Hilbert space methods to approximate Gaussian processes using Stan

ARTICLE HISTORY
Compiled August 26, 2019

ABSTRACT

KEYWORDS
Gaussian processes; Low-rank Gaussian processes; Hilbert Space methods; Sparse Gaussian processes

Contents

1. Introduction

Gaussian processes (GPs) are flexible statistical models for specifying probability distributions over multi-dimensional non-linear functions [Neal, 1997, Rasmussen and Williams, 2006]. Their name stems from the fact that any finite set of function values is jointly distributed as a multivariate Gaussian. GPs are defined by a mean and a covariance function. The covariance function encodes our prior assumptions about the functional relationship, such as continuity, smoothness, periodicity and scale properties. GPs not only allow for non-linear effects but can also implicitly handle interactions between covariates. Different types of covariance functions can be combined for further increased flexibility. Due to their generality and flexibility, GPs are of broad interest across machine learning and statistics [Neal, 1997, Rasmussen and Williams, 2006]. Among others, they find application in the fields of spatial epidemiology [Carlin et al., 2014, Diggle, 2013], robotics and control [Deisenroth et al., 2015], signal processing [Särkkä et al., 2013], as well as Bayesian optimization and probabilistic numerics [Briol et al., 2015, Hennig et al., 2015, Roberts, 2010].

Given n observations in the data, the computational complexity and memory requirements of exact GP implementation in general scale as $O(n^3)$ and $O(n^2)$, respectively. This limit their application to rather small data sets of a few tens of thousands observations at most. The problem becomes more severe when performing full Bayesian inference via sampling methods, where in each sampling step we need $O(n^3)$ computations when inverting the Gram matrix of the covariance function, usually through Cholesky factorization. To alleviate these computational demands, several approximate methods have been proposed.

Sparse GPs are based on low-rank approximations of the covariance matrix. The low-rank approximation with $m \ll n$ *inducing points* implies reduced memory requirements of $O(nm)$ and corresponding computational complexity of $O(nm^2)$. A unifying view on sparse GPs based on approximate generative methods is provided in Quiñonero-Candela and Rasmussen [2005], while a general review can be found in Rasmussen and Williams [2006]. Burt et al. [2019] show that for regression with normally distributed covariates in D dimensions and using the squared exponential covariance function, $M = O(\log^D N)$ is sufficient for accurate approximation.

An alternative class of low-rank approximations is based on forming a basis function approximation with $m \ll n$ basis functions. The basis functions are usually presented explicitly, but can also be used to form a low rank covariance matrix approximation. Common basis function approximations rest on the spectral analysis and series expansions of Gaussian processes [Adler, 1981, Cramér and Leadbetter, 2013, Loève, 1977, Trees, 1968]. Sparse spectrum GPs are based on a sparse approximation to the frequency domain representation of a GP [Gal and Turner, 2015, Lázaro Gredilla, 2010, Quiñonero-Candela et al., 2010]. Recently, Hensman et al. [2017] presented a variational Fourier feature approximation for Gaussian processes that was derived for the Matérn class of kernels. Another related method for approximating kernels relies on random fourier features [Rahimi and Recht, 2008, 2009]. Further, certain spline smoothing basis functions are equivalent to GPs with certain covariance functions [Furrer and Nychka, 2007, Wahba, 1990].

Paul: Can we clarify how our discussed method relates to the former paragraph, which introduces all sorts of basis function approximations. I.e. is our method just one out of many or what makes it special? Perhaps we can somehow use the information provided in: While Sparse Spectrum GP is based on a sparse spectrum, the reduced-rank method proposed in this paper aims to make the spectrum as 'full' as possible at a given rank. Recent Splines models can reproduce the Matern family of covariance functions (see, e.g., Wood [2003]), however our approach can reproduce basically all of the stationary covariance functions.

In this paper we focus on the basis function approximation via Laplace eigenfunctions

for stationary covariance functions proposed by Solin and Särkkä [2018]. Basis function approaches behave computationally like linear models, which is an attractive property in modular probabilistic programming models where there is a big benefit if approximation specific computation is simple (Paul: What do you mean with the the last part of the sentence?). The Laplace eigenfunctions can be computed analytically and they are independent of the particular choice of the covariance kernel including the hyperparameters. While the pre-computation cost of the basis functions is $O(m^2n)$, the computational cost when only changing the covariance function parameters is $O(mn + m)$.This is a big advantage in terms of speed for iterative algorithms such as Markov chain Monte Carlo (MCMC). Another advantage is the reduced memory requirements of automatic differentation methods used in modern probabilistic programming frameworks (cite!). This is because the memory requirements of automatic differentation rather scale with the computational complexity instead of with the usual memory requirements for the posterior density computation. The basis function approach also provides an easy way to apply the non-centered parameterization of GPs, which reduces the posterior dependency between parameters representing the estimated function and the hyperparameters of the covariance function, which further improves MCMC efficiency.

Aki: I moved this from section 2, needs to be combined with other intro We propose an approximate framework for fast and accurate inference for Gaussian processes. Using a basis function expansion, we approximate the Gaussian process with a linear model. This representation has three main advantages: 1) it makes inference considerably faster due to the linear structure, 2) it is simple to implement, which makes it easy to use Gaussian processes as building blocks in more complicated models and can be used as latent function in non-Gaussian observational models allowing modelling flexibility, 3) it can be made arbitrary accurate and the trade-off between computational complexity and approximation accuracy can easily be controlled.

While Solin and Särkkä [2018] have fully developed the mathematical theory behind this specific approximation of GPs, further work is needed for its practical implementation in probabilistic programming frameworks such as Stan [Carpenter et al., 2017]. In this paper, we analyze in detail the performance and accuracy of the method in relation to key factors such as the number of basis functions, desired prediction space, or properties of the true functional relationship between covariates and response variable. We provide intuitive visualizations and practical recommendations for the choice of these factors, which will help users to improve computational performance while maintaining close approximation to exact GPs.

Although there are several GP specific software packages available to date (GPML, GPstuff, GPy, GPflow, cite them!), each provide efficient implementations only for a restricted range of GP based models. In this paper, we do not focus on the fastest possible inference for some specific GP models, but instead are interested in how GPs can be easily used as modular components in probabilistic programming frameworks.

The remainder of the paper is structured as follows. In Section 2, we introduce GPs and their reduced rank approximations proposed by Solin and Särkkä [2018]. In Section 3, we analyze the accuracy of these approximations under several conditions using analytical and numerical methods. Several case studies in which we fit exact and approximate GPs to real and simulated data are provided in Section 4. We end with a discussion in Section 5.

2. Method

2.1. Gaussian process as a prior

A Gaussian process (GP) is a stochastic process which defines the distribution over a collection of random variables indexed by a continuous variable, i.e. $\{f(t) : t \in \mathcal{T}\}$ for some index set $\mathcal{T}$. Gaussian processes have the defining property that the marginal distribution of any finite subset of random variables, $\{f(t_1), f(t_2), \ldots, f(t_K)\}$, is a multivariate Gaussian distribution.

In this work, Gaussian processes will take the role of a prior distribution over function spaces for non-parametric latent functions in a Bayesian setting. Consider a data set $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where y_n is modelled conditionally as $p(y_n|f(\boldsymbol{x}_n), \phi)$, where p is some parametric distribution with parameters f and ϕ, and f is an unknown function with Gaussian process prior. This generalizes trivially to more complex models depending on several unknown functions, for example such as $p(y_n|f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$ or multilevel models. Our goal is to obtain posterior distribution for the value of the function $f^* = f(\boldsymbol{x}^*)$ evaluated at a new input point $\boldsymbol{x}^* \in \mathbb{R}^D$.

We assume a Gaussian process prior for $f \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$, where $\mu : \mathbb{R}^D \to \mathbb{R}$ and $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ are the mean and covariance functions, respectively,

$$\begin{aligned} \mu(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})]\,, \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}\big[(f(\boldsymbol{x}) - \mu(\boldsymbol{x}))\,\big(f(\boldsymbol{x}') - \mu(\boldsymbol{x}')\big)\big]\,. \end{aligned}$$

The mean and covariance functions completely characterize the Gaussian process prior, and control the a priori behavior of the function f. Let $\boldsymbol{f} = \{f(\boldsymbol{x}_n)\}_{n=1}^N$, then the resulting prior distribution for $\boldsymbol{f}$ is a multivariate Gaussian distribution $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$, where $\boldsymbol{\mu} = \{\mu(\boldsymbol{x}_n)\}_{n=1}^N$ is the mean and $\boldsymbol{K}$ the covariance matrix, where $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ might depend on a set of hyperparameters, $\boldsymbol{\theta}$, but we will not write this dependency explicitly to ease the notation. The joint distribution of $\boldsymbol{f}$ and a new f^* is also a multivariate Gaussian as,

$$p(\boldsymbol{f}, f^*) = \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{f} \\ f^* \end{array}\right] \middle| \mathbf{0}, \left[\begin{array}{cc} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} & \boldsymbol{k}_{\boldsymbol{f},f^*} \\ \boldsymbol{k}_{f^*,\boldsymbol{f}} & k_{f^*,f^*} \end{array}\right]\right),$$

where $\boldsymbol{k}_{\boldsymbol{f},f^*}$ is the covariance between $\boldsymbol{f}$ and f^*, and k_{f^*,f^*} is the prior variance of f^*.

If $p(y_n|f(\boldsymbol{x}_n), \phi) = N(y_n|f(\boldsymbol{x}_n), \sigma)$ then $\boldsymbol{f}$ can be integrated out analytically (with a computational cost of $O(n^3)$ for exact GP and $O(nm^2)$ for sparse GP). If $p(y_n|f(\boldsymbol{x}_n), \phi) = N(y_n|f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$ or $p(y_n|f(\boldsymbol{x}_n), \phi)$ is non-Gaussian, the marginalization does not have closed form solution. Furthermore, if a prior distribution is imposed on ϕ and θ to form a joint posterior for ϕ, θ and $\boldsymbol{f}$, approximate inference such as Markov chain Monte Carlo (MCMC) [Brooks et al., 2011] Laplace approximation ([Rasmussen and Williams, 2006, Williams and Barber, 1998], expectation propagation [Minka, 2001], or variational Bayes methods [Csató et al., 2000, Gibbs and MacKay, 2000] need to be used. In this paper we focus on use of MCMC for integrating over the joint posterior. MCMC is not usually the fastest approach, but allows accurate inference for general models in probabilistic programming setting. We consider the computational costs of GPs specifically from this point of view.

2.2. Covariance function and spectral density

The covariance function is the crucial ingredient in a Gaussian process as it encodes our prior assumptions about the function, and defines a correlation structure which characterize the correlations between function values at different inputs. A stationary covariance function is

a function of $\boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}' \in \mathbb{R}^D$, such that it can be written $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{\tau})$, which means that the covariance is invariant to translations. Isotropic covariance functions are those that are function of the distance between observations, $k(\boldsymbol{x}, \boldsymbol{x}') = k(|\boldsymbol{x} - \boldsymbol{x}'|) = k(r), r \in \mathbb{R}$, which means that the covariance is both translation and rotation invariant. The most commonly used distance between observations is the norm L2 ($|\boldsymbol{x} - \boldsymbol{x}'|_{L2}$), also known as Euclidean distance, although other types of distances can be considered.

The Matérn class of isotropic covariance functions is given by,

$$k_\nu(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu} r}{\ell} \right),$$

where ν is the order the kernel, K_ν the modified Bessel function, and the ℓ and σ are the length-scale and magnitude, respectively, of the kernel. The particular case where $\nu = \infty$ and $\nu = 3/2$ are probably the most commonly used kernels [Rasmussen and Williams, 2006],

$$k_\infty(r) = \sigma^2 \exp\left(-\frac{1}{2} \frac{r^2}{\ell^2} \right),$$
$$k_{\frac{3}{2}}(r) = \sigma^2 \left(1 + \frac{\sqrt{3} r}{\ell} \right) \exp\left(-\frac{\sqrt{3} r}{\ell} \right).$$

The former is commonly known as squared exponential (exponentiated quadratic) covariance function. Assuming the Euclidean distance between observations, $r = |\boldsymbol{x} - \boldsymbol{x}'|_{L2} = \sqrt{\sum_{i=1}^D (x_i - x_i')^2}$, the kernels written above take the form

$$k_\infty(|\boldsymbol{x} - \boldsymbol{x}'|_{L2}) = \exp\left(-\frac{1}{2} \sum_{i=1}^D \frac{(x_i - x_i')^2}{\ell_i^2} \right),$$
$$k_{\frac{3}{2}}(|\boldsymbol{x} - \boldsymbol{x}'|_{L2}) = \left(1 + \sqrt{\sum_{i=1}^D \frac{3(x_i - x_i')^2}{\ell_i^2}} \right) \exp\left(-\sqrt{\sum_{i=1}^D \frac{3(x_i - x_i')^2}{\ell_i^2}} \right).$$

Notice that the previous expressions have been easily generalized to using a multidimensional length-scale $\boldsymbol{\ell} \in \mathbb{R}^D$. The use of a multidimensional length-scale basically turns the isotropic covariance function into non-isotropic.

Stationary covariance functions can be represented in terms of their spectral densities [Rasmussen and Williams, 2006]. In this sense, the covariance function of a stationary process can be represented as the Fourier transform of a positive finite measure (*Bochner's theorem*, see, e.g. Akhiezer and Glazman [1993]). If this measure has a density, it is known as the spectral density of the covariance function, and the covariance function and the spectral density are Fourier duals, known as the *Wiener-Khintchine theorem* [Rasmussen and Williams, 2006]. The spectral density functions associated with the Matérn class of covariance functions is given by

$$S_\nu(\omega) = \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu) l^{2\nu}} \left(\frac{2\nu}{l^2} + 4\pi^2 \omega^2 \right)^{(\nu + D/2)}$$

in D dimensions, where variable $\omega \in \mathbb{R}$ is a distance in the frequency domain, and ℓ and σ are

the lengthscale and magnitud, respectively, of the kernel. The particular cases where $\nu = \infty$ and $\nu = 3/2$ take the form

$$S_{\infty}(\omega) = \sigma^2\sqrt{2\pi}^{D}\ell^{D}\exp\left(-0.5\ell^2\omega^2\right), \tag{1}$$

$$S_{\frac{3}{2}}(\omega) = \sigma^2\frac{2^D\pi^{D/2}\Gamma(\frac{D+3}{2})\sqrt{3}^3}{\frac{1}{2}\sqrt{\pi}\ell^3}\left(\frac{3}{\ell^2}+\omega^2\right)^{-\frac{D+3}{2}}. \tag{2}$$

Particularizing to an input dimension $D = 3$ and Euclidean distance $\omega = \sqrt{\sum_{i=1}^{D=3} s_i^2}$, and considering a multidimensional lengthscale $\boldsymbol{\ell} \in \mathbb{R}^{D=3}$, the spectral densities written above take the form

$$S_{\infty}(\omega) = \sigma^2\sqrt{2\pi}^{D=3}\prod_{i=1}^{D=3}\ell_i\exp\left(-\frac{1}{2}\sum_{i=1}^{D=3}\ell_i^2 s_i^2\right),$$

$$S_{\frac{3}{2}}(\omega) = \sigma^2 32\pi\sqrt{3}^3\prod_{i=1}^{D=3}\ell_i\left(3+\sum_{i=1}^{D=3}\ell_i^2 s_i^2\right)^{-3}.$$

2.3. Hilbert space approximate Gaussian process model

The approximate Gaussian process method, developed by Solin and Särkkä [2018] and implemented in this paper, is based on considering the covariance operator of a homogeneous (stationary) covariance function as a pseudo-differential operator constructed as a series of Laplace operators. Then, the pseudo-differential operator is approximated with Hilbert space methods on a compact subset $\Omega \subset \mathbb{R}^D$ subject to some boundary condition. For brevity, we will refer to these approximate Gaussian processes as HSGPs. Below, we will present the main results around HSGPs relevant for practical application. More details and mathematical proofs are provided in Solin and Särkkä [2018].

We begin by focusing on the case of a unidimensional input space (i.e., on GPs with just a single covariate) such that $\Omega \in [-L, L] \subset \mathbb{R}$, where L is some positive real value to which we also refer as boundary condition. As Ω describes the interval in which the approximations are valid, L plays a critical role in the accuracy of HSGPs. We will come back to this issue in Section 3.

Within Ω, we can write any stationary covariance function with input values $\{x, x'\} \in \Omega$ as

$$k(x, x') = \sum_{j=1}^{\infty} S_{\theta}(\sqrt{\lambda_j})\phi_j(x)\phi_j(x'), \tag{3}$$

where S_θ is the spectral density of the stationary covariance function k (see Section 2.1) and θ the set of hyperparameters of k [Rasmussen and Williams, 2006]. The terms $\{\lambda_j\}_{j=1}^{\infty}$ and $\{\phi_j(x)\}_{j=1}^{\infty}$ are the sets of eigenvalues and eigenvectors, respectively, of the Laplacian operator. Namely, they satisfy the following eigenvalue problem in Ω when applying the Dirichlet boundary condition (other boundary conditions could be used as well):

$$\begin{aligned} -\nabla^2\phi_j(x) &= \lambda\phi_j(x), && x \in \Omega \\ \phi_j(x) &= 0, && x \notin \Omega. \end{aligned} \tag{4}$$

The eigenvalues $\lambda_j > 0$ are real and positive because the Laplacian is a positive definite

Hermitian operator, and the eigenfunctions ϕ_j for the eigenvalues problem in Equation (4) are sinusoidal functions. Independently of the covariance function, they can be computed as

$$\lambda_j = \left(\frac{j\pi}{2L}\right)^2, \tag{5}$$

$$\phi_j(x) = \sqrt{\frac{1}{L}} \sin\left(\sqrt{\lambda_j}(x+L)\right). \tag{6}$$

If we truncate the sum in (3) to the first m terms, the approximate covariance function becomes

$$k(x,x') \approx \sum_{j=1}^{m} S_\theta(\sqrt{\lambda_j})\phi_j(x)\phi_j(x') = \boldsymbol{\phi}(x)^\intercal \Delta \boldsymbol{\phi}(x'),$$

where $\boldsymbol{\phi}(x) = \{\phi_j(x)\}_{j=1}^m \in \mathbb{R}^m$ is the column vector of basis functions, and $\Delta \in \mathbb{R}^{m\times m}$ is the diagonal matrix of the spectral densities $S_\theta(\sqrt{\lambda_j})$:

$$\Delta = \begin{bmatrix} S_\theta(\sqrt{\lambda_1}) & & \\ & \ddots & \\ & & S_\theta(\sqrt{\lambda_m}) \end{bmatrix}.$$

Thus, the Gram matrix K of the covariance function k for a set of observations $i = 1, \ldots, n$ and corresponding input values $\{x_i\}_{i=1}^n \in \Omega^n$ can be represented as

$$\mathrm{K} = \Phi \Delta \Phi^\intercal,$$

where $\Phi \in \mathbb{R}^{n\times m}$ is the matrix of eigenfunctions $\phi_j(x_i)$:

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_m(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_m(x_n) \end{bmatrix}.$$

As a result, the model for f can be written as

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \Phi\Delta\Phi^\intercal).$$

This equivalently leads to a linear representation of f via

$$f(x) \approx \sum_{j}^{m} \left(S_\theta(\sqrt{\lambda_j})\right)^{1/2} \phi_j(x)\beta_j, \tag{7}$$

where $\beta_j \sim \text{Normal}(0,1)$. Thus, the function f is approximated with a finite basis function expansion (using the eigenfunctions ϕ_j of the Laplace operator), scaled by the square root of spectral density values. A key property of this approximation is that the eigenfunctions ϕ_j do not depend on the covariance hyperparameters θ. Instead, the only dependence on θ is through the spectral density S_θ. The eigenvalues λ_j are monotonically increasing with j and S_θ goes rapidly to zero for bounded covariance functions. Therefore, Equation (7) can be

expected to be a good approximation for a finite number of m terms in the series as long as the inputs values x_i are not too close to the boundaries $-L$ and L of Ω. The computational cost of univariate HSGPs scales as $O(nm + m)$, where n is the number of observations and m the number of basis functions.

