(\tau)$, the noiseless physiological quantity, for any future time point τ_{test}, given new treatment times $\mathbf{t}_{\text{test}}$ and treatment covariates $\mathbf{x}_{\text{test}}$. In order to evaluate the performance on unseen data, we split the dataset $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{text}}$, using a time-series holdout split, such that $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$. Given the historical dataset $\mathcal{D}_{\text{train}} = \{\boldsymbol{\tau}_{\text{train}}\mathbf{y}_{\text{train}}, \mathbf{t}_{\text{train}}, \mathbf{x}_{\text{train}}\}$. The estimate $f(\tau)$ is then evaluated at $\boldsymbol{\tau}_{\text{test}}$ and compared with $\boldsymbol{y}_{\text{test}}$, the true noisy observations at $\boldsymbol{\tau}_{\text{test}}$. 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 independent between each other, which is known as an *unpooled* formulation [17].

Motivated by the fact that we want to share data across multiple individuals, we now introduce the notation for multiple individuals. 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 an individual p. Then, let $\boldsymbol{y}^{(p)} \in \mathbb{R}^{N^{(p)}}$ the noisy observations, measured at times $\boldsymbol{\tau}^{(p)} \in \mathbb{R}^{N^{(p)}}$. Let $\boldsymbol{t}^{(p)} \in \mathbb{R}^{M^{(p)}}$ be the treatment times, and $\mathbf{x}^{(p)} \in \mathbb{R}^{K \times M^{(p)}}$ 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:

$$f^{(p)}(\tau) = f_b^{(p)}(\tau) + \sum_{m=1}^{M} f_t^{(p)}(\tau - t_m).$$

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

Chapter 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.

$$f(\tau) = f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m),$$

The baseline function f_b is a constant, motivated by the facts that in the dataset we see no clear patterns outside of the ones explained by treatments and that in non-diabetic patients the baseline can be accurately approximated by a constant baseline [18].

$$f_b(\tau) = k,$$

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

$$f_t(\cdot) \sim \mathcal{GP}(0, k_{f_t f_t}(\cdot, \cdot)).$$

4.1.2 Non-independent treatments

Consider two treatments f_t, f_t'. They are two functions sampled from a Gaussian Process $\mathcal{GP}(0, k_{f_t f_t}(\cdot, \cdot))$. In the LFM for Treatment Response Estimation model

[6], the assumption is that f_t and f'_t are two *independent* samples from the GP. Thus $\text{cov}(f_t, f'_t) = 0$, or $f_t, f'_t \overset{\text{iid}}{\sim} \mathcal{GP}(0, k(\cdot,\cdot))$. For our model, we assume that they are the *same* sample from the GP, or $f_t = f'_t, f_t \overset{\text{iid}}{\sim} \mathcal{GP}(0, k(\cdot,\cdot))$.

Thus, if we assume that all treatments are independent, the covariance function is nonzero only between two treatments administered at the same time, which we consider to be the same treatment:

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

whereas, in our non-independent formulation, since we assume that all treatments have the same response function, regardless of administration time, the correlation between the responses is always nonzero:

$$\text{cov}(f_t(\tau - t_m), f_t(\tau' - t_{m'})) = k_{f_t f_t}(\tau - t_m, \tau' - t_{m'}), \forall m, m' \in 1, \ldots, M.$$

4.1.3 Time-Limited Squared Exponential Kernel

The main issue with using a SE kernel for treatment responses is that the GP samples are not time-limited functions. On the other hand, we expect the treatment effect to have a starting time and a finite duration [15]. To model this, we modify the SE kernel to generate functions that are zero before 0 and zero again after T, the treatment's duration. A variant of this procedure has been discussed in [19, 6], to model time-marked data with GPs.

To design a kernel that produces zero-valued functions we take a regular SE kernel and then set its value to zero whenever $\tau, \tau' < 0$ or $\tau, \tau' > T$. We call this *Time-Limited Squared Exponential* kernel (TLSE):

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

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

In Figure 4.1 we compare the covariance matrices and their samples for GPs with SE and TLSE kernels. As expected, the samples for the SE GP have infinite duration, while ones from the TLSE GP have limited duration between the starting time t_m and the final time $t_m + T$.

4.1.4 Output kernel

Let us now derive the output kernel for this model. Since now we are using a single treatment response function instead of M independent ones, the covariance

structure will be different and we cannot reuse the results from Section 2.5.

$$
\begin{aligned}
k_{ff}(\tau, \tau') &= \operatorname{cov}(y(\tau), y(\tau')) \\
&= \operatorname{cov}\left[f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m), f_b(\tau') + \sum_{m'=1}^{M'} f_t(\tau' - t_{m'}) \right] \\
&= \mathbb{E}\left[\left(\sum_{m=1}^{M} f_t(\tau - t_m) \right) \left(\sum_{m'=1}^{M'} f_t(\tau' - t_{m'}) \right) \right] \\
&= \sum_{m=1}^{M} \sum_{m'=1}^{M'} \mathbb{E}(f_t(\tau - t_m) f_t(\tau' - t_{m'})) \\
&= \sum_{m=1}^{M} \sum_{m'=1}^{M'} k_{f_t f_t}(\tau - t_m, \tau' - t_{m'}).
\end{aligned}
$$

Comparing it with the output kernel from Section 2.5 we see that here we have two summations over t_m whereas the other model only had one. This is because here we are sharing the same response function for all treatments whereas the other model considers all treatment responses as independent.

4.1.5 Limitations

This is the simplest model that has been developed as part of this dissertation. While its simplicity is an attractive factor, mainly thanks to its interpretability and performance, it has two limitations:

- **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.

- **Performance**. Because one single response function is shared across all treatments, the covariance function must consider the effect of all treatments when computing each treatment. In other words, each treatment depends on all of the other ones, which means that for M treatments, computing the covariance between any two time point requires $O(M^2)$ operations.