The parameterization in (7) is naturally in the non-centered parameterization form with independent prior distribution on β_j, which makes the posterior inference easier.

2.4. Generalization to multidimensional GPs

The results from the previous section can be generalizes to a multidimensional input space with compact regular domain $\Omega = [-L_1, L_1] \times \cdots \times [-L_d, L_d]$ and Dirichlet boundary conditions. In a D-dimensional input space, the total number of eigenfunctions and eigenvalues in the approximation is equal to the number of D-tuples, that is possible combinations of univariate eigenfunctions over all dimensions. The number of D-tuples is given by

$$m^* = \prod_{d=1}^{D} m_d, \tag{8}$$

where m_d is the number of basis function for the dimension d. Let $\mathbb{S} \in \mathbb{N}^{m^* \times D}$ be the matrix of all those D-tuples. For example, suppose we have $D = 3$ dimensions and use $m_1 = 2$, $m_2 = 2$ and $m_3 = 3$ eigenfunctions and eigenvalues for the first, second and third dimension, respectively. Then, the number of multivariate eigenfunctions and eigenvalues is $m^* = m_1 \cdot m_2 \cdot m_3 = 12$ and the matrix $\mathbb{S} \in \mathbb{N}^{12\times 3}$ is given by

$$\mathbb{S} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 1 & 3 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix}$$

Each multivariate eigenfunction ϕ_j^* corresponds to the product of the univariate eigenfunctions whose indices corresponds to the elements of the D-tuple $\mathbb{S}_{j\cdot}$, and each multivariate eigenvalue $\boldsymbol{\lambda}_j^*$ is a D-vector with elements that are the univariate eigenvalues whose indices correspond to the elements of the D-tuple $\mathbb{S}_{j\cdot}$. Thus, for $\boldsymbol{x} = \{x_d\}_{d=1}^{D} \in \Omega$ and $j = 1, \ldots, m^*$, we have

$$\phi_j^*(\boldsymbol{x}) = \prod_{d=1}^{D} \phi_{\mathbb{S}_{jd}}(x_d) = \prod_{d=1}^{D} \sqrt{\frac{1}{L_d}} \sin\left(\sqrt{\lambda_{\mathbb{S}_{jd}}}(x_d + L_d)\right) \tag{9}$$

$$\boldsymbol{\lambda}_j^* = \left\{\lambda_{\mathbb{S}_{jd}}\right\}_{d=1}^{D} = \left\{\left(\frac{\pi \mathbb{S}_{jd}}{2L_d}\right)^2\right\}_{d=1}^{D}. \tag{10}$$

The approximate covariance function is then represented as

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{j=1}^{m^*} S_\theta^* \left(\sqrt{\boldsymbol{\lambda}_j^*}\right) \phi_j^*(\boldsymbol{x}) \phi_j^*(\boldsymbol{x}'), \tag{11}$$

where S_θ^* is the spectral density of the D-dimensional covariance function (see Section 2.1). We can now write the approximate series expansion of the multivariate function f as,

$$f(\boldsymbol{x}) \approx \sum_{j=1}^{m^*} \left(S_\theta^* \left(\sqrt{\boldsymbol{\lambda}_j^*}\right)\right)^{1/2} \phi_j^*(\boldsymbol{x}) \beta_j, \tag{12}$$

where, again, $\beta_j \sim \text{Normal}(0, 1)$. The computational cost of multivariate HSGPs scales as $O(nm^* + m^*)$, where n is the number of observations and m^* is the number of multivariate basis functions. Although this still implies linear scaling in n, the approximation is more costly than in the univariate case, as m^* is the product of the number of univariate basis functions over the input dimensions and grows exponentially with respect to the number of dimensions.

2.5. Learning hyperparameters and model inference

• It has an attractive computational cost as this basically turns the regular GP model into a linear model.

• The design matrix of the proposed linear model, which is composed of a basis of Laplace eigenfunctions, can be computed analytically and does not depend on the hyperparameters of the model, then it has to be computed only once with $O(n + m)$ computational demands.

• The weights associated to the basis functions in this linear model is a m-dimensional vector (m is the number of basis functions) and their computation is an operation with $O(m)$ computational demands. The weights depend on the hyperparameters, then they have to be computed in every step of the HMC sampling method.

• The linear model is computed with complexity $O(nm)$, computed in every step of the HMC sampling method.

• In a fully Bayesian inference framework using sampling methods, the proposed approximate GP model has a computational complexity of $O(nm + m)$ in every step of the HMC method. In addition, the computation of the automatic differentiation to compute the gradients in this linear model scales $O(n)$?, an operation that must be computed in every step of the HMC method.

• Using maximizing marginal likelihood methods, the proposed model has a overall complexity of $O(nm^2)$. After this, evaluating the marginal likelihood and marginal likelihood gradients is an $O(m^3)$ operation in every step of the optimizer. (Arno's paper, pag. 7)

• The parameter posterior distribution in this approximate GP model is m-dimensional ($m << n$) which helps the use of GP priors as latent functions. especially when sampling methods for inference are used. GP prior as latent functions is needed in generalized models.

In regular GPs and other approximate GP models and Splines models these features do not have so nice properties:

• In a regular GPs, the main computational complexity comes from the inversion of the covariance matrix which is in general a $O(n^3)$ operation. This operation has to be computed at

every step of the HMC or optimizer.

• In regular GPs, the parameter posterior distributions is N-dimensional. It is known that when N is of medium or large size there is high correlation between the N-dimensional latent function and the hyperparameters of the GP prior.

• In conventional sparse GP approximations, although the rank of the GP is reduced considerably to the number of inducing points, this still needs to do the autodiff and covariance matrix inversion.

• The Splines models are also a sort of basis functions expansion model, then the computational demands are similar to that in this approach. However in Splines models the lengthscale hyperparameter tend to be fixed and then the fit is covered by the magnitude parameter. In that sense, Splines models tend to loose the useful interpretation of the lengthscale parameter.

• In addition, the computation of the automatic differentiation to compute gradients in this linear model scales $O(n)$, which is an operation that must be computed in every step of the HMC method.

• In a regular GP model the automatic differentiation to compute the gradients of the covariance function scales $O(n^2)$, the dimension of the covariance matrix, and the full inversion of the covariance matrix scales $O(n^3)$. This operation has to be computed at every step of the HMC.

• In a sparse GP approach based on inducing points, although the rank of the GP is reduced considerably to the number of inducing points, this still needs to do the autodiff and covariance matrix inversion.

• The Splines models are also a sort of basis functions expansion model, then the computational demands are similar to that in this approach.

3. The accuracy of the approximation

The accuracy and speed of the HSGP model depends on several interrelated factors, most notably on the number of basis functions and on the boundary condition of the Laplace eigenfunctions. Furthermore, appropriate values for these factors will depend on the non-linearity of the estimated function, which is in turn characterized by the lengthscale of the covariance function. In this section, we analyze the effects of the number of basis functions and the boundary condition on the approximation accuracy. We present recommendations on how they should be chosen and diagnostics to check the accuracy of the obtained approximation.

(Gabi: The following paragraph is a first idea to introduce that the recommendations and diagnostics depends on the kernel considered. And that, at the moment, we have built these recommendations only for the square exponential kernel. I know that what I wrote in not enough and should be a bit more elaborate; your suggestions where and how elaborate it would be useful.)

Ultimately, these recommendations lie on the relationships among the number of basis functions, the boundary factor and the lengthscale of the function, which depend on the particular choice of the kernel function. In this work we built these relationships for the square exponential covariance function and Matern ($\nu = 3/2$) covariance function. For other kernels, the relationships will be slightly different, in function mainly of the smoothness or wigglyness of the kernel effects.

3.1. Dependency on the number of basis functions and the boundary condition

As explained in Section 2, the approximation of the covariance function is a series expansion of eigenfunctions and eigenvalues of the Laplace operator in a given domain Ω, for instance in a one-dimensional input space $\Omega = [-L, L] \subset \mathbb{R}$:

$$k(\tau) = \sum_{j=1}^{\infty} S_\theta \left(\sqrt{\lambda_j}\right) \phi_j(\tau)\phi_j(0),$$

where L describes the boundary condition, j is the index for the eigenfunctions and eigenvalues, and $\tau = x - x'$ is the difference between two covariate values x and x' in Ω. The eigenvalues λ_j and eigenfunctions ϕ_j are given in Equations (5) and (6) for the unidimensional case and in Equations (10) and (9) for the multidimensional case. The number of basis functions can be truncated at some finite positive value m such that the difference between the densities of the exact and approximate covariance functions is less than a predefined threshold $\varepsilon > 0$:

$$\int k(\tau)\, d\tau - \int \sum_{j=1}^{m} S_\theta \left(\sqrt{\lambda_j}\right) \phi_j(\tau)\phi_j(0)\, d\tau < \varepsilon. \tag{13}$$

The finite number m of basis functions in the approximation needed to satisfy Equation (13) depends on the non-linearity of the function to be learned, that is on its lengthscale ℓ, which constitutes a hyperparameter of the GP. The approximation also depends on the boundary L (see Equations (5), (6), (10) and (9)), which will affect its accuracy especially near the boundaries. As we will see later on, L will also influence the number of basis functions required in the approximation. In the present paper, we will set L an extension of the desired covariate input domain Ψ. Without loss of generality, we can assume Ψ to be symmetric around zero, that is $\Psi = [-S, S] \subset \mathbb{R}$. We now define L as

$$L = c \cdot S, \tag{14}$$

where S (for $S > 0$) represents the half-range of the input space, and c (for $c \geq 1$) is the proportional extension factor. In the following, we will to refer to c as the boundary factor of the approximation. The boundary factor can also be regarded as the boundary L normalized by the half-range S of the input space.

We start with an illustration on how the number of basis functions m and boundary factor c influences the accuracy of the HSGP approximations, separately. For this purpose, a set of noisy observations are drawn from an exact GP model with lengthscale $\ell = 0.3$ and marginal variance $\alpha = 1$, using input values from the zero-mean input domain with half-range $S = 1$. Several HSGP models with varying m and L are fitted to this data. In this example, the lengthscale and marginal variance parameters used in the HSGPs are fixed to the true values of the data-generating model. Figures 1 and 2 illustrate the individual effects of m and c, respectively, on the posterior predictions of the estimated function and on the covariance function itself. For c fixed to a large enough value, Figure 1 shows clearly how m affects the accuracy on the approximation and the non-linearity of the estimated function, in the sense that fewer basis functions inaccurately imply larger lengthscales and consequently more linear functional forms. The higher the "wigglyness" of the function to be estimated, the more basis functions will be required. If m fixed to a large enough value, Figure 2 shows that c mainly affects the approximation near the boundaries as well as covariances at long distances.

Next, we will focus on analyzing the interaction effects between these m and c on the

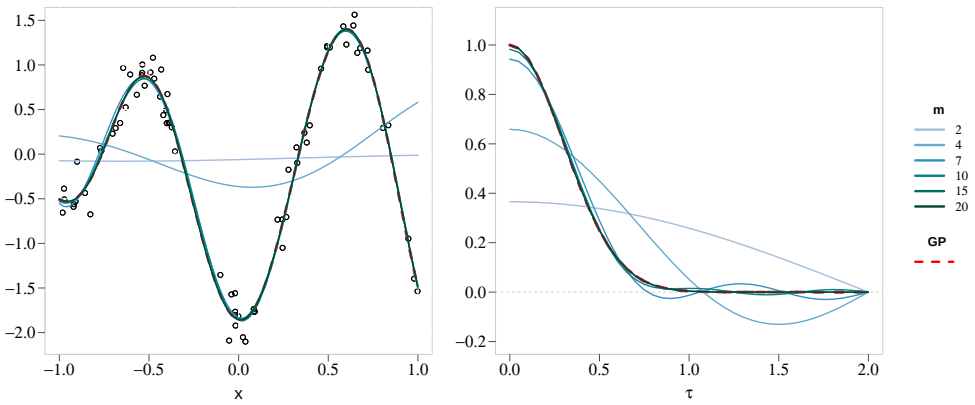

Figure 1. Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model and the HSGP model for different number of basis functions m, with the boundary factor fixed to a large enough value.

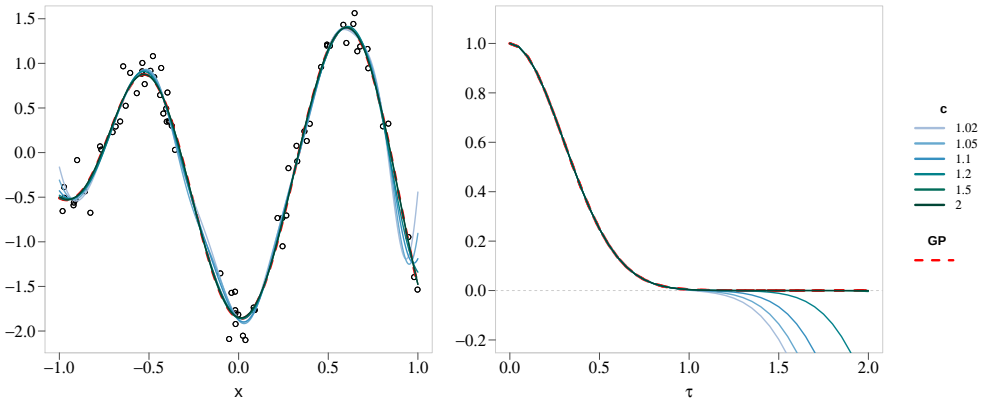

Figure 2. Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model and the HSGP model for different values of the boundary factor c, with a large enough fixed number of basis functions.

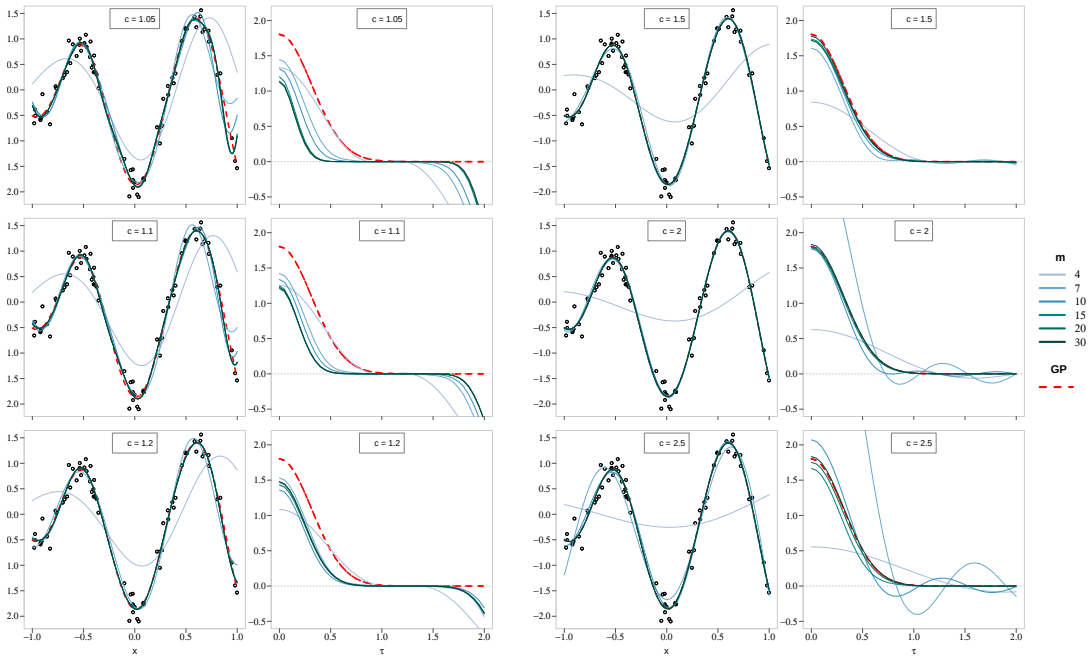

Figure 3. Mean posterior predictive functions (left-first and right-first columns) and covariance functions (left-second and right-second columns) of both the regular GP model and the HSGP model for different number of basis functions m and for different values of the boundary factor c.

performance of the approximation. The lengthscale and marginal variance will no longer be fixed but rather estimated in both regular GP and HSGP models. Figure 3 shows the functional posterior predictions and the covariance function obtained after fitting the data, for varying m and c. Figure 4 shows the root mean square error (RMSE) of the HSGP models, computed against the regular GP model. Figure 5 shows the estimated lengthscale and marginal variance for the regular GP model and the HSGP models. Looking at the RMSEs in Figure 4, we can conclude that the optimal choice in terms of precision and computations would be $m = 15$ basis functions and a boundary factor between $c = 1.5$ and $c = 2.5$. Further, the choice of $m = 10$ and $c = 1.5$ could still be an accurate enough choice. We may also come to the same conclusion by looking at the posterior predictions and covariance function plots in Figure 3. From these results, some general conclusions may be drawn:

- As c increases, m has to increase as well (and vice versa).
- There exists a minimum c below which a close approximation will never be achieved regardless of m.

Additionally, there is a clear relation of the number of basis functions m and the boundary factor c with the lengthscale ℓ of the approximated function. Figures 6 and 7 depicts how these three factors interact with each other in relation to a close approximation of the HSGP model, in the cases of a GP with square exponential covariance function and Matérn ($\nu = 3/2$) covariance function, respectively, and a single input dimension. More precisely, for a given GP model (with a square exponential covariance function) with lengthscale ℓ and given a boundary factor c, Figure 6 shows the minimum m required to achieve a close approximation in terms of satisfying Equation (13). Similarly for Figure 7 in the case of a Matérn ($\nu = 3/2$) covariance function. We have considered an approximation to be a close enough when the difference between densities of the approximate covariance function and the exact covariance function, ε

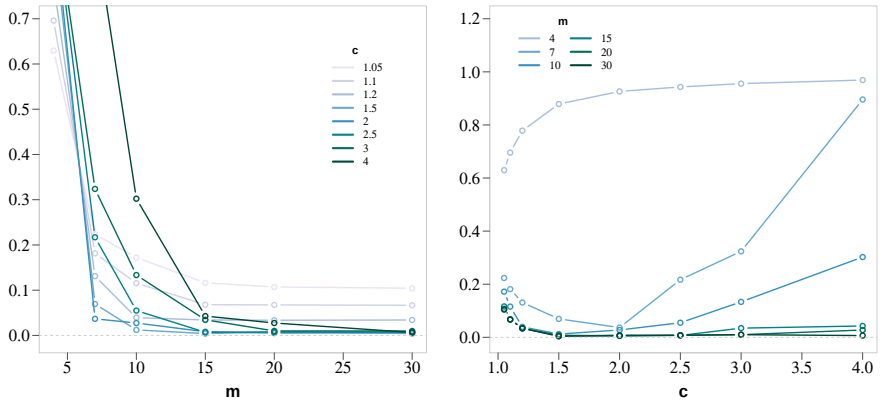

Figure 4. Root mean square error (RMSE) of the proposed HSGP models computed against the regular GP model. (left) RMSE versus the number of basis functions m and for different values of the boundary factor c. (right) RMSE versus the boundary factor c and for different values of the number of basis functions m.

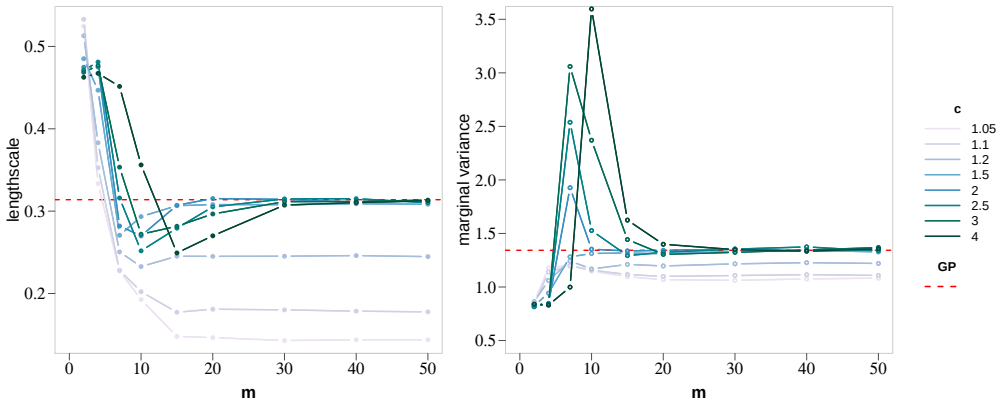

Figure 5. Estimated lengthscale (left) and marginal variance (right) parameters of both regular GP and HSGP models, plotted versus the number of basis functions m and for different values of the boundary factor c.

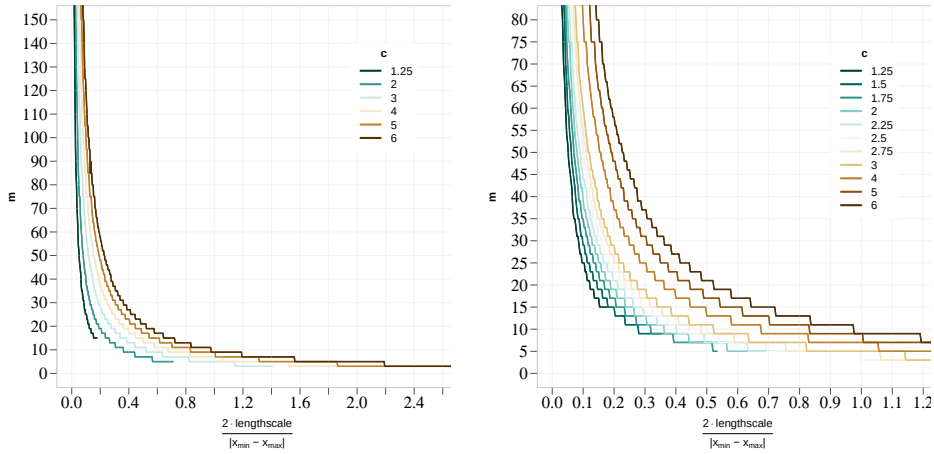

Figure 6. Relation among the minimum number of basis functions m, the boundary factor c ($c = \frac{L}{S}$) and the lengthscale normalized by the half-range of the data ($\frac{\ell}{S}$), in the case of a square exponential covariance function. The right-side plot is a zoom in of the left-side plot.

in Equation (13), is below 1% of the density of the exact covariance function,

$$\frac{\varepsilon}{\int k(\tau)\, d\tau} < 0.01.$$

Alternatively, this figure could be understood as providing the minimum c that we should use for given ℓ and m. Of course, we may also read it as providing the minimum ℓ that can be closely approximated given m and c. We obtain the following main conclusions:

- As ℓ increases, c and m required for a close enough approximation decrease.
- The lower c, the smaller m can and ℓ must be to achieve a close approximation.
- For a given ℓ there exist a minimum c under which a close approximation is never going to be achieved regardless of m. This fact can be appreciated in the Figure as the contour lines which represent c have an end in function of ℓ (Valid c are restricted in function of ℓ).

As stated above, Figures 6 and 7 provide the minimum lengthscale that can be closely approximated given m and c. This information serves as a powerful diagnostic tool in determining if the obtained accuracy is acceptable. As the lenghscale ℓ controls the "wigglyness" of the functional relationship, it strongly influences the difficulty of obtaining accurate inference about the function from the data. Basically, if the lengthscale estimate is accurate, we can expect the HSGP approximation to be accurate as well. Thus, having obtained an estimate $\hat{\ell}$ of ℓ from the HSGP model based on prespecified m and c, we can check whether or not $\hat{\ell}$ exceeds the minimum lengthscale provided in Figure 6. If $\hat{\ell}$ exceeds this recommended minimum lengthscale, the approximation should be close enough. If, however, it does not exceed it, the approximation may be inaccurate and m should be increased or c decreased. We may also use this diagnostic in a iterative procedure. Starting from some initial guess of ℓ, we can choose initial values for m and c and fit an HSGP model, then check the approximation accuracy, and, if not accurate enough because the estimated $\hat{\ell}$ is below the minimum lengthscale provided by Figure 6, repeat the process while increasing m or decreasing c. Note that, as commented before, c can not be decreased as much as desired because it is restricted to the lengthscale.

If we look back to the conclusions drawn from Figures 4 and 5, where $m = 10$ basis

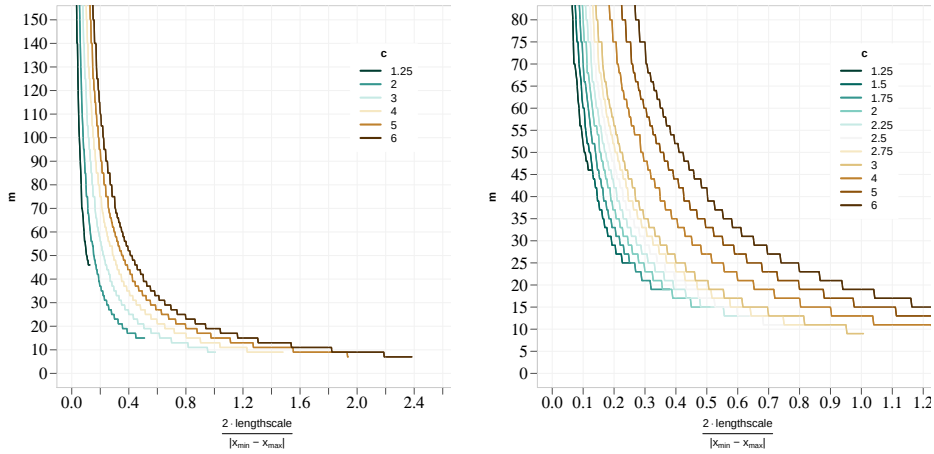

Figure 7. Relation among the minimum number of basis functions m, the boundary factor c ($c = \frac{L}{S}$) and the lengthscale normalized by the half-range of the data ($\frac{\ell}{S}$), in the case of a Matèrn($\frac{3}{2}$) covariance function. The right-side plot is a zoom in of the left-side plot.

functions and a boundary factor of $c = 1.5$ were enough to closely approximate a function with $\ell = 0.3$, we can recognize that these conclusions also matches those obtained from Figure 6.

(Gabi: The following paragraph is just an idea to extend the discussion to the multidimensional case)

Figures 6 and 7 were build for a GP with a unidimensional covariance function, and result in a surface depending on three variables, m, c and ℓ. An equivalent figure for a GP model with a two-dimensional covariance function would result in a surface depending on four variables, m, c, ℓ_1 and ℓ_2, which can not be directly represented. However, as an approximation we can use the unidimensional GP conclusions in Figure 6 to analyze individually the different dimensions of a multidimensional GP model.

•

3.2. Comparing lengthscale estimates

In this example, we make a comparison of the lengthscale estimates obtained from the regular GP and HSGP models. We also have a look at those recommended minimum lengthscales provided by Figure 6.

For this analysis, we will use various datasets consisting of noisy draws from a GP prior model with a squared exponential covariance function and varying lengthscale values. Different values of the number of basis functions m are used when estimating the HSGP models, and the boundary factor c is set to a valid and optimum value in every case.

Figure 8 shows the posterior predictions of both regular GP and HSGP models fitted to those datasets. The lengthscale estimates as obtained by regular GP and HSGP models are depicted in Figure 9. As noted previously, an accurate estimate of the lengthscale can be a good indicator of a close approximation of the HSGP model to the regular GP model. Further, Figure 10 shows the root mean square error (RMSE) of the HSGP models, computed against the regular GP models, as a function of the lengthscale and number of basis functions.

Comparing the accuracy of the lengthscale in Figure 9 to the RMSE in Figure 10, we see that they agree closely with each other for medium lengthscales. That is, a good estimation

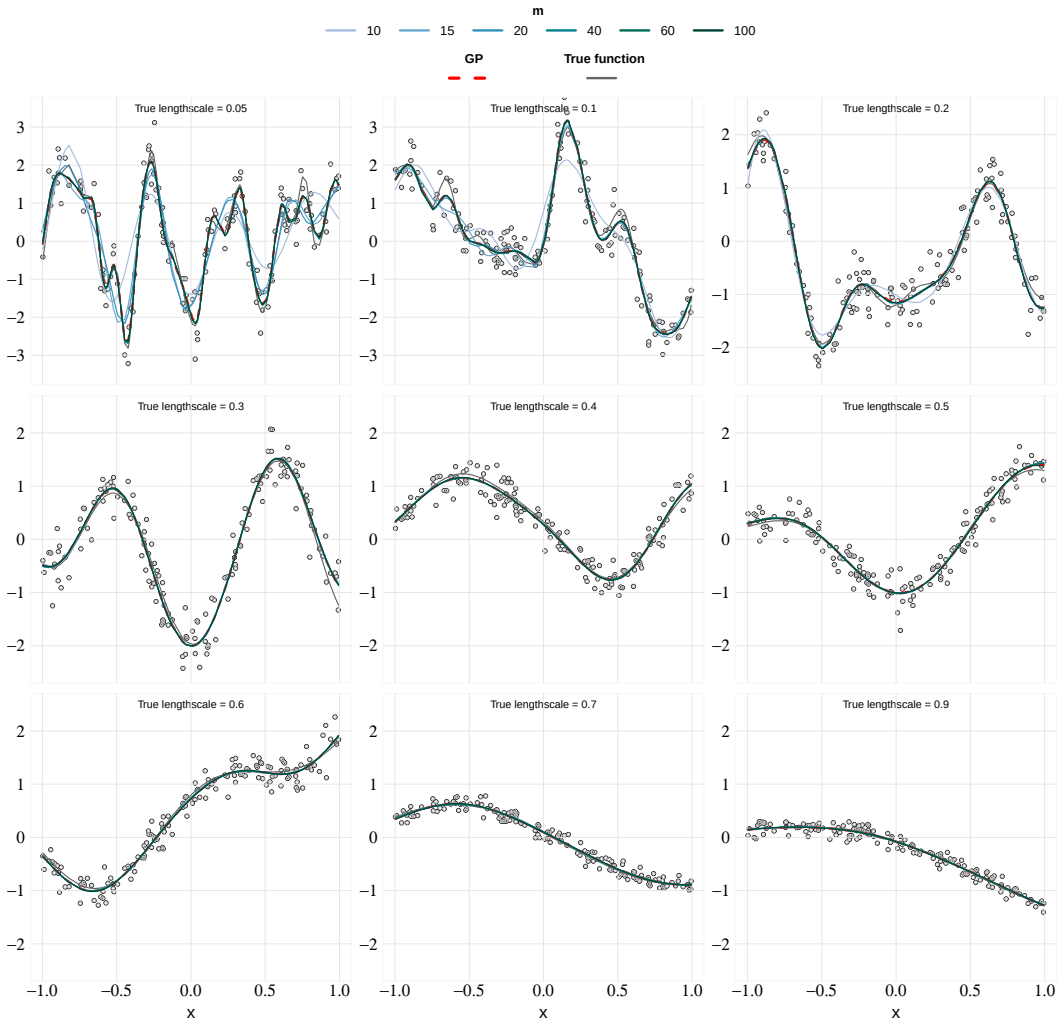

Figure 8. Mean posterior predictions of both regular GP and HSGP models, fitted over various datasets drawn from square exponential GP models with different characteristic lengthscales ($lscale$) and same marginal variance (α) as the data-generating functions (*True function*).

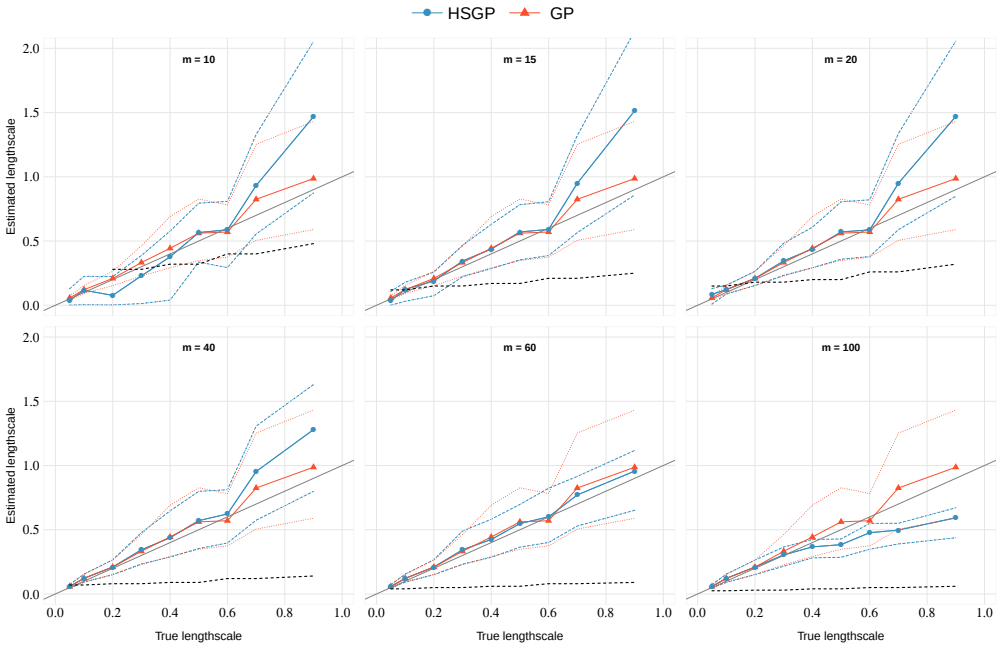

Figure 9. Data-generating functional lengthscales (X-axis), of the various datasets illustrated in Figure 8, versus the corresponding lengthscale estimates from the regular GP and HSGP models (Y-axis). 95% confident intervals of the lengthscale estimates are plotted as dot lines. The different plots represent the use of different number of basis functions m in the HSGP model. The dashed black line represents the recommended minimum lengthscales provided by Figure 6 that can be closely approximated by the HSGP model in every case.

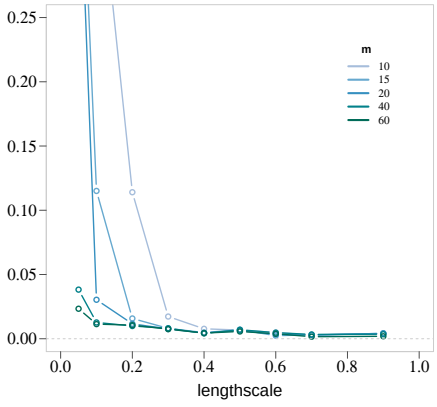

Figure 10. RMSE of the HSGP models with different number of basis functions m, for the various datasets with different wiggly effects (lengthscales).

of the lengthscale implies a small RMSE. This is no longer true for very small or large lengthscales. In small lengthscales, even very small inaccuracies may have a strong influence on the posteriors predictions and thus on the RMSE. In large lengthscales, larger inaccuracies change the posterior predictions only little and may thus not yield large RMSEs. The dashed black line in Figure 9 represents the minimum lengthscale that can be closely approximated under the given condition, according to the results presented in Figure 6. We observe that whenever the estimated lengthscale exceeds the minimially estimable lengthscale, the RMSE of the posterior predictions is small (see Figure 10). Conversely, when the estimated lengthscale is smaller than the minimally estimable one, the RMSE becomes very large.

(Paul: Figure 8 and 9 are not rendered correctly in my PDF viewer after compiling the tex file, which has something to do with the font of the facet and legend labels I believe. Latex also issues warnings about this.)

4. Univariate examples

4.1. Study case I: Simulated data

This example consists of a simulated dataset with 250 ($i = 1, \ldots, n = 250$) single draws from a Gaussian process prior with a Matérn(ν=3/2) covariance function and hyperparameters marginal variance $\alpha = 1$ and lengthscale $\ell = 0.15$, with corresponding inputs values $\boldsymbol{x} = \{x_i\}_{i=1}^{250}$ with $x_i \in [-1, 1] \subset \mathbb{R}$. To form the final noisy dataset $\boldsymbol{y}$, Gaussian noise $\sigma = 0.2$ was added to the GP draws.

The regular GP model for fitting this simulated dataset $\boldsymbol{y}$ can be written as follows,

$$\begin{aligned} \boldsymbol{y} &= f(\boldsymbol{x}) + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \sigma^2 I) \\ f(\boldsymbol{x}) &\sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}', \theta)), \end{aligned}$$

where I represents the identity matrix. The previous formulation corresponds to the latent form of a GP model, where $f(\boldsymbol{x})$ represents the underlying functional model to the noisy data and is modeled by a GP prior with a Matérn($\nu = 3/2$) covariance function k. Saying that the function values $f(\boldsymbol{x})$ follows a GP model is equivalent to say that $f(\boldsymbol{x})$ are multivariate Gaussian distributed with covariance matrix K, where $K_{ij} = k(x_i, x_j, \theta)$.

A more computationally efficient formulation of a GP model with Gaussian likelihood, and for probabilistic inference using sampling methods such as HMC, would be its marginalized form,

$$\boldsymbol{y} \sim \mathcal{N}(0, K + \sigma^2 I)$$

where the function values $f(\boldsymbol{x})$ have been integrated out, yielding a lower-dimensional parameter space over which to do inference, reducing time of computation and improving the sampling and the effective number of samples.