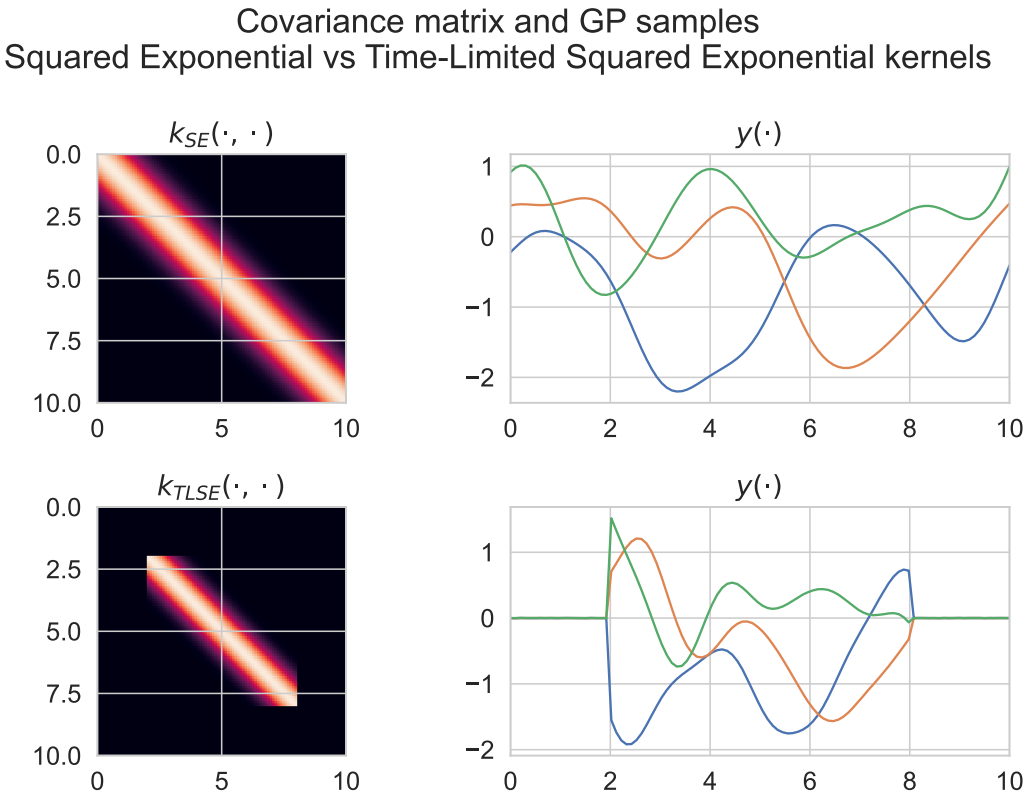


Figure 4.1: Comparing the covariance matrices and 3 GP samples for a Squared Exponential (SE) kernel and Time-Limited Squared Exponential (TLSE) kernel. Both kernels share the same length scale $\ell = 1$. This TLSE kernel is using a treatment duration $T = 6$ and its inputs were delayed by $t_m = 2$, 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.

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 [6], with some crucial modifications that we argue are required for adequate modeling of treatment.

Our main contribution is to update the existing SOTA [6] model to include the two constraints developed in our first model.

4.2.1 Model Definition

Starting from our common generative model:

$$f(\tau) = f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m),$$

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

The treatment response function f_t is inspired by LFM models and it is defined as the solution to an Ordinary Differential Equation. We define that the time evolution of the treatment function f_t depends on its current value scaled by a *decay rate* term D plus a *latent force* $f_l(\tau)$ scaled by a *sensitivity* term S:

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

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 [7] or a time-marked squared exponential kernel [6], but instead our newly developed time-limited squared exponential kernel (TLSE).

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

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:

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

Finally, we assume that there is only one latent force, sampled from a GP, that is shared by all treatments. On the other hand, other works [7, 6] assume that every treatment uses a separate latent force and that all latent forces are independent between each other.

In Figure 4.2, we compare the covariance matrix and samples for the latent forces and outputs between two models. The first model is an 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.

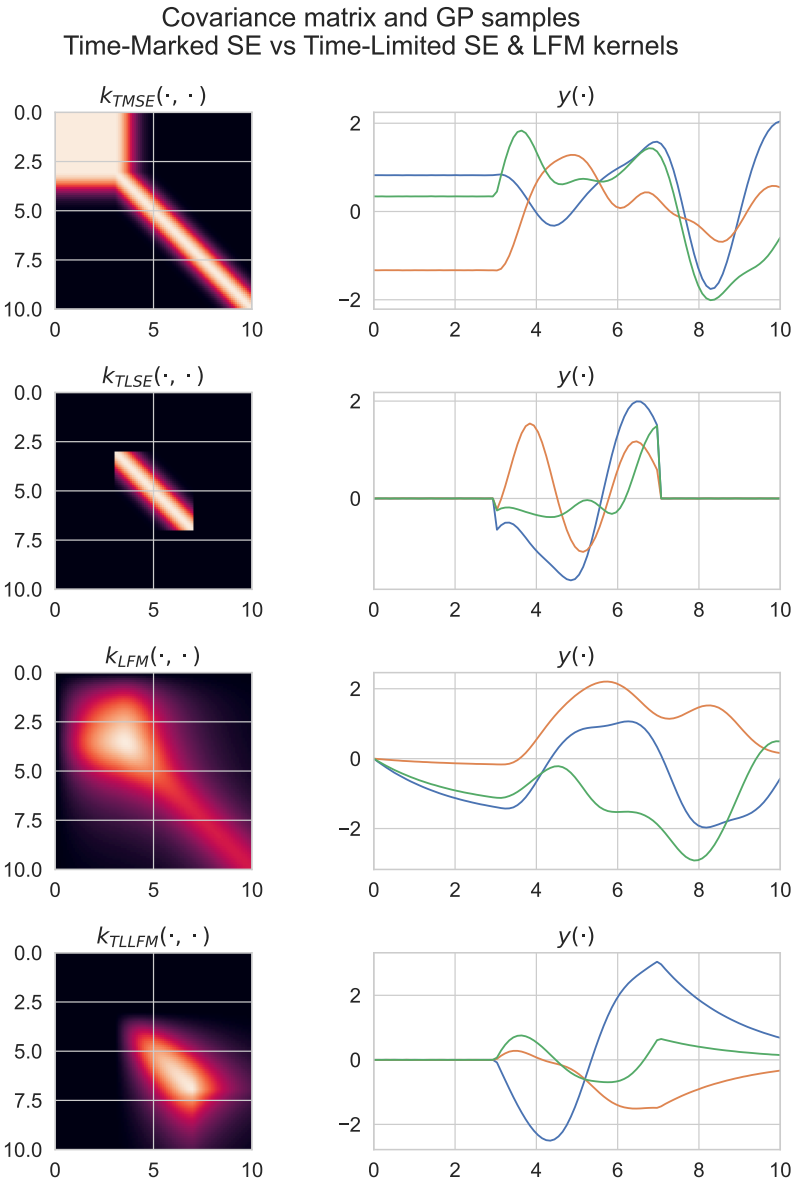


Figure 4.2: Comparison between Time-Marked (TM) and Time-Limited (TL) SE kernels for Latent Forces and LFM outputs. The first two panels display the two kernels used for the LFM's latent forces. Inside the $[t_m, t_m + T]$ interval the samples from the two kernels, TMSE and TLSE, behave identically. Before t_m the TMSE samples are constant while the TLSE samples are zero. After $t_m + T$ there is SE behavior when using the TMSE kernel. Using the TLSE, by contrast, all the samples are zero after $t_m + T$. The differences in the latent forces influence the LFM output kernels too. The LFM kernel uses TM latent forces, the TLLFM kernel uses TL latent forces. For the TMSE kernel, before t_m there is exponential decay behaviour, while for the TLSE kernel the LFM output is zero just like its latent forcing function. After $t_m + T$ the LFM kernel does not change while the TLLFM kernel shows exponential decay, due to the latent forces being zero. 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 $S = 1.5$.

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 three flaws:

- **Stronger Assumptions of LFM model**. The stronger assumptions introduced by the mechanistic ODE model allow the model to extrapolate beyond the observation better than a purely data-driven model. But, by using LFMs, we are forcing the model to learn latent functions inside a linear, inhomogeneous, constant coefficient ODE. This is a very strong assumption about the data generating process and it could be a very bad approximation of the real phenomenon.
- **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. Because of this, fitting the model 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 the 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 each one of the treatments f_t. The treatment covariates x_m are defined next. The scaling factor is a function of the covariates x_m of the m-th treatment:

$$f(\tau) = f_b(\tau) + \sum_{m=1}^{M} S(x_m) f_t(\tau - t_m).$$

4.3.1 Linear Scaling

Let $\mathbf{x}$ 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 dose 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.

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

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 assume that the treatment responses of two separate individuals are not independent of each other. With this assumption, the predictions for one individual can be improved by sharing statistical strength 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 add support for multiple individuals in the generative model defined in Section 3.1 using the framework of Multi-Output Gaussian Processes.

Let P be the number of individuals and let p be a specific individual. We define the generating process for the p-th individual as:

$$\begin{aligned} &f^{(p)}(\tau) : \mathbb{R} \to \mathbb{R} \\ &f^{(p)}(\tau) = f_b^{(p)}(\tau) + \sum_{m=1}^{M} f_t^{(p)}(\tau - t_m), \end{aligned}$$

If the set of treatment functions $\{f_t^{(p)}\}_{p=1}^{P}$ contains P independent functions, then we call this a *separate* treatment response model. If the set contains the same single function f_t for all p, i.e. one single treatment response curve for all treatments and individuals, then we call this the *shared* model.

Let us now consider the MOGP $\boldsymbol{f}(\boldsymbol{\tau})$ defined as:

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{\tau}) &: \mathbb{R}^P \to \mathbb{R}^P \\ \boldsymbol{f}(\boldsymbol{\tau}) &= \begin{bmatrix} f^{(1)}(\tau^{(1)}) \\ f^{(2)}(\tau^{(2)}) \\ \vdots \\ f^{(P)}(\tau^{(P)}) \end{bmatrix}, \end{aligned}$$

Then we can write the matrix-valued covariance function:

$$\text{cov}(\mathbf{f}(\boldsymbol{\tau}), \mathbf{f}(\boldsymbol{\tau}')) : \mathbb{R}^P \times \mathbb{R}^P \to \mathbb{R}^{P \times P}$$
$$\text{cov}(\mathbf{f}(\boldsymbol{\tau}), \mathbf{f}(\boldsymbol{\tau}')) =$$
$$\begin{bmatrix} \text{cov}(f^{(1)}(\tau^{(1)}), f^{(1)}(\tau'^{(1)})) & \dots & \text{cov}(f^{(1)}(\tau^{(1)}), f^{(P)}(\tau'^{(P)})) \\ \vdots & \ddots & \vdots \\ \text{cov}(f^{(P)}(\tau^{(P)}), f^{(1)}(\tau'^{(1)})) & \dots & \text{cov}(f^{(P)}(\tau^{(P)}), f^{(P)}(\tau'^{(P)})) \end{bmatrix}.$$

Notice that $\tau^{(p)} \neq \tau'^{(p')}$. The observations are not *homotopic*, but rather *heterotopic* i.e. not aligned. If they were aligned, or homotopic, we could exploit the block structure of the matrix to reduce the amount of computation [12].

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.