In the HSGP model, the latent function values $f(\boldsymbol{x})$ are approximated as (7),

$$f(\boldsymbol{x}) \approx \sum_{j}^{m} \left(S(\sqrt{\lambda_j}) \right)^{1/2} \phi_j(\boldsymbol{x}) \beta_j,$$

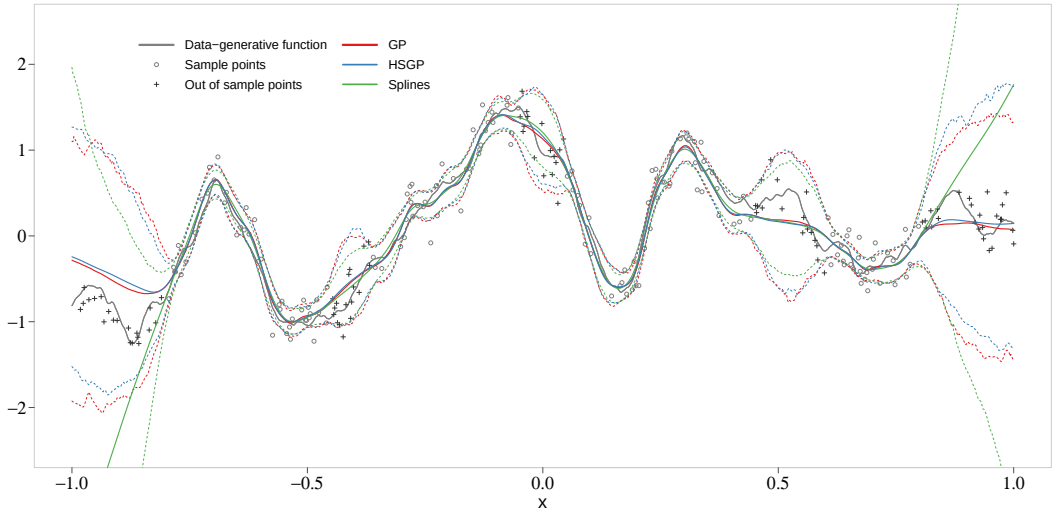

Figure 11. Posterior predictive distributions of the proposed HSGP model, the regular GP model, and the Splines model. 95% credible intervals are plotted as dot lines.

with the spectral density S (2) as a function of $\sqrt{\lambda_j}$,

$$S(\sqrt{\lambda_j}) \quad = \quad \alpha^2 \frac{4\sqrt{3}^3}{\ell^3}(\frac{3}{\ell^2} + \lambda_j)^{-2},$$

and eigenvalues λ_j (5) and eigenfunctions ϕ_j (6)

$$\lambda_j = \left(\frac{j\pi}{2L}\right)^2,$$
$$\phi_j(\boldsymbol{x}) = \sqrt{\frac{1}{L}} \sin\left(\sqrt{\lambda_j}(\boldsymbol{x} + L)\right).$$

In the previous equations, L is the boundary and m the number of basis functions. The parameters β_j are $\mathcal{N}(0, 1)$ distributed, and α and ℓ are the marginal variance and lengthscale parameters, respectively, of the approximate covariance function.

In order to do model comparison, in addition to the regular GP model and HSGP model, an splines-based model is also fitted using the Thin Plate Regression Splines approach in Wood [2003] and implemented in the R-package *mgcv*. A Bayesian approach is used to fit this spline model using the R-package *brms*.

Figure 11 shows the posteriors predictive distributions of the three models, the regular GP, the HSGP with $m = 80$ basis functions and boundary factor $c = 1.2$ ($L = c \cdot 1 = 1.2$ (14)), and the splines model with 80 knots. The true data-generative function and the noisy observations are also plotted in the Figure; the sample observations are plotted as circles and the out-of-sample or test data, which have not been taking part on training the models, are plotted as crosses. The test data located at the extremes of the plot are used for assessing model extrapolation, and the test data located in the middle are used for assessing model interpolation.

In order to assess performance of the models as a function the number of basis functions and number of knots, different models with different number of basis functions for the HSGP model and different number of knots for the splines model have been fitted. Figure 12 shows

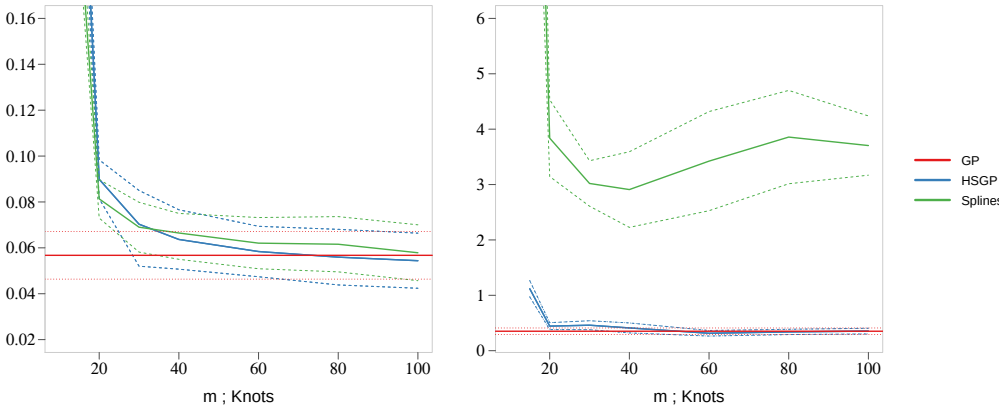

Figure 12. Standarized root mean square error (SRMSE) of the different methods against the data-generating function. (left) SRMSE for interpolation. (right) SRMSE for extrapolation. The Standard deviation of the mean of the SRMSE are plotted as dot lines.

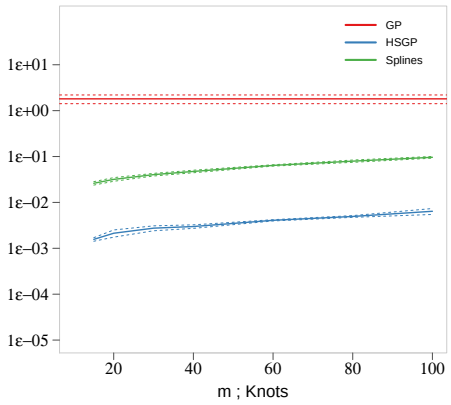

Figure 13. Computational time (Y-axis), in seconds per iteration, as a function of the number of basis functions m and knots. The Y-axis is in a logarithmic scale.

the standardized root mean squared error (SRMSE), for interpolation and extrapolating data, as a function of the number of basis functions and knots. The SRMSE is computed against the data-generating function. From Figures 11 and 12, it can be seen a close approximation of the HSGP model to the regular GP model, for interpolating and extrapolating data. However, the splines model does not extrapolate data properly. Both models show roughly similar interpolating performance.

Figure 13 shows computational times, in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m, for the HSGP model, and knots, for the splines model. The computational time is represented in the Y-axis of the figure, which is in a logarithmic scale. The HSGP model is on average roughly 400 times faster than the regular GP, in the particular case of applying over this dataset.

4.2. Study case II: Gay data

This data set relates the proportion of support for same-sex marriage to the age. The data consists of 74 observations of the amount of people y_i supporting same-sex marriage from a population n_i per age group i ($i = 1, \cdots, 74$). The observational model is a binomial model with parameters population n_i and probability of supporting same-sex marriage p_i per age group i,

$$y_i \sim \text{Binomial}(p_i, n_i).$$

The population per age group n_i is a known quantity and the goal is to estimate the same-sex support probability p_i or mean number of support people per age group. The probabilities $\boldsymbol{p} = \{p_i\}_{i=1}^{74}$ are modeled by a GP function $f(\boldsymbol{x})$ with a squared exponential covariance function k and age input values $\boldsymbol{x} = \{x_i\}_{i=1}^{74}$, with the *logit* function as a link function,

$$\begin{aligned} p_i &= \text{logit}(f(x_i)) \\ f(\boldsymbol{x}) &\sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}', \theta). \end{aligned}$$

In the HSGP model the function $f(\boldsymbol{x})$ is approximated as in Equation (7), with the square exponential spectral density as in Equation (1), and eigenvalues λ_j and eigenfunctions ϕ_j as in Equations (5) and (6).

In order to do model comparison, in addition to the regular GP model and HSGP model, an splines-based model is also fitted using the Thin Plate Regression Splines approach in Wood [2003] and implemented in the R-package *mgcv*. A Bayesian approach is used to fit this splines model using the R-package *brms*.

Figure 14 shows the posteriors predictive distributions of the three models, the regular GP, the HSGP model with $m = 20$ basis functions and boundary factor $c = 1.5$, and the splines model with 20 knots. Sample observations are plotted as circles in the figure, and the out-of-sample observations, which have been used for testing, are plotted as crosses.

For the HSGP model, different models with different number of basis functions and boundary factor have been fitted. The root mean square errors (RMSE) for every one of these models have been computed against the regular GP model and plotted as a function of the number of basis functions m and the boundary factor c in Figure 15, for sample (left) and test (right) data. The expected patterns of the approximation as a function of the number of basis functions and boundary factor are recognized: as the boundary factor increases, more basis functions are needed.

Figure 16 shows the RMSE of the regular GP, HSGP and splines models, computed against the actual data, for training and test data, as a function of the number of basis functions m and boundary factor c for the HSGP model, and knots for the splines model. We can see how the splines models does not extrapolate data properly.

Figure 17 shows computational times, in seconds per iteration, as a function of the number of basis functions m for the HSGP model and knots for the splines model. The computational times is represented in the Y-axis which is in a logarithmic scale. The HSGP model is on average roughly 10 times faster than the regular GP, in this particular case.

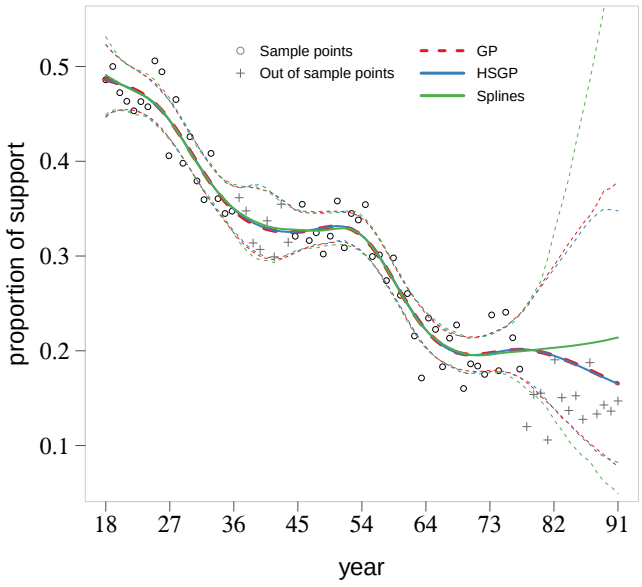

Figure 14. Posterior predictive distributions of the proposed HSGP model, the regular GP model, and the Splines model.

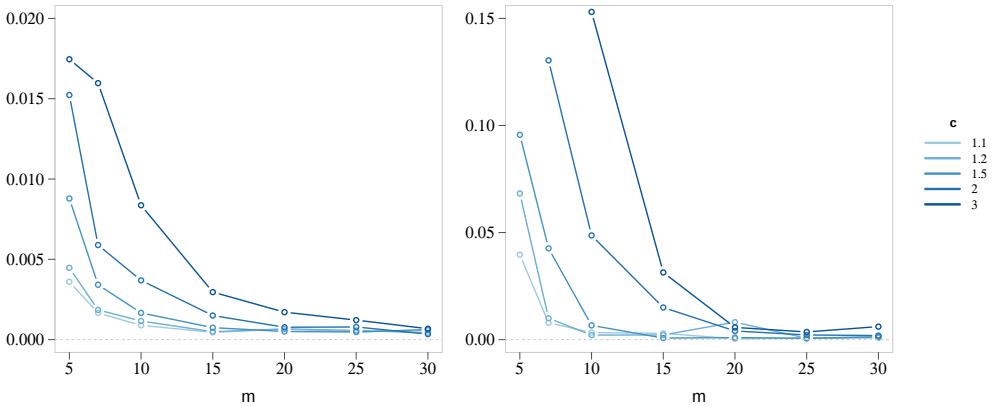

Figure 15. Root mean square error (RMSE) of the HSGP model, computed against the regular GP model, as a function of the number of basis functions m and boundary factor c. (left) RMSE for sample data. (right) RMSE for test data.

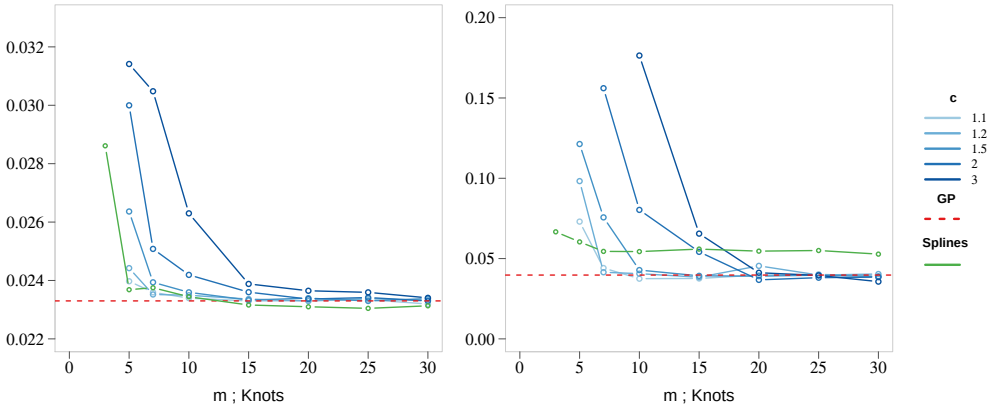

Figure 16. Root mean square error (RMSE) of the different methods, regular GP, HSGP and splines models, computed against the actual data, as a function of the number of basis functions m and boundary factor c for the HSGP model, and knots for the splines model. (left) RMSE for sample data. (right) RMSE for test data.

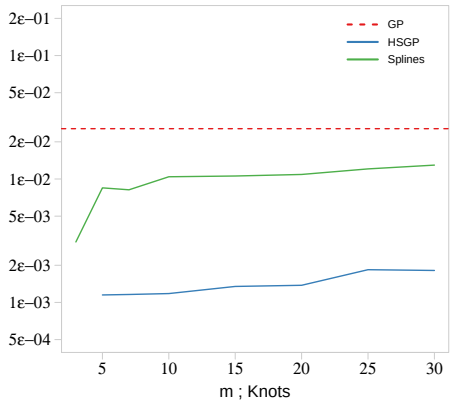

Figure 17. Computational time (Y-axis), in seconds per iteration, as a function of the number of basis functions m and knots. The Y-axis is in a logarithmic scale.

4.3. Study case III: Birthday data

5. Multivariate examples

5.1. Study case IV: Simulated data

This example consists of a simulated dataset with 120 ($i = 1, \ldots, n = 120$) single draws from a Gaussian process prior with two ($D = 2$) input dimensions. A square exponential covariance function, with hyperparameters marginal variance $\alpha = 1$ and lengthscales $\ell_1 = 0.10$ for dimension 1 and $\ell_2 = 0.35$ for dimension 2, is used. The corresponding matrix of input values is $X = \{\boldsymbol{x}_i\}_{i=1}^{n=120} \in \mathbb{R}^{n\times 2}$ with $\boldsymbol{x}_i \in \{[-1, 1], [-1, 1]\} \subset \mathbb{R}^2$. Gaussian noise $\sigma = 0.2$ was added to the GP draws to form the final noisy set of observations $\boldsymbol{y} \in \mathbb{R}^n$.

The regular GP model over the outcome variable $\boldsymbol{y}$ and $n \times 2$ matrix of inputs X, can be written as follows,

$$\begin{aligned}\boldsymbol{y} &= f(X) + \boldsymbol{\epsilon}\\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \sigma^2 I)\\ f(X) &\sim \mathcal{GP}(0, k(X, X', \theta)),\end{aligned}$$

where I represents the identity matrix and k the square exponential covariance function with hyperarameters $\theta = \{\alpha, \ell_1, \ell_2\}$. The marginalized form, by integrating out the latent values $f(X)$, of the previous latent GP model results

$$\boldsymbol{y} \sim \mathcal{N}(0, K + \sigma^2 I),$$

with covariance matrix K, and its element $K_{rs} = k(\boldsymbol{x}_r, \boldsymbol{x}_s, \theta)$ is the evaluation of the covariance function k at points r and s.

In the HSGP model with D input dimensions, the latent function f, evaluated at input vector $\boldsymbol{x} \in \mathbb{R}^D$, is approximated as (12),

$$f(\boldsymbol{x}) \approx \sum_{j}^{m} \left(S(\sqrt{\boldsymbol{\lambda}_j})\right)^{1/2} \phi_j(\boldsymbol{x})\beta_j,$$

where $\boldsymbol{\lambda}_j$ is the D-vector (10) which elements are the univariate eigenvalues (5) whose indices correspond to the elements of the D-tupple $\mathbb{S}_j$.,

$$\boldsymbol{\lambda}_j = \left\{\lambda_{\mathbb{S}_{jd}}\right\}_{d=1}^{D} = \left\{\left(\frac{\pi \mathbb{S}_{jd}}{2L_d}\right)^2\right\}_{d=1}^{D},$$

and ϕ_j is the multivariate eigenfunction (9) as the product of univariate eigenfunctions (6) whose indices correspond to the elements of the D-tupple $\mathbb{S}_j$.,

$$\phi_j(\boldsymbol{x}) = \prod_{d=1}^{D} \phi_{\mathbb{S}_{jd}} = \prod_{d=1}^{D} \sqrt{\frac{1}{L_d}} \sin\left(\sqrt{\lambda_{\mathbb{S}_{jd}}}(x_d + L_d)\right),$$

where x_d is the input value corresponding to dimension d. S is the spectral density (1), as a function of $\sqrt{\boldsymbol{\lambda}_j} = \left\{\sqrt{\lambda_{\mathbb{S}_{jd}}}\right\}_{d=1}^{D}$, of the D-dimensional square exponential covariance

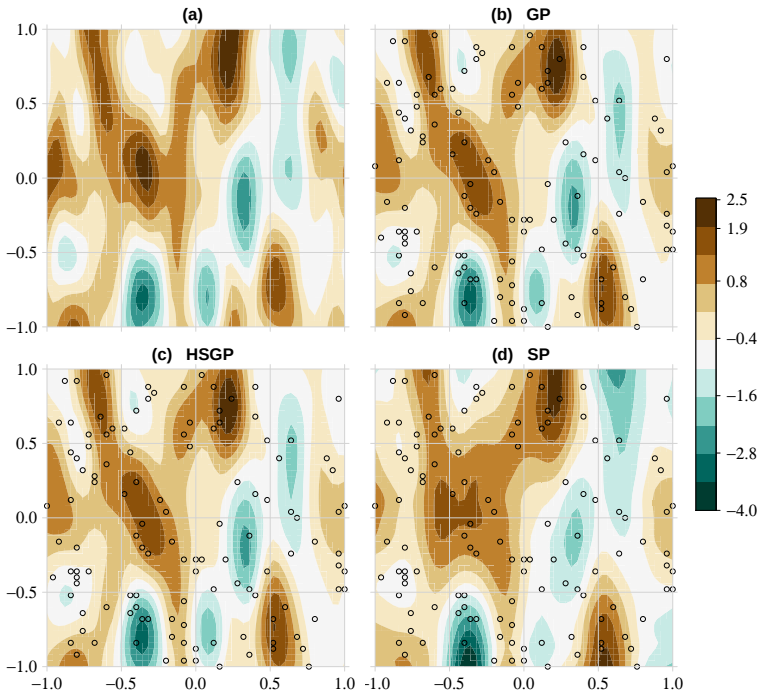

Figure 18. (a) Data-generating function. (b) Mean posterior predictive function of the GP model. (c) Mean posterior predictive function of the HSGP model. (d) Mean posterior predictive function of the splines (SP) model.

function,

$$S(\sqrt{\boldsymbol{\lambda}_j}) = \sigma^2 2\pi \prod_{d=1}^{D} \ell_d \exp\left(-\frac{1}{2} \sum_{d=1}^{D} \ell_d^2 \sqrt{\lambda_{\mathbb{S}_{jd}}}^2 \right).$$

In the previous equations, j denotes the index for the $m = \prod_{d=1}^{D} m_d$ multivariate basis functions (8), where m_d is the number of basis functions considered for dimension d. $\mathbb{S}$ is the matrix of D-tuples, with rows being the indices of every possible combinations of univariate eigenvalues (5) over the D dimensions. L_d is the boundary for the dimension d. The parameters β_j are $\mathcal{N}(0, 1)$ distributed, and α and ℓ_d are the marginal variance and lengthscale of dimension d, respectively, of the approximate multivariate covariance function.