4.4.2 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:

$$f^{(p)}(\tau) = f_b^{(p)}(\tau) + \sum_{m=1}^{M} S^{(p)}(\mathbf{x_m}) f_t(\tau; t_m),$$
$$S^{(p)}(x_m) = (\beta^{(p)})^T x_m + \gamma^{(p)}.$$

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.

$$\begin{aligned}\beta^{(p)} &\sim \mathcal{N}(\mu_\beta, \sigma_\beta)\\ \gamma^{(p)} &\sim \mathcal{N}(\mu_\gamma, \sigma_\gamma).\end{aligned}$$

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

$$\begin{aligned}\mu_\beta^{(p)} &\sim \mathcal{N}(\nu_\beta, \tau_\beta)\\ \mu_\gamma^{(p)} &\sim \mathcal{N}(\nu_\gamma, \tau_\gamma),\\ \beta^{(p)} &\sim \mathcal{N}(\mu_\beta^{(p)}\sigma_\beta^{(p)}),\\ \gamma^{(p)} &\sim \mathcal{N}(\mu_\gamma^{(p)}\sigma_\gamma^{(p)}).\end{aligned}$$

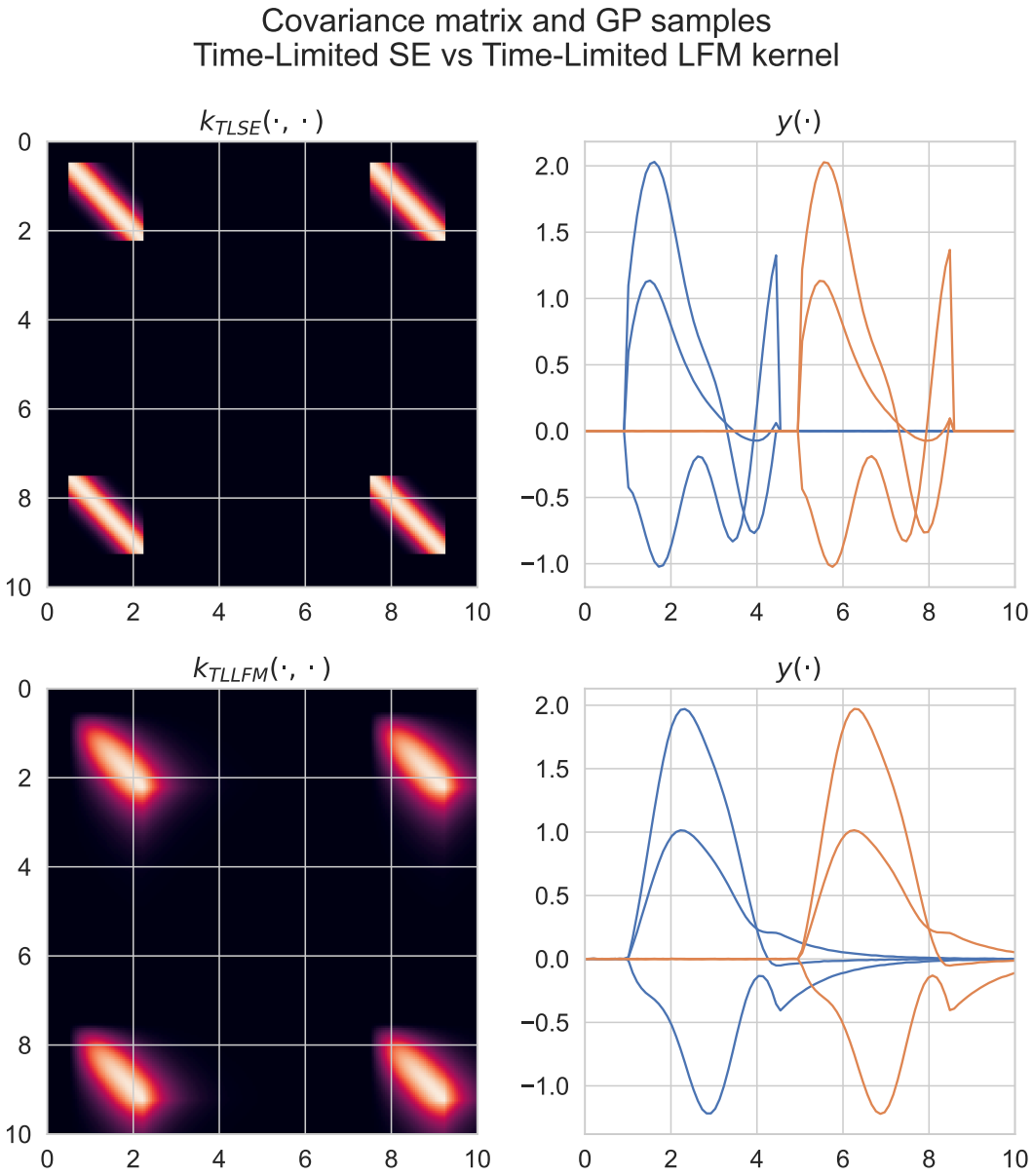


Figure 4.3: Comparison of the covariance matrices and GP samples generated by the multi-output kernel. The kernel is evaluated on two individuals, each one receiving one treatment at times 1 and 5, respectively. Two base kernels are used: a time-limited SE Kernel and a time-limited LFM kernel. Both kernels 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.

Chapter 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.

$$
\begin{aligned}
f(\tau) &= f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m) \\
f_b(\tau) &= 0 \\
f_t'(\tau) &= B - D f_t(\tau) + S f_l(\tau) \\
f_l(\tau) &= \exp\left(-\frac{1}{2}\frac{\tau^2}{1^2}\right),
\end{aligned}
$$

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 $\sigma = 0.05$.

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

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 [6]. 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 constant function, instead of a GP with a squared exponential plus periodic kernel.

$$f(\tau) = f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m),$$
$$f_b(\tau) = k,$$
$$f_t'(\tau) = -D f_t(\tau) + S f_l(\tau),$$
$$f_l(\cdot) \sim \mathcal{GP}(0, k_{\text{TMSE}}(\cdot, \cdot)).$$

- **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.

$$f(\tau) = f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m),$$
$$f_b(\tau) = k,$$
$$f_t(\cdot) \sim \mathcal{GP}(0, k_{\text{TLSE}}(\cdot, \cdot)).$$

- **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.

$$
\begin{aligned}
f(\tau) &= f_b(\tau) + \sum_{m=1}^{M} f_t(\tau - t_m), \\
f_b(\tau) &= k, \\
f_t'(\tau) &= -D f_t(\tau) + S f_l(\tau), \\
f_l(\cdot) &\sim \mathcal{GP}(0, k_{\text{TLSE}}(\cdot, \cdot)).
\end{aligned}
$$

5.1.3 Results

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

- **Time-Marked Latent Force**

  The model successfully fits the dataset. This is the expected behaviour 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.