In order to do model comparison, in addition to the regular GP and HSGP models, a two-dimensional splines-based model is also fitted using a cubic spline basis, penalized by the conventional intergrated square second derivative cubic spline penalty [Wood, 2017], and implemented in the R-package *mgcv*. A Bayesian approach is used to fit this spline model using the R-package *brms*.

Figure 18 shows the data-generating GP function from where the dataset was drawn, and the mean posterior predictive functions of the three models, the regular GP, the HSGP, and the splines, fitted over the dataset. Sample observations are also plotted in the plots as circles. For the HSGP model, $m_1 = 40$ and $m_2 = 40$ basis functions for each dimension respectively, were used, which lead to a total of 1600 multivariate basis functions. A boundary factor for each dimension $c_1 = 1.5$ and $c_2 = 1.5$ were used. For the splines model, 40 knots in each dimension were used.

In order to assess performance of the models as a function of the number of basis functions

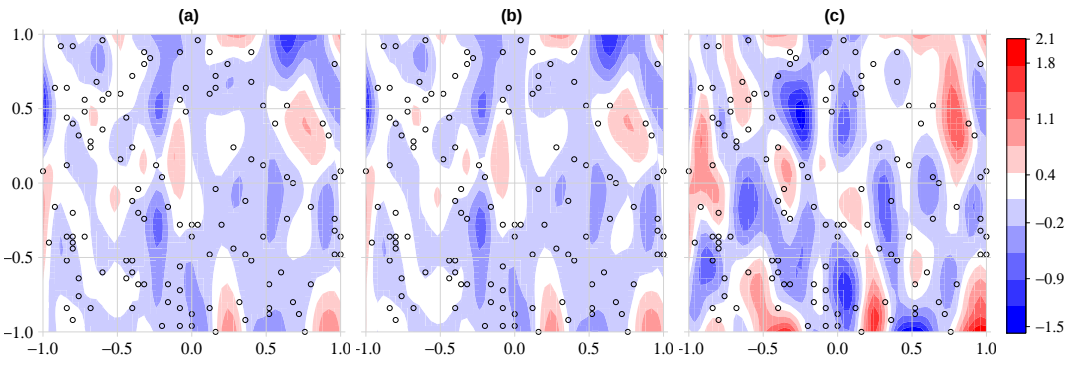

Figure 19. Mean error between the data-generating function and the GP (a), HSGP (b) and splines (c) models.

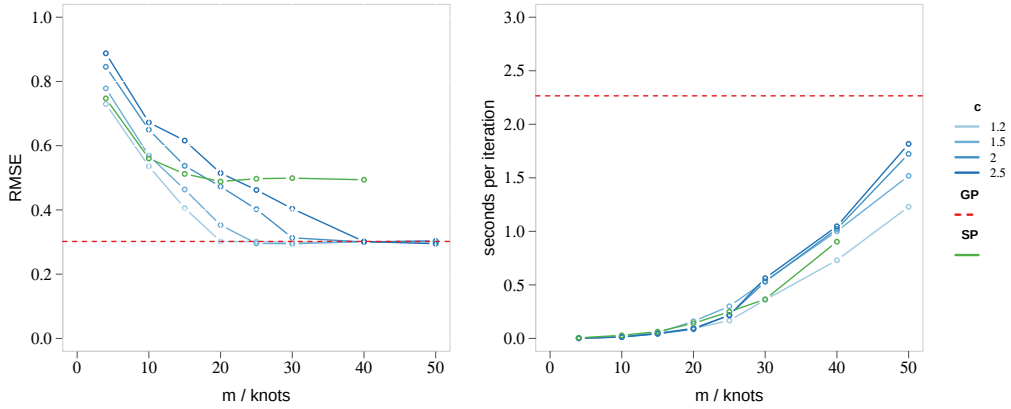

Figure 20. Root mean square error (RMSE) (left) and computational time in seconds per iteration (right) of the different methods computed against the data-generating function, as a function of the boundary factor c, number of basis functions m and knots.

and knots, different models with different number of basis functions for the HSGP model and different number of knots for the splines model have been fitted. In all models, the same number of basis functions and knots per dimension were used. Figure 20(left) shows the root mean squared error (RMSE), computed against the data-generating function, as a function of the boundary factor c, the number of univariate basis functions m and knots. From Figures 19 and 20(left), it can be seen a close approximation of the HSGP model to the regular GP model. However, the performance of the splines model is significantly worse. Figure 20(right) shows the computational times of the different models as a function of the boundary factor, number of basis functions and knots. Figure 20 reveals that choosing the optimal boundary factor allows for less number of basis functions and less computational time.

5.2. Study case V: Diabetes data

The next example presents an epidemiological study of the diabetes diseases. The study aims to relate the probability of suffering from diabetes to some risk factors. The data contains 392 individuals ($i = 1, \cdots, n = 392$) from which the binary variable of suffering ($y_i = 1$) or not suffering ($y_i = 0$) from diabetes have been observed. The matrix $X = \{\boldsymbol{x}_i\}_{i=1}^{n=392} \in \mathbb{R}^{n\times 4}$, with $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, x_{i3}, x_{i4}\} \in \mathbb{R}^4$, contains the risk factors, *Glucose* (x_{i1}), *Pregnancy* (x_{i2}),

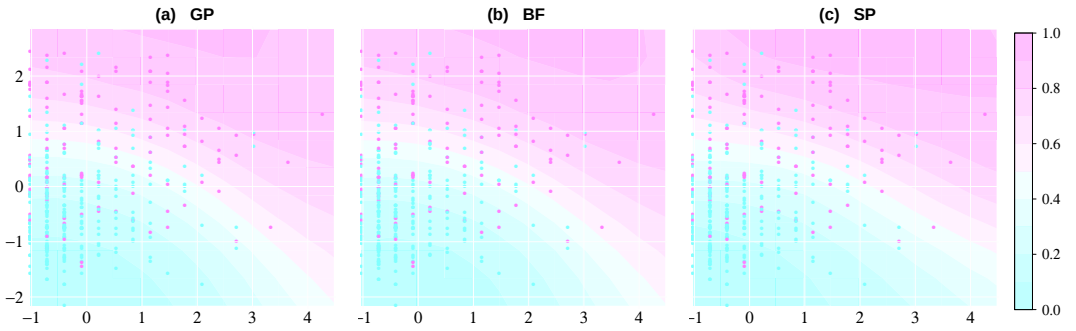

Figure 21. (a) Mean posterior predictive functions of the GP model. (b) Mean posterior predictive functions of the HSGP model. (c) Mean posterior predictive functions of the splines (SP) model.

Age (x_{i3}) and *BMI* (x_{i4}), per individual i. The observational model is a bernoulli model with parameter the probability p_i of suffering from diabetes per observation i,

$$y_i \sim \text{Bernoulli}(p_i).$$

The goal is to estimate the probability p_i as a function of the risk factors, which function $f(X)$ is modeled as a Gaussian process with a squared exponential covariance function k depending on the matrix X of risk factors and hyperparameters $\theta = \{\alpha, \ell\}$, and related to the probabilities p_i through the *logit* link function,

$$\begin{aligned} p_i &= \text{logit}(f(\boldsymbol{x}_i)) \\ f(X) &\sim \mathcal{GP}(0, k(X, X, \theta). \end{aligned}$$

The hyperparameters α and ℓ represent the marginal variance and lengthscale, respectively, of the GP process. Notice that a scalar lengthscale is considered in the multivariate covariance function.

In the HSGP model with D input dimensions, the function $f(\boldsymbol{x})$, evaluated at input vector $\boldsymbol{x} \in \mathbb{R}^D$, is approximated as in equation (12), with the D-dimensional (with a scalar lengthscale) square exponential spectral density S as in equation (1), and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$, and the multivariate eigenfunctions ϕ_j as in equations (10) and (9), respectively.

In order to do model comparison, in addition to the regular GP and HSGP models, a D-dimensional splines-based model is also fitted using a cubic spline basis, penalized by the conventional intergrated square second derivative cubic spline penalty [Wood, 2017], and implemented in the R-package *mgcv*. A Bayesian approach is used to fit this spline model using the R-package *brms*.

Figure 21 shows the mean posterior predictions of probabilities (p_i) of the three models, the regular GP, the HSGP, and the splines, fitted over the dataset with the 2 input dimensions ($D = 2$) *Glucose* and *Pregnancy*. The binary observations y_i are also plotted in the plots as colored points. For the HSGP model, $m_1 = 20$ and $m_2 = 20$ basis functions for each dimension, respectively, were used, which lead to a total of 400 multivariate basis functions. A boundary factor for each dimension $c_1 = 4$ and $c_2 = 4$ were used. For the splines model, 20 knots per dimension were used.

In order to assess performance of the models as a function of the boundary factor, the number of basis functions and knots, different models with different number of basis functions and boundary factor for the HSGP model and different number of knots for the splines model

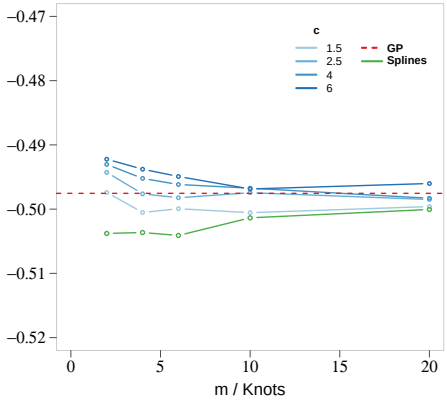

Figure 22. Expected log predictive density (ELPD) of the different methods as a function of the boundary factor c and the number of basis functions m and knots.

have been fitted. In all models, the same boundary factor, number of basis functions and knots per dimension were used. Figure 22 shows the expected log predictive density (ELPD) as a function of the boundary factor c, the number of univariate basis functions m and knots. The ELPD is computed over the actual observations. Basically, with slightly differences, all models show similar performances, due to the fact that the process is very smooth with a relatively very large lengthscale estimate $\ell = 4.51$. Even tough, a slightly pattern of performance improvement can be appreciated as the boundary factor c increases, which fact is because small boundary factors are not allowed when large lengthscales (6).

Figure 24 shows the computational times in the logarithm scale of the different models, regular GP, HSGP and splines, fitted over the dataset with 2 input dimensions (left), 3 input dimensions (center) and 4 input dimensions (right), as a function of the boundary factor c, number of univariate basis functions m and knots. We can appreciate the significant increase of computational time with higher dimensions for the HSGP and splines models. This fact reveals that choosing optimal values for the number of basis functions and boundary factor, looking at the recommendations and diagnosis provided by Figure 6, is essential to avoid excessive computational time, specially in high input dimensionality. It is interesting to be noticed that considering more than 10 knots per dimension in the splines model with 3D is not allowed for an amount of 392 observations. Similarly, just the computation of the input data for the splines model with 4D input dimension is computationally very expensive.

5.3. Study case VI: Leukemia data

The next example presents a survival analysis in acute myeloid leukemia (AML) in adults, with data recorded between 1982 and 1998 in the North West Leukemia Register in the United Kingdom. The data set consist in survival times t_i and censoring indicator z_i (0 for observed and 1 for censored) for 1043 cases ($i = 1, \cdots, n = 1043$). Some 16% of cases were censored. Predictors are *age* ($\boldsymbol{x}_1$), *sex* ($\boldsymbol{x}_2$), *white blood cell* (WBC) ($\boldsymbol{x}_3$) count at diagnosis with 1 unit = $50 \times 109/L$, and the *Townsend deprivation index* (TDI) ($\boldsymbol{x}_4$) which is a measure of deprivation for district of residence. We denote the matrix $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\} \in \mathbb{R}^{n\times 4}$ which contains the predictors.

As the WBC measurements were strictly positive and highly skewed, we fit the model to its logarithm. Continuous predictors were normalized to have zero mean and unit standard

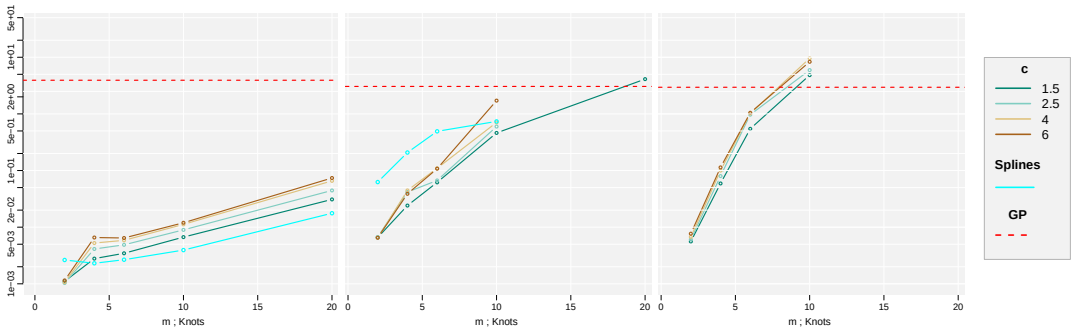

Figure 23. Time of computation in the log scale of the different models fitted over the dataset with 2 input dimensions (left) and 3 input dimensions (center) and 4 input dimensions (right), as a function of the boundary factor c, number of basis functions m and knots.

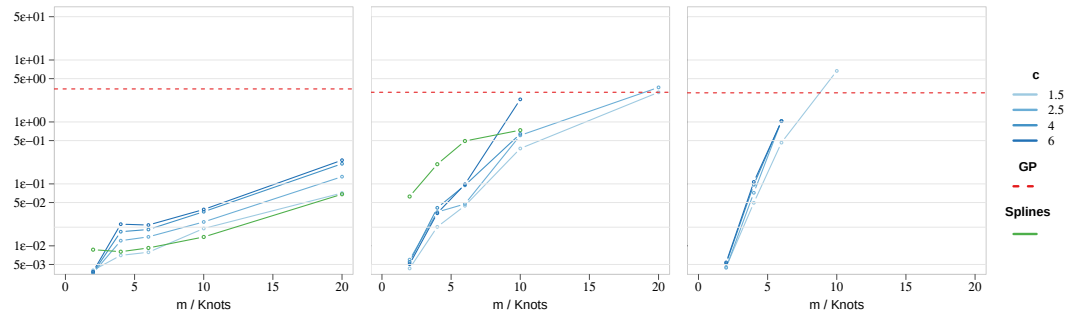

Figure 24. Time of computation in the log scale of the different models fitted over the dataset with 2 input dimensions (left) and 3 input dimensions (center) and 4 input dimensions (right), as a function of the boundary factor c, number of basis functions m and knots.

deviation. Survival time was normalized to have zero mean for the logarithm of time. We assume a Gaussian observation model for the logarithm of the observed survival time, $t_i' = log(t_i)$, with a function of the predictors, $f(X)$, as the location parameter, and σ as the Gaussian noise:

$$p(t_i') = \mathcal{N}(t_i'|f(X_i), \sigma^2)$$

As we do not have a model for the censoring process, we do not have a full observation model, and the observational model for the censored data t_i is assumed to be the complementary cumulative normal probability distribution:

$$p(y_i > t_i') = \int_{t_i'}^{\infty} \mathcal{N}(y_i|f(X_i), \sigma^2)dy_i = 1 - \Phi\left(\frac{y_i - f(X_i)}{\sigma}\right),$$

where y_i denotes the logarithm of the uncensored time.

The latent function $f(X)$ is modeled as a Gaussian process, centered in a linear model of the predictors X, and with a squared exponential covariance function k depending on predictors and hyperparameters $\theta = \{\alpha, \ell\}$,

$$f(X) \sim \mathcal{GP}(c + \boldsymbol{\beta}X, k(X, X, \theta)).$$

where c and $\boldsymbol{\beta}$ are the intercept and vector of coefficients, respectively, of the linear model. The hyperparameters α and ℓ represent the marginal variance and lengthscale, respectively, of the GP process. Notice that a scalar lengthscale is considered in the multivariate covariance function.

Due to the predictor *sex* ($\boldsymbol{x}_2$) is a categorical variable (1 for female and 2 for male), we can outline a multilevel model for the GP function, in a similar way like categorical effects are treated in linear models. The relative contribution of a GP function given one of the levels of the predictor (16) to a general mean GP function (15) is defined. For the other level of the predictor, the GP function effects are set to zero. This multilevel construction is depicted as following:

$$f(X) \sim \mathcal{GP}(c + \boldsymbol{\beta}X, k(X, X, \theta_1)) \tag{15}$$
$$g(X|\boldsymbol{x}_2 = 2) \sim \mathcal{GP}(0, k(X, X, \theta_2|\boldsymbol{x}_2 = 2)) \tag{16}$$
$$f(X|\boldsymbol{x}_2 = 2) = f(X|\boldsymbol{x}_2 = 2) + g(X|\boldsymbol{x}_2 = 2)$$

In the previous equations, θ_1 contains the hyperparameters α_1 and ℓ_1 which are the marginal variance and lengthscale, respectively, of the general mean GP function, and θ_2 contains the hyperparameters α_2 and ℓ_2 which are the marginal variance and lengthscale, respectively, of the GP function restricted to the male sex ($\boldsymbol{x}_2 = 2$).

Using the HSGP approximation, the functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x}|x_2 = 2)$ are approximated as in equation (12), with the D-dimensional (with a scalar lengthscale) square exponential spectral density S as in equation (1), and the multivariate eigenfunctions ϕ_j and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$ as in equations (10) and (9), respectively.

Figure 25 shows estimated conditional comparison of each predictor with all others fixed to their mean values. These posterior estimates correspond to the HSPG model with $m = 10$ basis functions and $c = 3$ boundary factor. The model has found smooth non-linear patterns and the right bottom subplot also shows that the conditional comparison associated with WBC has an interaction with TDI.