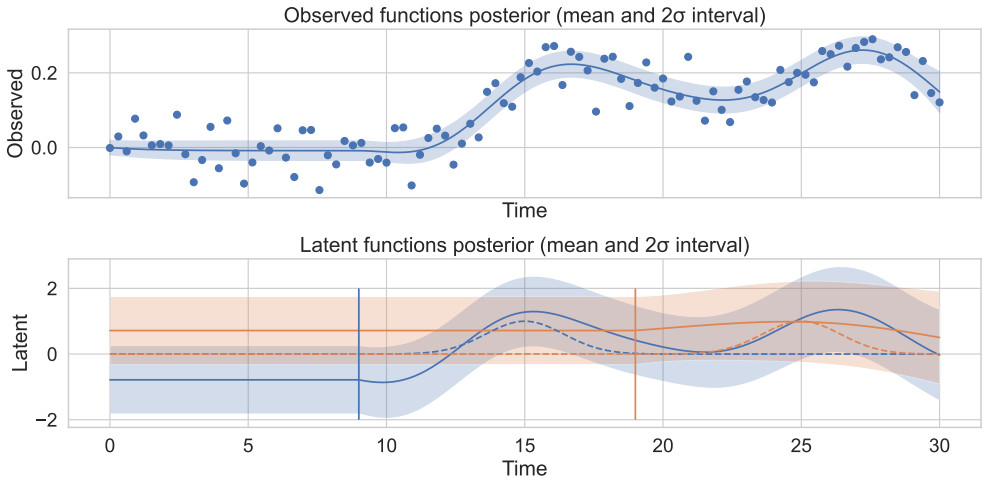
  Furthermore, we stress the fact that since all latent forces are assumed to be independent, it is not clear what the prediction setup would look like.

  Finally, the uncertainty of the latent forces' means is 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.

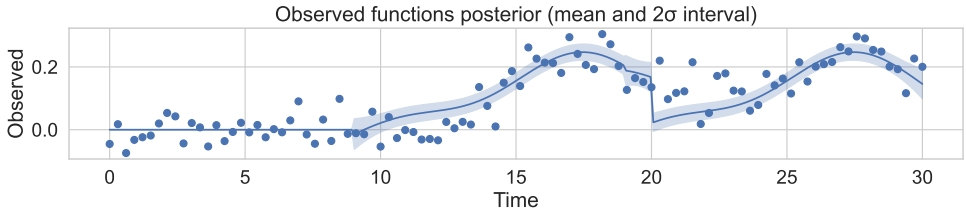
- **Time-Limited Treatment Response**

  The model fits the dataset, albeit the fit is worse than the other two. For this model, latent forces plot are not available as the treatment response is directly modelled with a time-limited SE GP.

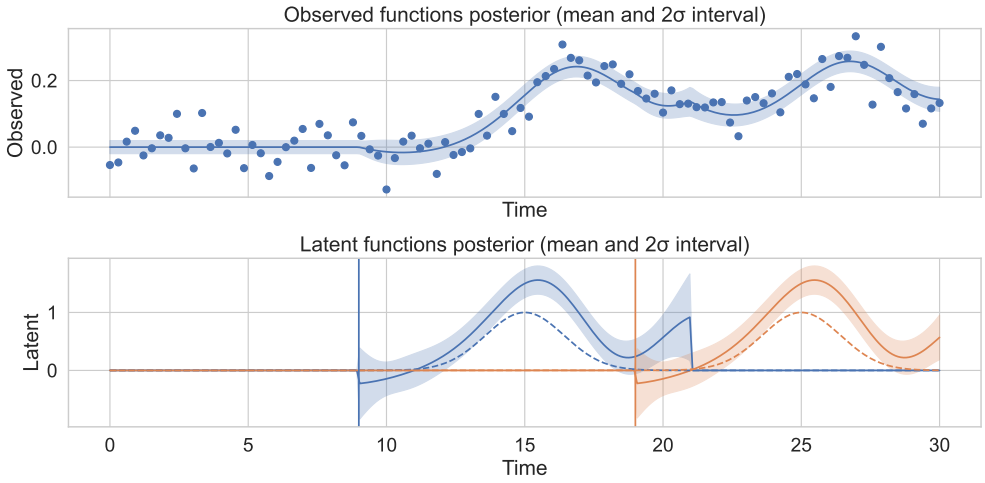
  Introducing a constraint on the support of the treatment responses shows its benefits immediately with this model. The two latent forces are not "fighting" each other anymore, which introduced a lot of variance in the last model.



(a) Time-Marked Latent Force



(b) Time-Limited Treatment Response



(c) Time-Limited Latent Force

Figure 5.1: Comparison of model fits on simulated data. The three groups show one distinct model each: Time-Marked Latent Force (TMLF), Time-Limited Treatment Response (TLTR), Time-Limited Latent Force (TLLF). The first panel of each group displays the posterior mean and 95% credible interval for the physiological quantity f. The second panel, only available for latent force models, displays the posterior mean and 95% credible interval for the latent forces f_t. In the second panel, the true latent forces that were used to simulate the dataset are plotted as dashed lines. Finally, the vertical lines indicate the treatment times $\boldsymbol{t}$.

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**

  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 the time $\tau \in [19, 21]$, 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 experiment, we can draw the following conclusions.

First, Time-Marked Treatment Response model, while being a great foundation for treatment response curve estimation, has two critical flaws: First, the latent forces have infinite duration. Second, the latent forces are independent between each other. 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.

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

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

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 compare the predictive performance against a baseline model, described in [4], which we call EiV model. This model uses a parametric bell-shaped treatment response function, and a GP model that combines constant and squared exponential kernels for the baseline function. Unlike our proposed models, the EiV model uses a sophisticated Errors-in-Variables (hence the name) or measurement error model in order to be robust to the many sources of noise of this dataset.