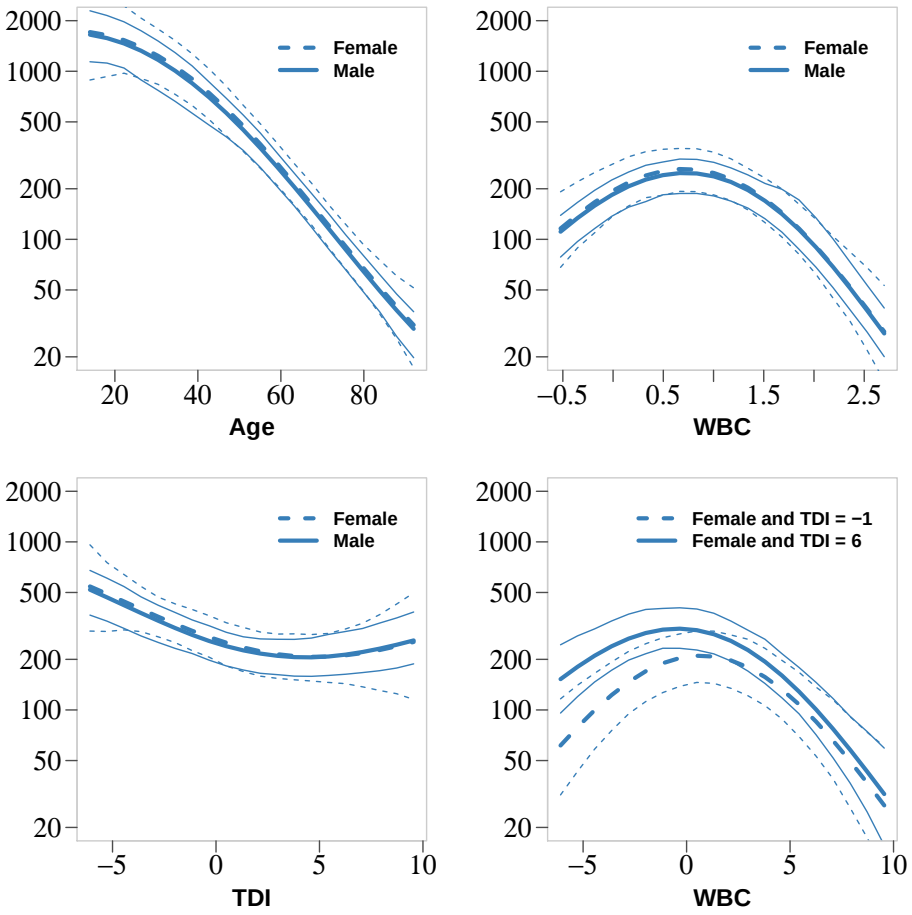

Figure 25. Expected lifetime conditional comparison for each predictor with other predictors fixed to their mean values. The thick line in each graph is the posterior mean estimated using a HSGP model, and the thin lines represent pointwise 95% intervals.

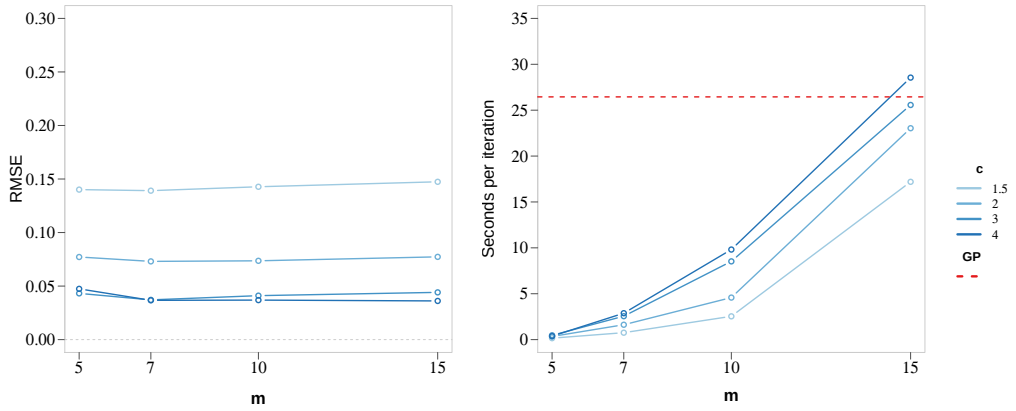

Figure 26. Root mean square error (RMSE) as a function of the number of basis functions m and boundary factor c.

Figure 26 shows the root mean square error (RMSE) and the time of computation in seconds per iteration as a function of the number of univariate basis functions m and boundary factor c. As the functions are smooth, few number of basis functions and a large boundary factor are required to obtain a good approximation (Figure 26-right); Small boundary factors are not allowed when large lengthscales as can be seen in 6. Increasing the boundary factor also significantly increases the time of computation (Figure 26-rleft).

5.4. Study case VII: Land use spatio-temporal classification task

The next example presents an spatio-temporal classification task in land-use of parcels between 2006 and 2015 in a part of the territory of Valencia in Spain dedicated to growing citrus fruits. A set of 200 parcels with known class are used for training the model and a set of 800 parcels in which to predict their unknown classes. The data is recorded in a time series of 5 years (2006,2008,2010,2012,2015) within the period. The class of each parcel i and time t is stored by a categorical variable y_{it} representing the 5 different possible classes ($k = 1, \cdots, K = 5$): $k = 1$, adult independent citrus fruits; $k = 2$, aligned citrus fruits; $k = 3$, irregular citrus fruits; $k = 4$, abandoned citrus fruits; $k = 5$, young citrus fruits.

A bunch of 52 characteristic variables were measured as predictors for every parcel and time. These variables were computed from satellite color images and cadastral map, and they concern spectral intensities and empirical semivariogram of the pixels within a parcel, as well as descriptive statistics of the shape of the parcels.

Due to the fact that 52 input variables are too many for a multivariate HSGP model as it scales $O(n \cdot m^D)$ with m the number of basis functions in the HSGP model and D the number of input variables, the multivariate HSGP model will be formulated as an additive HSGP model.

In order to avoid high-order components in the additive HSGP model as much as possible, since it is known multivariate HSGP models to be computationally demanding, a Principal Component Analysis is carried out over the characteristic variables. The resulting principal components are linearly uncorrelated each other and will be used jointly with the time variable as predictors in the classifying first-order additive HSGP model. Let's denote the matrix $X = \{\boldsymbol{x}_{it}\}$ which contains the predictors, where $\boldsymbol{x}_{it} \in \mathbb{R}^D$, $D = 53$ (52 principal components + time) is the D-vector of characteristic variables of the parcel i and time t. The use of a first-

Estimate \ True	caind (k=1)	calign (k=2)	irr (k=3)	aband (k=4)	young (k=5)	
caind (k=1)	90	39	14	3	11	42%
calign (k=2)	46	301	8	2	3	16%
irr (k=3)	8	4	59	4	1	19%
aband (k=4)	5	2	6	342	5	5%
young (k=5)	8	2	1	0	38	19%
	42%	13%	32%	2%	34%	**17%**

Table 1. Confusion matrix after the Q-fold cross-validation procedure over the training data.

order additive model is justified due to the input variables correspond to principal components which are linearly uncorrelated each other.

The observational model is a multinomial model with parameters the vector of probabilities $\boldsymbol{p}_i = \{p_{ik}\}_{k=1}^{K=5}$ of belonging to class k per parcel i,

$$y_i \sim \text{multinomial}(\boldsymbol{p}_i).$$

The goal is to estimate the vector of probabilities $\boldsymbol{p}_i$ as a function of the input values, which is a multivariate function $f(\boldsymbol{x}_i) : \boldsymbol{x_i} \in \mathbb{R}^D \rightarrow \{f_k(\boldsymbol{x}_i)\}_{k=1}^{K} \in \mathbb{R}^K$, related to the vector of probabilities $\boldsymbol{p}_i$ through the 'softmax' link function.

$$\boldsymbol{p}_i = \text{softmax}(f(\boldsymbol{x}_i)).$$

Each individual function $f_k(\boldsymbol{x})$ is modeled as a first-order additive model,

$$f_k(\boldsymbol{x}) \sim \sum_{d=1}^{D} g_d(x^d)$$

where each individual component $g_d(x^d)$ is modeled as a unidimensional HSGP model,

$$g_d(x^d) \sim \mathcal{HSGP}(x^d, S, \theta_d),$$

In the HSGP model, the function $g_d(x^d)$, evaluated at input value $x^d \in \mathbb{R}$, is approximated as in Equation (7), with the square exponential spectral density S as in Equation (1), and eigenvalues λ_j and eigenfunctions ϕ_j as in Equations (5) and (6). The vector of hyperparameters $\theta_d = \{\alpha_d, \ell_d\}$ contains the marginal variance α_d and lengthscale ℓ_d of the HSGP process.

For each individual component $g_d(x^d)$ modeled as a HSGP model, m=15 basis functions and a boundary factor c=2.5 were used. All the input variables were previously standarized with the exception of the time variable.

After the adjustment of the model, the lengthscales estimates $\{\hat{\ell}_d\}_{d=1}^{D=14}$ normalized by half of the input range ($\frac{2 \cdot \hat{\ell}_d}{|x^d_{max} - x^d_{min}|}$) are all bigger than the minimun lengthscale reported by Figure 6 as function of m, c. Which means that the used number of basis functions ($m = 15$) and boundary factor ($c = 2.5$) are suitable values for modeling the data.

Table 1 shows the confusion classification matrix after fit the model following a Q-fold cross-validation procedure over the training data. Every fold was composed of Q=10 observations.

5.5. Study case VII: Land use spatio-temporal classification task

The next example presents an spatio-temporal classification task in land-use of parcels between 2006 and 2015 in a part of the territory of Valencia in Spain dedicated to growing citrus fruits. A set of 200 parcels with known class are used for training the model and a set of 800 parcels in which to predict their unknown classes. The data is recorded in a time series of 5 years (2006,2008,2010,2012,2015) within the period. The class of each parcel i and time t is stored by a categorical variable y_{it} representing the 5 different possible classes ($k = 1, \cdots, K = 5$): $k = 1$, adult independent citrus fruits; $k = 2$, aligned citrus fruits; $k = 3$, irregular citrus fruits; $k = 4$, abandoned citrus fruits; $k = 5$, young citrus fruits.

A bunch of 52 characteristic variables were available for every parcel and time. These variables were computed from satellite color images and cadastral map, and they concern spectral intensities and empirical semivariogram of the pixels within a parcel, as well as descriptive statistics of the shape of the parcels.

Due to the fact that 52 input variables are too many for a multivariate HSGP model as it scales $O(n \cdot m^D)$ with m the number of basis functions in the HSGP model and D the number of input variables, the multivariate HSGP model will be formulated as an additive HSGP model.

In order to avoid as much as possible high-order components in the additive HSGP model, since it is known multivariate HSGP models to be computationally demanding, a Principal Component Analysis is carried out over the characteristic variables. The resulting principal components are linearly uncorrelated each other and will be used jointly with the time variable as predictors in the classifying additive HSGP model. Let's denote the matrix $X = \{\boldsymbol{x}_{it}\}$ which contains the predictors, where $\boldsymbol{x}_{it} \in \mathbb{R}^D$, $D = 53$ (52 principal components + time) is the D-vector of characteristic variables of the parcel i and time t. The classifying additive HSGP model will be composed of first-order additive components with the 53 predictors and second-order additive components corresponding to the interaction components among the 52 principal components and the time variable.

The observational model is a multinomial model with parameters the vector of probabilities $\boldsymbol{p}_{it} = \{p_{it,k}\}_{k=1}^{K=5}$ of belonging to class k per parcel i and time t,

$$y_{it} \sim \text{multinomial}(\boldsymbol{p}_{it}).$$

The goal is to estimate the vector of probabilities $\boldsymbol{p}_{it}$ as a function of the predictors, which is a multivariate function $f(\boldsymbol{x}_{it}) : \boldsymbol{x}_{it} \in \mathbb{R}^D \rightarrow \{f_k(\boldsymbol{x}_{it})\}_{k=1}^{K} \in \mathbb{R}^K$, which is related to the vector of probabilities $\boldsymbol{p}_{it}$ through the 'softmax' link function.

$$\boldsymbol{p}_{it} = \text{softmax}(f(\boldsymbol{x}_{it})).$$

Each individual function $f_k(\boldsymbol{x})$ is modeled as a second-order additive model in the form:

$$f_k(\boldsymbol{x}) \sim \sum_{d=1}^{D} g_d(x^d) + \sum_{d=1}^{D-1} g_{d+D}(x^d, x^D), \tag{17}$$

where x^D represents the time variable. The components $g_d(x^d)$, for $d = 1, \ldots, D$, in Equation 17, are modeled as unidimensional HSGP models

$$g_d(x^d) \sim \mathcal{HSGP}(x^d, S, \theta_d).$$

In the HSGP model, the function $g_d(x^d)$, evaluated at input value $x^d \in \mathbb{R}$, is approximated

Estimate \ True	caind (k=1)	calign (k=2)	irr (k=3)	aband (k=4)	young (k=5)	
caind (k=1)	90	39	14	3	11	42%
calign (k=2)	46	301	8	2	3	16%
irr (k=3)	8	4	59	4	1	19%
aband (k=4)	5	2	6	342	5	5%
young (k=5)	8	2	1	0	38	19%
	42%	13%	32%	2%	34%	**17%**

Table 2. Confusion matrix after the Q-fold cross-validation procedure over the training data.

as in Equation (7), with the square exponential spectral density S as in Equation (1), and eigenvalues λ_j and eigenfunctions ϕ_j as in Equations (5) and (6).

The components $g_{d+D}(x^d, x^D)$, for $d = 1, \ldots, D-1$, , in Equation 17, are modeled as two-dimensional HSGP models

$$g_{d+D}(x^d, x^D) \sim \mathcal{HSGP}(x^d, x^D, S, \theta_{d+D}).$$

In the HSGP model, the function $g_{d+D}(x^d, x^D)$, evaluated at input values $x^d \in \mathbb{R}$ and $x^D \in \mathbb{R}$, is approximated as in Equation (12), with the two-dimensional (with a scalar lengthscale) square exponential spectral density S as in Equation (1), and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$ and the multivariate eigenfunctions ϕ_j as in Equations (10) and (9), respectively.

The vectors of hyperparameters $\{\theta_d = [\alpha_d, \ell_d]\}_{d=1}^{2D-1}$ contains the marginal variance α_d and lengthscale ℓ_d of the HSGP model component.

For the unidimensional HSGP components $\{g_d(x^d)\}_{d=1}^{D}$, m=15 basis functions and a boundary factor c=2.5 were used. For the two-dimensional HSGP components $\{g_{d+D}(x^d, x^D)\}_{d=1}^{2D-1}$, m_1=15 and m_2=15 basis functions for each dimension, respectively, were used, which lead to a total of 225 multivariate basis functions. A boundary factor for each dimension c_1=2.5 and c_2=2.5 were used. All the input variables were previously standarized with the exception of the time (x^D) variable.

After the adjustment of the model, the lengthscales estimates $\{\hat{\ell}_d\}_{d=1}^{D=14}$ normalized by half of the input range ($\frac{2 \cdot \hat{\ell}_d}{|x^d_{max} - x^d_{min}|}$) are all bigger than the minimun lengthscale reported by Figure 6 as function of m, c. Which means that the used number of basis functions ($m = 15$) and boundary factor ($c = 2.5$) are suitable values for modeling the data.

Table 1 shows the confusion classification matrix after fit the model following a Q-fold cross-validation procedure over the training data. Every fold was composed of Q=10 observations.

Appendix A. Related work

The GP prior entails an $O(n^3)$ complexity that is computationally intractable for many practical problems, and this problem especially becomes severe when we want to conduct inference using sampling methods. To overcome this scaling problem several schemes have been proposed. One approach is to partition the data set into separate groups [Snelson and Ghahramani, 2007, Urtasun and Darrell, 2008] and performing local inference in each partition. Other global approach is to build a low-rank approximation to the covariance matrix of the complete data based around 'inducing variables' [Bui et al., 2017, Quiñonero-Candela and Rasmussen, 2005]. Other global approach make use of basis functions to approximate the covariance function. In

Snelson and Ghahramani [2007] the authors conduct an approach that combines the idea of local and global approaches.

The literature contains many parametric models that approximate Gaussian process behaviours; for example Bui and Turner [2014] included tree-structures in the approximation for extra scalability, and Moore and Russell [2015] combined local Gaussian processes with Gaussian random fields.

A.1. Inducing points methods

The approach based on inducing points employs a small set of pseudo data points to summarise the actual data. The storage requirements are reduced to $O(nm)$ and complexity to $O(nm^2)$, where $m < n$. Some of these methods have been reviewed in Rasmussen and Williams [2006], and Quiñonero-Candela and Rasmussen [2005] provide a unifying view of these methods based on approximate generative methods. From a spectral point of view, several of these methods (e.g., SOR, DTC, VAR, FIC) can be interpreted as modfications to the so-called Nyström method (see Arthur [1979] and Williams and Seeger [2001]), a scheme for approximating the eigenspectrum. These methods are basically based on choosing a set of m inducing inputs x_u and scaling the corresponding eigendecomposition of their corresponding covariance matrix $K_{u,u}$ to match that of the actual covariance.

This scheme was originally introduced to the GP context by Williams and Seeger [2001]. As discussed by Quiñonero-Candela and Rasmussen [2005], the Nyström method by Williams and Seeger [2001] does not correspond to a well-formed probabilistic model. However, several methods modifying the inducing point approach are widely used. The Subset of Regressors (SoR) [Smola and Bartlett, 2001] method uses the Nyström approximation scheme and a finite linear-in-the-parameters model for approximating the whole (training and test) covariance function, whereas the sparse Nyström method [Williams and Seeger, 2001] only replaces the training data covariance matrix. The SoR method is based on a degenerate prior which produces unreasonable predictive uncertainties, which is a general problem of linear models (for more details see Rasmussen and Williams [2006]).

The Deterministic Training Conditional (DTC) method [Ro and Opper, 2001, Seeger et al., 2003]) retains the true covariance for the training data, but uses the approximate cross-covariances between training and test data, which reverse the problem of nonsensical predictive uncertainties. However, since the covariances for training and test cases are computed differently, this method results not to actually be a Gaussian process. This method was presented as Projected Latent Variables (PLV) in Seeger et al. [2003] and Projected Process Approximation (PPA) in Rasmussen and Williams [2006].

The Variational Approximation (VAR) [Titsias, 2009] suggests a variational approach which provides an objective function for optimizing the selection of inducing points. This basically modifies the DTC method by an additional trace term in the likelihood that comes from the variational bound. Hensman et al. [2013] extended this idea by introducing additional variational parameters to enable stochastic variational inference [Hoffman et al., 2013], achieving a more computationally scalable bound which allows GPs to be fitted to millions of data.

The Fully Independent (Training) Conditional (FIC) [Quiñonero-Candela and Rasmussen, 2005] method originally introduced as Sparse Pseudo-Input GP by Snelson and Ghahramani [2006] is also based on the Nyström approximation, where they allow the pseudo-point input locations to be optimised by maximising the new model's marginal likelihood whose covariance is parameterized by the locations of an active set not constrained to be a subset of the training and test data.