We begin by comparing the Time-Limited Treatment Response model against the Time-Limited Latent Force model. Because both of these models use a time-limited response or force, we will drop it from their name and refer to the Time-Limited Treatment Response Model as SE-ITR and the Time-Limited Latent Force model as LF-ITR. 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 SE-ITR and LF-ITR 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 SE-ITR 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 suggest that, for this dataset, the

SE-ITR model's errors are lower than the LF-ITR model. Additionally, we have found that learning the hyperparameters of the LF-ITR model is challenging and that the optimization's results heavily depend on the initial conditions. This is because even small changes in the ODE parameters such as the decay rate or the sensitivity cause big variations in the response curve's shape and magnitude.

In light of the model comparison results, we run a second set of experiments using the best performing model: SE-ITR. 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 SE-ITR with identical 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 SE-ITR models with fixed scaling coefficients. The bottom panel compares the MSEs of two SE-ITR 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.

Model	Treatment Response Curve	Scaling Coefficients	mMSE
EiV [4]			0.738
SE-ITR	separate	fixed	0.640
SE-ITR	separate	unpooled	0.641
LF-ITR	separate	fixed	0.670
LF-ITR	separate	unpooled	1.221
SE-ITR	shared	fixed	0.572
SE-ITR	shared	unpooled	0.573
SE-ITR	shared	pooled	**0.564**
SE-ITR	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 SE kernel for modeling the treatment response curves, shared treatment response curve between patients, and a single set of scaling coefficients for all patients (unpooled scaling coefficients).

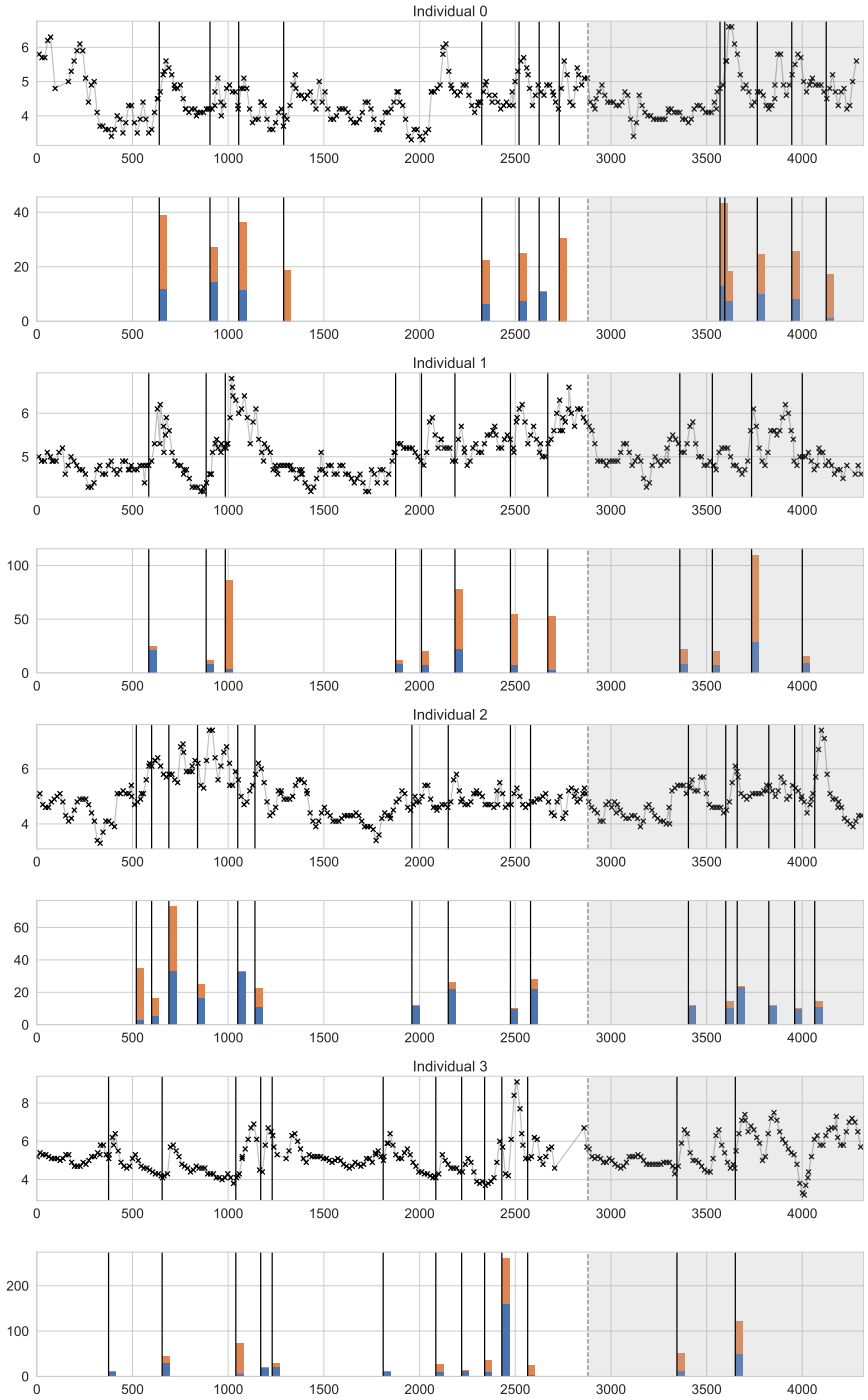


Figure 5.2: Visualization of the 3-days of blood glucose dataset. For each pair of panels, the top panel displays the blood glucose observation time-series with black signs joined by solid gray lines. The treatment times appear on all panels as solid black vertical lines. The bottom panels displays the 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.