More recently Bui et. al (2017) revisit the inducing points-based sparse approximation

methods, in which all the necessary approximation is performed at inference time, rather than at the modelling time. The new framework is built on standard methods for approximate inference (variational-free-inference, EP and Power EP methods).

In practice, the inducing points-based sparse approximation methods works reasonable well in cases where the field is relatively smooth. Vanhatalo et al. [2010] propose the use of compactly supported covariance function in conjunction with sparse approximations to model both short and long range correlations.

Wilson and Nickisch [2015] introduce a new unifying framework for inducing point methods, called structured kernel interpolation (SKI). This framework improves the scalability and accuracy of fast kernel approximations through kernel interpolation, and naturally combines the advantages of inducing point and structure exploiting for scalability (such as Kronecker [Saatçi, 2012] or Toeplitz [Cunningham et al., 2008]) approaches.

The number of inducing points or their locations are crucial in order to capture the correlation structure. For a discussion on the effects of the inducing points, see Vanhatalo et al. [2010]. This behavior applies to all the methods from the Nystr$\acute{o}$m family.

This kind of 'projected process' approximation has also been discussed by e.g. Banerjee et al. [2008].

A.2. Basis function methods

The spectral analysis and series expansions of Gaussian processes has a long history. A classical result (see, e.g, Adler [1981], Cramér and Leadbetter [2013], Loève [1977], Trees [1968], and references therein) is that the covariance function can be approximated with a finite truncation of Mercer series and the approximation is guaranteed to converge to the exact covariance function when the number of terms is increased.

Another related classical connection is to the works in the relationship of spline interpolation and Gaussian process priors [Kimeldorf and Wahba, 1970, Wahba, 1978, 1990]. In particular, it is well-known (see, e.g., Wahba [1990]) that spline smoothing can be seen as Gaussian process regression with a specific choice of covariance function. The relationship of the spline regularization with Laplace operators then leads to series expansion representations that are closely related to the approximations considered here.

Random Fourier Features [Rahimi and Recht, 2008, 2009] is a method for approximating kernels. The approximate kernel has a finite basis function expansion.

The Sparse Spectrum GP is based on a sparse approximation to the frequency domain representation of a GP [Lázaro Gredilla, 2010, Quiñonero-Candela et al., 2010], where the spectral representation of the covariance function is used. This model is a stationary sparse GP that can approximate any desired stationary full GP. However, as argued by the authors, this option does not converge to the full GP and can suffer from overfitting to the training data. [Gal and Turner, 2015] sought to improve the model by integrating out, rather than optimizing the frequencies. Gal and Turner derived a variational approximation that made use of a tractable integral over the frequency space. The result is an algorithm that suffers less overfitting than the Sparse Spectrum GP, yet remains flexible.

While Sparse Spectrum GP is based on a sparse spectrum, the reduced-rank method proposed in this paper aims to make the spectrum as 'full' as possible at a given rank.

Recently [Hensman et al., 2017] presented a variational Fourier feature approximation for Gaussian processes that was derived for the Matérn class of kernels, where the approximation structure is set up by a low-rank plus diagonal structure. They combine the variational methodology with Fourier based approximations.

In spatial statistics similar approaches are called low-rank models [Diggle et al., 2007]. The

low rank models assume that the Gaussian field is a linear combination of m basis functions. The type of an approximation depends on the basis functions used. Familiar examples include spectral representation [Diggle et al., 2007, Paciorek, 2007, 007b] and splines [Wood, 2003].

Recent Splines models can reproduce the Matern family of covariance functions, however our approach can reproduce basically all of the stationary covariance functions.

Appendix B. Contributions of the method

This work is based on the novel method developed by Solin and Särkkä [2018] for reduced-rank approximations of GP models. This method is based on interpreting the covariance function as the kernel of a pseudo-differential operator and approximating it using Hilbert space methods. This results in a reduced-rank approximation for the covariance function. This method has some nice features:

• It has an attractive computational cost as this basically turns the regular GP model into a lineal model.

• In a fully Bayesian inference framework using sampling methods, the proposed approximate GP model has a computational complexity of $O(nm + m)$ in every step of the HMC method. In addition, the computation of the automatic differentiation to compute the gradients in this linear model scales $O(n)$?, an operation that must be computed in every step of the HMC method.

• Using maximizing marginal likelihood methods, the proposed model has a overall complexity of $O(nm^2)$. After this, evaluating the marginal likelihood and marginal likelihood gradients is an $O(m^3)$ operation in every step of the optimizer. (Arno's paper, pag. 7)

• The parameter posterior distribution in this approximate GP model is m-dimensional ($m << n$) which helps the use of GP priors as latent functions. especially when sampling methods for inference are used. GP prior as latent functions is needed in generalized models.

In regular GPs and other approximate GP models and Splines models these features do not have so nice properties:

• In a regular GPs, the main computational complexity comes from the inversion of the covariance matrix which is in general a $O(n^3)$ operation. This operation has to be computed at every step of the HMC or optimizer.

• In regular GPs, the parameter posterior distributions is N-dimensional. It is known that when N is of medium or large size there is high correlation between the N-dimensional latent function and the hyperparameters of the GP prior.

• In conventional sparse GP approximations, although the rank of the GP is reduced considerably to the number of inducing points, this still needs to do the autodiff and covariance matrix inversion.

• The Splines models are also a sort of basis functions expansion model, then the computational demands are similar to that in this approach. However in Splines models the lengthscale hyperparameter tend to be fixed and then the fit is covered by the magnitude parameter. In that sense, Splines models tend to loose the useful interpretation of the lengthscale parameter.

Appendix C. Contributions of our work

As said above the proposed method was already developed by Solin and Särkkä [2018] where they fully develop, describe and generalize the methodology. Though, they do not put much effort in describing and analyzing the relation among the key factors of the box size (or boundary condition), the number of basis functions, and the smoothness or roughness of the function. The performance and accuracy of the method are directly related with the number of basis functions and the box size. At the same time, successful values for these two factors depend on the smoothness or roughness of the process to be modeled. The time of computation is mainly dependent on the number of basis functions. Our main contributions to this recently developed methodology for low-rank GP model by Solin and Särkkä [2018] goes around these aspects.

• Firstly, clear summarized formulae of the method for the univariate and multivariate cases is presented.

• We investigate the relations going on among these factors, the number of basis functions, the box size, and the lengthscale of the functions.

• We make recommendations for the values of these factors based on the recognized relations among them. We provide useful graphs of these relations that will help the users to improve performance and save time of computation.

• We also diagnose if the chosen values for the number of basis functions and the box size are adequate to fit to the actual data.

• We describe the generalization of the method to the multidimensional case.

• We implement the approach in a fully probabilistic framework and for the Stan programming probabilistic software.

• We show several illustrative examples, simulate and real datasets, of the performance of the model, and accompanied by their Stan codes.

Appendix D. Spectral densities of stationary covariance functions

The covariance function of a stationary process, that is function of $\boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x'}$ can be represented as the Fourier transform of a positive finite measure ($Bochner's theorem$).

(Bochner's theorem) A complex-valued function k on $\mathbb{R}^D$ is the covariance function of a weakly stationary mean square continuous complex valued random process on $\mathbb{R}^D$ if and only if it can be represented as

$$k(\boldsymbol{\tau}) = \int_{\mathbb{R}^D} e^{2\pi i \boldsymbol{s}\cdot\boldsymbol{\tau}} d\mu(\boldsymbol{\tau}),$$

where μ is a positive finite measure.

If the measure μ has a density, it is known as the spectral density $S(\omega)$ of the covariance function, and the covariance function and the spectral density are Fourier duals, known as the Wiener-Khintchine theorem. It gives the following relations:

$$
\begin{aligned}
k(\boldsymbol{\tau}) &= \int S(\boldsymbol{s}) e^{2\pi i \boldsymbol{s}\cdot\boldsymbol{\tau}} d\boldsymbol{s} \\
S(\boldsymbol{s}) &= \int k(\boldsymbol{\tau}) e^{-2\pi i \boldsymbol{s}\cdot\boldsymbol{\tau}} d\boldsymbol{s}
\end{aligned}
$$

Appendix E. Approximate the covariance function using Hilbert space methods

Associated to each covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ we can also define a covariance operator $\mathcal{K}$ as follows:

$$
\mathcal{K} f(\boldsymbol{x}) = \int k(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}') d\boldsymbol{x}'. \tag{E1}
$$

Assuming that the spectral density function $S(\cdot)$ is regular enough, then it can be represented as a polynomial expansion:

$$
S(\boldsymbol{w}) = a_0 + a_1 \boldsymbol{w}^2 + a_2 (\boldsymbol{w}^2)^2 + a_1 (\boldsymbol{w}^2)^3 + \cdots \tag{E2}
$$

If the negative Laplace operator $-\nabla^2$ is defined as the covariance operator of the covariance function k,

$$
-\nabla^2 f(\boldsymbol{x}) = \int k(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}') d\boldsymbol{x}', \tag{E3}
$$

then the covariance function can be represented as

$$
k(\boldsymbol{x}, \boldsymbol{x}') = \sum_j \lambda_j \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}'), \tag{E4}
$$

where $\{\lambda_j\}_{j=1}^{\infty}$ and $\{\phi_j(x)\}_{j=1}^{\infty}$ are the set of eigenvalues and eigenvectors, respectively, of the Laplacian operator. Namely, they satisfy the following eigenvalue problem in the compact subset $x \in \{-L, L\}$ and with the Dirichlet boundary condition (another boundary condition could be used as well):

$$
\begin{aligned}
-\nabla^2 \phi_j(x) &= \lambda \phi_j(x), \qquad && x \in \{-L, L\} \\
\phi_j(x) &= 0, && x \notin \{-L, L\}.
\end{aligned} \tag{E5}
$$

a series expansion of eigenvalues and eigenfunctions

Appendix F. Example of generalization to the multivariate case

Next, as an example we show the matrix $\mathbb{S}$ and eigenfunctions and eigenvalues for a two-dimensional input vector $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ ($D = 2$) and three eigenfunctions and eigenvalues

$(J = 3)$ for every dimension. The number of new multidimensional eigenfunctions ϕ_j^* and eigenvalues λ_j^* is $J^D = 3^2 = 9$ ($j = \{1, \cdots, J^D\}$). The matrix $\mathbb{S} \in \mathbb{R}^{9\times 2}$ is

$$\mathbb{S} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 2 & 1 \\ 2 & 2 \\ 2 & 3 \\ 3 & 1 \\ 3 & 2 \\ 3 & 3 \end{bmatrix}$$

and the multidimensional eigenfunctions and eigenvalues

$$\begin{aligned}
\phi_1^*(\boldsymbol{x}) &= \phi_1(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) & \boldsymbol{\lambda}_1^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2)\} \\
\phi_2^*(\boldsymbol{x}) &= \phi_1(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) & \boldsymbol{\lambda}_2^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2)\} \\
\phi_3^*(\boldsymbol{x}) &= \phi_1(\boldsymbol{x}_1) \cdot \phi_3(\boldsymbol{x}_2) & \boldsymbol{\lambda}_3^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_3(\boldsymbol{x}_2)\} \\
\phi_4^*(\boldsymbol{x}) &= \phi_2(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) & \boldsymbol{\lambda}_4^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2)\} \\
\phi_5^*(\boldsymbol{x}) &= \phi_2(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) & \boldsymbol{\lambda}_5^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2)\} \\
\phi_6^*(\boldsymbol{x}) &= \phi_2(\boldsymbol{x}_1) \cdot \phi_3(\boldsymbol{x}_2) & \boldsymbol{\lambda}_6^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_3(\boldsymbol{x}_2)\} \\
\phi_7^*(\boldsymbol{x}) &= \phi_3(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) & \boldsymbol{\lambda}_7^* &= \{\lambda_3(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2)\} \\
\phi_8^*(\boldsymbol{x}) &= \phi_3(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) & \boldsymbol{\lambda}_8^* &= \{\lambda_3(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2)\} \\
\phi_9^*(\boldsymbol{x}) &= \phi_3(\boldsymbol{x}_1) \cdot \phi_3(\boldsymbol{x}_2) & \boldsymbol{\lambda}_9^* &= \{\lambda_3(\boldsymbol{x}_1), \lambda_3(\boldsymbol{x}_2)\}
\end{aligned}$$

Now, we show another example where different number of eigenfunctions and eigenvalues are used for every dimension. We consider a three-dimensional ($D = 3$) input space, and sets of $J_1 = 2$, $J_2 = 2$ and $J_3 = 3$ eigenfunctions and eigenvalues for the first, second and third dimensions, respectively. The number of new multidimensional eigenfunctions ϕ^* and eigenvalues λ^* is $J_1 \cdot J_2 \cdot J_3 = 2 \cdot 2 \cdot 3 = 12$. The matrix $\mathbb{S} \in \mathbb{R}^{12\times 3}$ is

$$\mathbb{S} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 1 & 3 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix}$$

and the multidimensional eigenfunctions and eigenvalues

$$
\begin{aligned}
\phi_1^* &= \phi_1(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) \cdot \phi_1(\boldsymbol{x}_3) & \boldsymbol{\lambda}_1^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2), \lambda_1(\boldsymbol{x}_3)\} \\
\phi_2^* &= \phi_1(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) \cdot \phi_2(\boldsymbol{x}_3) & \boldsymbol{\lambda}_2^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2), \lambda_2(\boldsymbol{x}_3)\} \\
\phi_3^* &= \phi_1(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) \cdot \phi_3(\boldsymbol{x}_3) & \boldsymbol{\lambda}_3^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2), \lambda_3(\boldsymbol{x}_3)\} \\
\phi_4^* &= \phi_1(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) \cdot \phi_1(\boldsymbol{x}_3) & \boldsymbol{\lambda}_4^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2), \lambda_1(\boldsymbol{x}_3)\} \\
\phi_5^* &= \phi_1(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) \cdot \phi_2(\boldsymbol{x}_3) & \boldsymbol{\lambda}_5^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2), \lambda_2(\boldsymbol{x}_3)\} \\
\phi_6^* &= \phi_1(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) \cdot \phi_3(\boldsymbol{x}_3) & \boldsymbol{\lambda}_6^* &= \{\lambda_1(\boldsymbol{x}_1), \lambda_1(\boldsymbol{x}_2), \lambda_3(\boldsymbol{x}_3)\} \\
\phi_7^* &= \phi_2(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) \cdot \phi_1(\boldsymbol{x}_3) & \boldsymbol{\lambda}_7^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2), \lambda_1(\boldsymbol{x}_3)\} \\
\phi_8^* &= \phi_2(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) \cdot \phi_2(\boldsymbol{x}_3) & \boldsymbol{\lambda}_8^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2), \lambda_2(\boldsymbol{x}_3)\} \\
\phi_9^* &= \phi_2(\boldsymbol{x}_1) \cdot \phi_1(\boldsymbol{x}_2) \cdot \phi_3(\boldsymbol{x}_3) & \boldsymbol{\lambda}_9^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2), \lambda_3(\boldsymbol{x}_3)\} \\
\phi_{10}^* &= \phi_2(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) \cdot \phi_1(\boldsymbol{x}_3) & \boldsymbol{\lambda}_{10}^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2), \lambda_1(\boldsymbol{x}_3)\} \\
\phi_{11}^* &= \phi_2(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) \cdot \phi_2(\boldsymbol{x}_3) & \boldsymbol{\lambda}_{11}^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2), \lambda_2(\boldsymbol{x}_3)\} \\
\phi_{12}^* &= \phi_2(\boldsymbol{x}_1) \cdot \phi_2(\boldsymbol{x}_2) \cdot \phi_3(\boldsymbol{x}_3) & \boldsymbol{\lambda}_{12}^* &= \{\lambda_2(\boldsymbol{x}_1), \lambda_2(\boldsymbol{x}_2), \lambda_3(\boldsymbol{x}_3)\}
\end{aligned}
$$

Appendix G. Multidimensional generalization of covariance functions and spectral densities

G.1. Square Exponential covariance function (k) and spectral density (S)

G.1.1. Using norm-L2 (Euclidean distance)

$$\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{D=2};\ \ \tau_{L2} = |\boldsymbol{x} - \boldsymbol{x}'|_{L2} = \sqrt{(\boldsymbol{x} - \boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')} = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2} = \sqrt{r_1^2 + r_2^2} \in \mathbb{R};\ \ k(\tau_{L2}, \ell) = \exp\left(-\frac{1}{2}\frac{\tau^2}{\ell^2}\right)$$

$$\omega_{L2} = \sqrt{s_1^2 + s_2^2} \in \mathbb{R};\ \ S(\omega_{L2}, \ell) = \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\left(-\frac{1}{2}\ell^2 \omega_{L2}^2\right)$$

$\ell \in \mathbb{R}$	$\begin{aligned} k(\tau_{L2}, \ell) &= k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L2}, \ell) \\ &= \exp\left(-\frac{1}{2}\frac{(\boldsymbol{x} - \boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')}{\ell^2}\right) \\ &= \exp\left(-\frac{1}{2}\frac{\sum_{i=1}^2 (x_i - x_i')^2}{\ell^2}\right) \\ &= \exp\left(-\frac{1}{2}\sum_{i=1}^D \frac{r_i^2}{\ell^2}\right) \end{aligned}$	$\begin{aligned} S(\omega_{L2}, \ell) &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\left(-\frac{1}{2}\ell^2 (s_1^2 + s_2^2)\right) \\ &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^D \ell^2 s_i^2)\right) \end{aligned}$	-ISOTROPIC -SEPARABLE: $k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L2}, \boldsymbol{\ell})$ $= k(\lvert x_1 - x_1'\rvert, \ell_1) k(\lvert x_2 - x_2'\rvert, \ell_2)$ $S(\omega_{L2}, \boldsymbol{\ell}) = S(s_1, \ell_1) S(s_2, \ell_2)$
$\boldsymbol{\ell} \in \mathbb{R}^2$	$\begin{aligned} k(\tau_{L2}, \boldsymbol{\ell}) &= k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L2}, \boldsymbol{\ell}) \\ &= \exp\left(-\frac{1}{2}\sum_{i=1}^D \frac{r_i^2}{\ell_i^2}\right) \end{aligned}$	$S(\omega_{L2}, \boldsymbol{\ell}) = \sqrt{2\pi}^D \cdot \prod_{i=1}^D \ell_i \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^D \ell_i^2 s_i^2\right)$	
$\boldsymbol{\ell} \in \mathbb{R}^2$ Separable kernel	$\begin{aligned} & k(\lvert x_1 - x_1'\rvert, \ell_1) k(\lvert x_2 - x_2'\rvert, \ell_2) \\ &= \exp\left(-\frac{1}{2}\frac{r_1^2}{\ell_1^2}\right) \exp\left(-\frac{1}{2}\frac{r_2^2}{\ell_2^2}\right) \\ &= \exp\left(-\frac{1}{2}\sum_{i=1}^D \frac{r_i^2}{\ell_i^2}\right) \end{aligned}$	$\begin{aligned} & S(s_1, \ell_1) S(s_2, \ell_2) \\ &= \sqrt{2\pi} \cdot \ell_1 \cdot \exp\left(-\frac{1}{2}\ell_1^2 s_1^2\right) \\ &\times \sqrt{2\pi} \cdot \ell_2 \cdot \exp\left(-\frac{1}{2}\ell_2^2 s_2^2\right) \\ &= \sqrt{2\pi}^D \cdot \prod_{i=1}^D \ell_i \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^D \ell_i^2 s_i^2\right) \end{aligned}$	