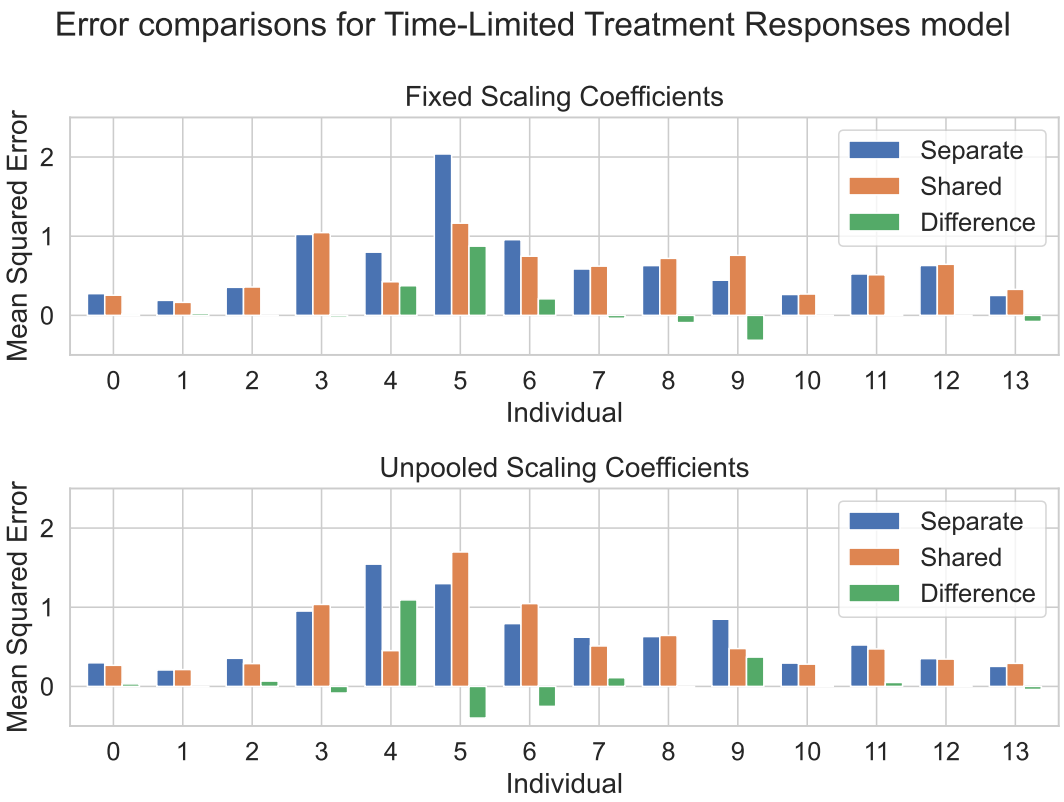
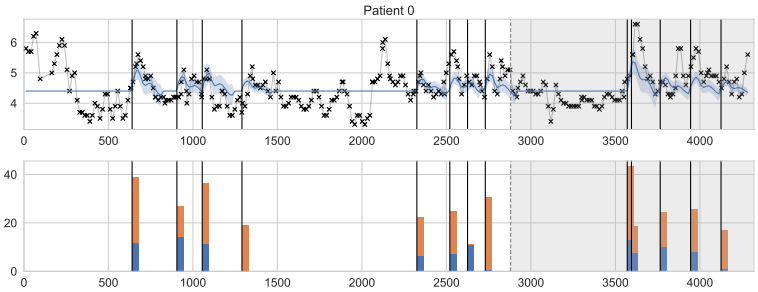
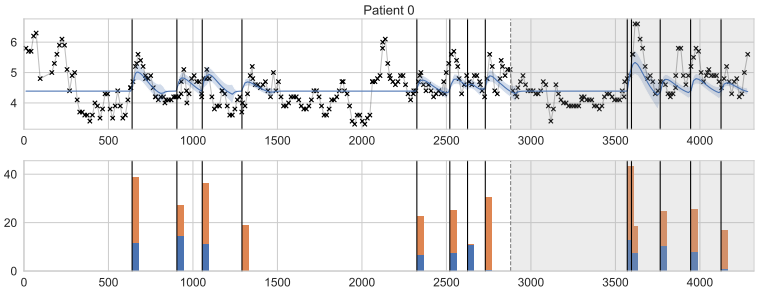


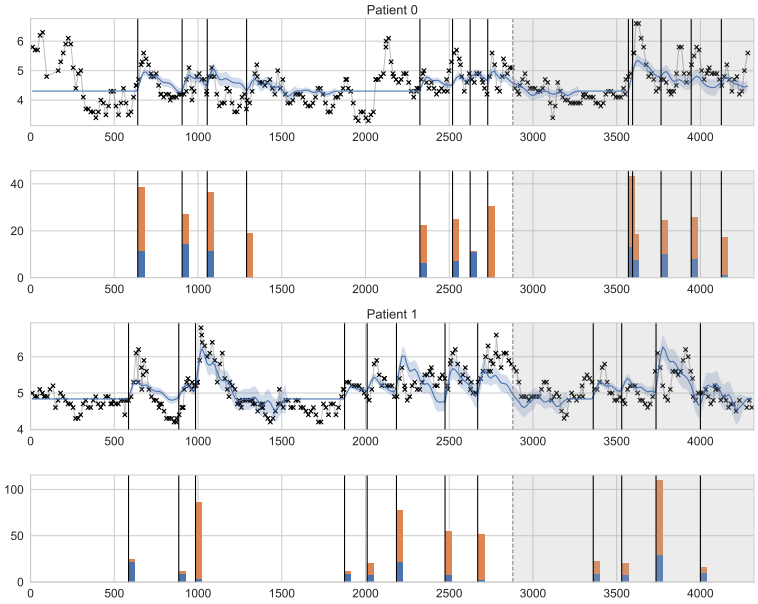
Figure 5.3: Comparison of the Mean Squared Errors (MSEs) for every individual. The top panel compares the MSEs of two SE-ITR models with fixed scaling coefficients and the bottom panel compares the MSEs of two SE-ITR models with unpooled scaling coefficients.
The difference between separate and shared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



(a) Time-Limited Treatment Response model (SE-ITR), separate treatment response curves, unpooled scaling coefficients.



(b) Time-Limited Latent Force model (LF-ITR), separate treatment response curves, unpooled scaling coefficients.



(c) Time-Limited Treatment Response model (SE-ITR), shared treatment response curves, pooled scaling coefficients.

Figure 5.4: Prediction of blood glucose level after a meal. The first panel of each group displays with a solid blue line the posterior mean and 95% credible interval for the blood glucose levels. The black vertical lines mark the treatment times. The second panel displays the amounts of sugar and starch for every meal, referred to as treatment covariates. The last group contains two pairs of panels because the model was trained on two patients at the same time, in order to share the treatment response curve and scaling coefficients.

Chapter 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, we warn about the risks of biased estimates and overfitting on datasets with 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] J. S. Beckmann and D. Lew. «Reconciling evidence-based medicine and precision medicine in the era of big data: challenges and opportunities». In: *Genome Med* 8.1 (Dec. 2016), p. 134 (cit. on p. 1).

[2] Yanbo Xu, Yanxun Xu, and Suchi Saria. «A Non-parametric Bayesian Approach for Estimating Treatment-Response Curves from Sparse Time Series». In: *Proceedings of the 1st Machine Learning for Healthcare Conference*. Ed. by Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens. Vol. 56. Proceedings of Machine Learning Research. Northeastern University, Boston, MA, USA: PMLR, 2016, pp. 282–300. URL: https://proceedings.mlr.press/v56/Xu16.html (cit. on pp. 1, 22).

[3] M. M. Ghassemi, S. E. Richter, I. M. Eche, T. W. Chen, J. Danziger, and L. A. Celi. «A data-driven approach to optimized medication dosing: a focus on heparin». In: *Intensive Care Med* 40.9 (2014). [PubMed Central:PMC4157935] [DOI:10.1007/s00134-014-3406-5] [PubMed:12655395], pp. 1332–1339 (cit. on p. 1).

[4] Guangyi Zhang, Reza A. Ashrafi, Anne Juuti, Kirsi Pietiläinen, and Pekka Marttinen. «Errors-in-Variables Modeling of Personalized Treatment-Response Trajectories». In: *IEEE Journal of Biomedical and Health Informatics* 25.1 (2021), pp. 201–208. DOI: 10.1109/JBHI.2020.2987323 (cit. on pp. 1, 22, 43, 45).

[5] Hossein Soleimani, Adarsh Subbaswamy, and Suchi Saria. *Treatment-Response Models for Counterfactual Reasoning with Continuous-time, Continuous-valued Interventions*. 2017. DOI: 10.48550/ARXIV.1704.02038. URL: https://arxiv.org/abs/1704.02038 (cit. on p. 1).

[6] Li-Fang Cheng, Bianca Dumitrascu, Michael Zhang, Corey Chivers, Michael Draugelis, Kai Li, and Barbara Engelhardt. «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 (cit. on pp. 1, 2, 17, 18, 20, 27, 29, 30, 38).

[7] 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 (cit. on pp. 2, 12, 30).

[8] 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 (cit. on pp. 2, 12).

[9] M. Ghassemi, T. Naumann, P. Schulam, A. L. Beam, I. Y. Chen, and R. Ranganath. «A Review of Challenges and Opportunities in Machine Learning for Health». In: *AMIA Jt Summits Transl Sci Proc* 2020 (2020), pp. 191–200 (cit. on p. 2).

[10] 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 (cit. on pp. 4, 14).

[11] 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 (cit. on p. 6).

[12] 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 (cit. on pp. 8, 10, 34).

[13] W.E. Boyce, R.C. DiPrima, and D.B. Meade. *Boyce's Elementary Differential Equations and Boundary Value Problems*. Wiley, 2017. ISBN: 9781119390756. URL: https://books.google.fi/books?id=pi1EDwAAQBAJ (cit. on p. 11).

[14] Jacob D. Moss, Felix L. Opolka, Bianca Dumitrascu, and Pietro Lió. *Approximate Latent Force Model Inference*. 2021. DOI: 10.48550/ARXIV.2109.11851. URL: https://arxiv.org/abs/2109.11851 (cit. on p. 12).

[15] Daniel F. B. Wright, Helen R. Winter, and Stephen B. Duffull. «Understanding the time course of pharmacological effect: a PKPD approach». In: *British Journal of Clinical Pharmacology* 71.6 (2011), pp. 815–823. DOI: https://doi.org/10.1111/j.1365-2125.2011.03925.x. eprint: https://bpspubs.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2125.2011.03925.x. URL: https://bpspubs.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2125.2011.03925.x (cit. on pp. 21, 27).

[16] Peter Schulam and Suchi Saria. «Reliable Decision Support using Counterfactual Models». In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/299a23a2291e2126b91d54f3601ec162-Paper.pdf (cit. on p. 22).

[17] Andrew Gelman et al. *Bayesian Data Analysis*. 3rd ed. Chapman and Hall/CRC, 2013. URL: /bib/gelman/Gelman2003/BDA3.pdf,/bib/gelman/Gelman2003/%28Texts%20in%20Statistical%20Science%29%20Andrew%20Gelman%2C%20John%20B.%20Carlin%2C%20Hal%20S.%20Stern%2C%20Donald%20B.%20Rubin-Bayesian%20Data%20Analysis-Chapman%20and%20Hall_CRC%20%282004%29.pdf,https://github.com/avehtari/BDA_course_Aalto (cit. on p. 24).

[18] Reza A. Ashrafi, Aila J. Ahola, Milla Rosengård-Bärlund, Tuure Saarinen, Sini Heinonen, Anne Juuti, Pekka Marttinen, and Kirsi H. Pietiläinen. «Computational modelling of self-reported dietary carbohydrate intake on glucose concentrations in patients undergoing Roux-en-Y gastric bypass versus one-anastomosis gastric bypass». In: *Annals of Medicine* 53.1 (2021). PMID: 34714211, pp. 1885–1895. DOI: 10.1080/07853890.2021.1964035. eprint: https://doi.org/10.1080/07853890.2021.1964035. URL: https://doi.org/10.1080/07853890.2021.1964035 (cit. on p. 26).

[19] John Cunningham, Zoubin Ghahramani, and Carl Rasmussen. «Gaussian Processes for time-marked time-series data». In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Neil D. Lawrence and Mark Girolami. Vol. 22. Proceedings of Machine Learning Research. La Palma, Canary Islands: PMLR, 2012, pp. 255–263. URL: https://proceedings.mlr.press/v22/cunningham12.html (cit. on p. 27).