G.1.2. Using norm-L1

$$\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{D=2};\ \ \tau_{L1} = |\boldsymbol{x} - \boldsymbol{x}'|_{L1} = |x_1 - x_1'| + |x_2 - x_2'| = r_1 + r_2 \in \mathbb{R};\ \ k(\tau_{L1}, \ell) = \exp\left(-\frac{1}{2}\frac{\tau_{L1}^2}{\ell^2}\right)$$

$$\omega_{L1} = s_1 + s_2 \in \mathbb{R};\ \ S(\omega_{L1}, \ell) = \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\left(-\frac{1}{2}\ell^2\omega_{L1}^2\right)$$

$\ell \in \mathbb{R}$	$\begin{aligned} k(\tau_{L1}, \ell) &= k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L1}, \ell) \\ &= \exp\left(-\frac{1}{2}\frac{(r_1 + r_2)(r_1 + r_2)}{\ell^2}\right) \\ &= \exp\left(-\frac{1}{2}\frac{(r_1 + r_2)}{\ell}\frac{(r_1 + r_2)}{\ell}\right) \\ &= \exp\left(-\frac{1}{2}\left(\frac{r_1}{\ell} + \frac{r_2}{\ell}\right)\left(\frac{r_1}{\ell} + \frac{r_2}{\ell}\right)\right) \end{aligned}$	$\begin{aligned} S(\omega_{L1}, \ell) &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\Big(-\frac{1}{2}\ell^2(s_1 + s_2) \\ &\cdot\ (s_1 + s_2)\Big) \\ &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\Big(-\frac{1}{2}\ell(s_1 + s_2) \\ &\cdot\ \ell(s_1 + s_2)\Big) \\ &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\Big(-\frac{1}{2}(\ell s_1 + \ell s_2) \\ &\cdot\ (\ell s_1 + \ell s_2)\Big) \end{aligned}$	-ISOTROPIC -NO SEPARABLE: $k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L1}, \boldsymbol{\ell})$ $\neq k(\lvert x_1 - x_1'\rvert, \ell_1)k(\lvert x_2 - x_2'\rvert, \ell_2)$ $S(\omega_{L1}, \boldsymbol{\ell}) \neq S(s_1, \ell_1)S(s_2, \ell_2)$
$\boldsymbol{\ell} \in \mathbb{R}^2$	$\begin{aligned} k(\tau_{L1}, \boldsymbol{\ell}) &= k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L1}, \boldsymbol{\ell}) \\ &= \exp\left(-\frac{1}{2}\left(\frac{r_1}{\ell_1} + \frac{r_2}{\ell_2}\right)\left(\frac{r_1}{\ell_1} + \frac{r_2}{\ell_2}\right)\right) \end{aligned}$	$\begin{aligned} S(\omega_{L1}, \boldsymbol{\ell}) &= \sqrt{2\pi}^D \cdot \ell_1\ell_2 \cdot \exp\Big(-\frac{1}{2}(\ell_1 s_1 + \ell_2 s_2) \\ &\cdot\ (\ell_1 s_1 + \ell_2 s_2)\Big) \end{aligned}$	
$\boldsymbol{\ell} \in \mathbb{R}^2$ Separable kernel	$\begin{aligned} &k(\lvert x_1 - x_1'\rvert, \ell_1)k(\lvert x_2 - x_2'\rvert, \ell_2) \\ &= \exp\left(-\frac{1}{2}\frac{r_1^2}{\ell_1^2}\right)\exp\left(-\frac{1}{2}\frac{r_2^2}{\ell_2^2}\right) \\ &= \exp\left(-\frac{1}{2}\sum_{i=1}^{2}\frac{r_i^2}{\ell_i^2}\right) \end{aligned}$	$\begin{aligned} &S(s_1, \ell_1)S(s_2, \ell_2) \\ &= \sqrt{2\pi} \cdot \ell_1 \cdot \exp\left(-\frac{1}{2}\ell_1^2 s_1^2\right) \\ &\times\sqrt{2\pi} \cdot \ell_2 \cdot \exp\left(-\frac{1}{2}\ell_2^2 s_2^2\right) \\ &= \sqrt{2\pi}^D \cdot \prod_{i=1}^{D}\ell_i \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\ell_i^2 s_i^2\right) \end{aligned}$	

G.1.3. Using vector difference of inputs

$$\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{D=2};\;\; \boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}' = (x_1 - x_1', x_2 - x_2') = (r_1, r_2) \in \mathbb{R}^2;\;\; k(\boldsymbol{\tau}, \ell) = \exp\left(-\frac{1}{2}\frac{\boldsymbol{\tau}^2}{\ell^2}\right)$$

$$\boldsymbol{\omega} = (s_1, s_2) \in \mathbb{R}^2;\;\; S(\boldsymbol{\omega}, \ell) = \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\left(-\frac{1}{2}\ell^2\boldsymbol{\omega}^2\right)$$

<table>
<tr>
<td>$\ell \in \mathbb{R}$</td>
<td>$$\begin{aligned} k(\boldsymbol{\tau}, \ell) &= k(\boldsymbol{x} - \boldsymbol{x}', \ell) \\ &= \exp\left(-\frac{1}{2}\frac{\boldsymbol{\tau}^\top\boldsymbol{\tau}}{\ell^2}\right) \\ &= \exp\left(-\frac{1}{2}\frac{(r_1, r_2)^\top(r_1, r_2)}{\ell^2}\right) \\ &= \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{r_i^2}{\ell^2}\right) \end{aligned}$$</td>
<td>$$\begin{aligned} S(\boldsymbol{\omega}, \ell) &= S(\boldsymbol{x} - \boldsymbol{x}', \ell) \\ &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\left(-\frac{1}{2}\ell^2\boldsymbol{\omega}^\top\boldsymbol{\omega}\right) \\ &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\Big(-\frac{1}{2}\ell^2(s_1, s_2)^\top \\ &\quad \cdot \;(s_1, s_2)\Big) \\ &= \sqrt{2\pi}^D \cdot \ell^D \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\ell^2 s_i^2\right) \end{aligned}$$</td>
<td rowspan="3">-ISOTROPIC

-SEPARABLE:

$k(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\ell})$
$= k(\lvert x_1 - x_1'\rvert, \ell_1)k(\lvert x_2 - x_2'\rvert, \ell_2)$

$S(\boldsymbol{\omega}, \boldsymbol{\ell}) = S(s_1, \ell_1)S(s_2, \ell_2)$</td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$</td>
<td>$$\begin{aligned} k(\boldsymbol{\tau}, \boldsymbol{\ell}) &= k(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\ell}) \\ &= \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{r_i^2}{\ell_i^2}\right) \end{aligned}$$</td>
<td>$$S(\boldsymbol{\omega}, \ell) = \sqrt{2\pi}^D \cdot \prod_{i=1}^{D}\ell_i \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\ell_i^2 s_i^2\right)$$</td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$
Separable kernel</td>
<td>$$\begin{aligned} &k(\lvert x_1 - x_1'\rvert, \ell_1)k(\lvert x_2 - x_2'\rvert, \ell_2) \\ &= \exp\left(-\frac{1}{2}\frac{r_1^2}{\ell_1^2}\right)\exp\left(-\frac{1}{2}\frac{r_2^2}{\ell_2^2}\right) \\ &= \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{r_i^2}{\ell_i^2}\right) \end{aligned}$$</td>
<td>$$\begin{aligned} &S(s_1, \ell_1)S(s_2, \ell_2) \\ &= \sqrt{2\pi}\cdot\ell_1\cdot\exp\left(-\frac{1}{2}\ell_1^2 s_1^2\right) \\ &\times\sqrt{2\pi}\cdot\ell_2\cdot\exp\left(-\frac{1}{2}\ell_2^2 s_2^2\right) \\ &= \sqrt{2\pi}^D \cdot \prod_{i=1}^{D}\ell_i \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\ell_i^2 s_i^2\right) \end{aligned}$$</td>
</tr>
</table>

G.2. Matern($\nu = 1/2$) covariance function (k) and spectral density (S)

G.2.1. Using norm-L2 (Euclidean distance)

$$\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{D=2};\;\; \tau_{L2} = |\boldsymbol{x} - \boldsymbol{x}'|_{L2} = \sqrt{(\boldsymbol{x} - \boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')} = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2} = \sqrt{r_1^2 + r_2^2} \in \mathbb{R};\;\; k(\tau_{L2}, \ell) = \exp\left(-\frac{\tau_{L2}}{\ell}\right)$$

$$\omega_{L2} = \sqrt{s_1^2 + s_2^2} \in \mathbb{R};\;\; S_\nu(\omega_{L2}, \ell) = \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\ell^{2\nu}} \left(\frac{2\nu}{\ell^2} + \omega_{L2}^2\right)^{-(\nu + D/2)} \;\; ; \;\; S_{1/2}(\omega_{L2}, \ell) = \frac{2^D \pi^{\frac{D}{2}} \Gamma(\frac{D+1}{2})}{\sqrt{\pi}\ell} \left(\frac{1}{\ell^2} + \omega_{L2}^2\right)^{-\frac{D+1}{2}}$$

<table>
<tr>
<td>$\ell \in \mathbb{R}$</td>
<td>$$\begin{aligned} k(\tau_{L2}, \ell) &= k(|\boldsymbol{x} - \boldsymbol{x}'|_{L2}, \ell) \\ &= \exp\left(-\frac{\sqrt{(\boldsymbol{x} - \boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')}}{\ell}\right) \\ &= \exp\left(-\sqrt{\frac{(\boldsymbol{x} - \boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')}{\ell^2}}\right) \\ &= \exp\left(-\sqrt{\frac{\sum_{i=1}^2 (x_i - x_i')^2}{\ell^2}}\right) \\ &= \exp\left(-\sqrt{\sum_{i=1}^2 \frac{r_i^2}{\ell^2}}\right) \end{aligned}$$</td>
<td>$$\begin{aligned} D &= 2 \\ S_{1/2}(\omega_{L2}, \ell) &= \frac{2\pi}{\ell}\left(\frac{1}{\ell^2} + \omega_{L2}^2\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \ell^2\omega_{L2}^2\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \ell^2 s_1^2 + \ell^2 s_2^2\right)^{-\frac{3}{2}} \\ D &= 3 \\ S_{1/2}(\omega_{L2}, \ell) &= \frac{8\pi}{\ell}\left(\frac{1}{\ell^2} + \omega_{L2}^2\right)^{-2} \\ &= 8\pi\ell^3\left(1 + \ell^2\omega_{L2}^2\right)^{-2} \\ &= 8\pi\ell^3\left(1 + \ell^2 s_1^2 + \ell^2 s_2^2 + \ell^2 s_3^2\right)^{-2} \end{aligned}$$</td>
<td>-ISOTROPIC

-NO SEPARABLE:

$k(|\boldsymbol{x} - \boldsymbol{x}'|_{L2}, \boldsymbol{\ell})$
$\neq k(|x_1 - x_1'|, \ell_1)k(|x_2 - x_2'|, \ell_2)$

$S_{1/2}(\omega_{L2}, \ell) \neq S_{1/2}(s_1, \ell)S_{1/2}(s_2, \ell)$</td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$</td>
<td>$$\begin{aligned} k(\tau_{L2}, \boldsymbol{\ell}) &= k(|\boldsymbol{x} - \boldsymbol{x}'|_{L2}, \boldsymbol{\ell}) \\ &= \exp\left(-\sqrt{\sum_{i=1}^2 \frac{r_i^2}{\ell_i^2}}\right) \end{aligned}$$</td>
<td>$$\begin{aligned} D &= 2 \\ S_{1/2}(\omega_{L2}, \boldsymbol{\ell}) &= 2\pi\ell_1\ell_2\left(1 + \sum_{i=1}^D \ell_i^2 s_i^2\right)^{-\frac{3}{2}} \\ S_{1/2}(\omega_{L2}, \boldsymbol{\ell}) &= 2\pi\prod_{i=1}^D \ell_i \left(1 + \sum_{i=1}^D \ell_i^2 s_i^2\right)^{-\frac{3}{2}} \end{aligned}$$</td>
<td></td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$
Separable
kernel</td>
<td>$$\begin{aligned} & k(|x_1 - x_1'|, \ell_1)k(|x_2 - x_2'|, \ell_2) \\ = \;& \exp\left(-\frac{r_1}{\ell_1}\right)\exp\left(-\frac{r_2}{\ell_2}\right) \\ = \;& \exp\left(-\sum_{i=1}^2 \frac{r_i}{\ell_i}\right) \end{aligned}$$</td>
<td>$$\begin{aligned} & S_{1/2}(s_1, \ell_1)S_{1/2}(s_2, \ell_2) \\ = \;& \frac{2}{\ell_1}\left(\frac{1}{\ell_1^2} + s_1^2\right)^{-1} \cdot \frac{2}{\ell_2}\left(\frac{1}{\ell_2^2} + s_2^2\right)^{-1} \\ = \;& 4\ell_1\ell_2\left(1 + \ell_1^2 s_1^2\right)^{-1}\left(1 + \ell_2^2 s_2^2\right)^{-1} \end{aligned}$$</td>
<td></td>
</tr>
</table>

G.2.2. Using norm-L1

$$\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{D=2};\;\; \tau_{L1} = |\boldsymbol{x} - \boldsymbol{x}'|_{L1} = |x_1 - x_1'| + |x_2 - x_2'| = r_1 + r_2 \in \mathbb{R};\;\; k(\tau_{L1}, \ell) = \exp\left(-\frac{\tau_{L1}}{\ell}\right)$$

$$\omega_{L1} = s_1 + s_2 \in \mathbb{R};\;\; S_\nu(\omega_{L2}, \ell) = \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\ell^{2\nu}} \left(\frac{2\nu}{\ell^2} + \omega_{L2}^2\right)^{-(\nu + D/2)};\;\; S_{1/2}(\omega_{L2}, \ell) = \frac{2^D \pi^{\frac{D}{2}} \Gamma(\frac{D+1}{2})}{\sqrt{\pi}\ell} \left(\frac{1}{\ell^2} + \omega_{L2}^2\right)^{-\frac{D+1}{2}}$$

<table>
<tr>
<td>$\ell \in \mathbb{R}$</td>
<td>$$\begin{aligned} k(\tau_{L1}, \ell) &= k(|\boldsymbol{x} - \boldsymbol{x}'|_{L1}, \ell) \\ &= \exp\left(-\frac{\tau}{\ell}\right) \\ &= \exp\left(-\frac{r_1 + r_2}{\ell}\right) \\ &= \exp\left(-\sum_{i=1}^{D} \frac{r_i}{\ell}\right) \end{aligned}$$</td>
<td>$$\begin{aligned} D &= 2 \\ S_{1/2}(\omega_{L1}, \ell) &= \frac{2\pi}{\ell}\left(\frac{1}{\ell^2} + \omega_{L1}^2\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \ell^2\omega_{L1}^2\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \ell^2(s_1 + s_2)(s_1 + s_2)\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + (\ell s_1 + \ell s_2)(\ell s_1 + \ell s_2)\right)^{-\frac{3}{2}} \end{aligned}$$
$$\begin{aligned} D &= 3 \\ S_{1/2}(\omega_{L1}, \ell) &= \frac{8\pi}{\ell}\left(\frac{1}{\ell^2} + \omega_{L1}^2\right)^{-2} \\ &= 8\pi\ell^3\left(1 + \ell^2\omega_{L1}^2\right)^{-2} \\ &= 8\pi\ell^3\left(1 + \ell^2(s_1 + s_2 + s_3)(s_1 + s_2 + s_3)\right)^{-2} \\ &= 8\pi\ell^3\left(1 + (\ell s_1 + \ell s_2 + \ell s_3)(\ell s_1 + \ell s_2 + \ell s_3)\right)^{-2} \end{aligned}$$</td>
<td rowspan="3">-ISOTROPIC

-SEPARABLE:

$k(|\boldsymbol{x} - \boldsymbol{x}'|_{L1}, \boldsymbol{\ell})$
$= k(|x_1 - x_1'|, \ell_1)k(|x_2 - x_2'|, \ell_2)$

IT SHOULD BE SEPARABLE IN THE SPECTRAL DENSITY AS WELL?
$S_{1/2}(\omega_{L1}, \ell)$ should be equal to $S_{1/2}(s_1, \ell)S_{1/2}(s_2, \ell)$</td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$</td>
<td>$$k(\tau_{L1}, \boldsymbol{\ell}) = \exp\left(-\sum_{i=1}^{D} \frac{r_i}{\ell_i}\right)$$</td>
<td>$$\begin{aligned} D &= 2 \\ S_{1/2}(\omega_{L1}, \boldsymbol{\ell}) &= 2\pi\ell_1\ell_2\left(1 + (\ell_1 s_1 + \ell_2 s_2)(\ell_1 s_1 + \ell_2 s_2)\right)^{-\frac{3}{2}} \\ &= 2\pi\ell_1\ell_2\left(1 + (\ell_1 s_1 + \ell_2 s_2)(\ell_1 s_1 + \ell_2 s_2)\right)^{-\frac{3}{2}} \end{aligned}$$</td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$
Separable kernel</td>
<td>$$\begin{aligned} &k(|x_1 - x_1'|, \ell_1)k(|x_2 - x_2'|, \ell_2) \\ &= \exp\left(-\frac{r_1}{\ell_1}\right)\exp\left(-\frac{r_2}{\ell_2}\right) \\ &= \exp\left(-\sum_{i=1}^{D} \frac{r_i}{\ell_i}\right) \end{aligned}$$</td>
<td>$$\begin{aligned} &S_{1/2}(s_1, \ell_1)S_{1/2}(s_2, \ell_2) \\ &= \frac{2}{\ell_1}\left(\frac{1}{\ell_1^2} + s_1^2\right)^{-1} \cdot \frac{2}{\ell_2}\left(\frac{1}{\ell_2^2} + s_2^2\right)^{-1} \\ &= 4\ell_1\ell_2\left(1 + \ell_1^2 s_1^2\right)^{-1}\left(1 + \ell_2^2 s_2^2\right)^{-1} \end{aligned}$$</td>
</tr>
</table>

G.2.3. Using the vector difference of inputs

$$\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{D=2};\;\; \boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}' = (x_1 - x_1', x_2 - x_2') = (r_1, r_2) \in \mathbb{R}^2;\;\; k(\boldsymbol{\tau}, \ell) = \exp\left(-\frac{\boldsymbol{\tau}}{\ell}\right)$$

$$\boldsymbol{\omega} = (s_1, s_2) \in \mathbb{R}^2;\;\; S_\nu(\boldsymbol{\omega}, \ell) = \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\ell^{2\nu}} \left(\frac{2\nu}{\ell^2} + \boldsymbol{\omega}^2\right)^{-(\nu + D/2)} ;\;\; S_{1/2}(\boldsymbol{\omega}, \ell) = \frac{2^D \pi^{\frac{D}{2}} \Gamma(\frac{D+1}{2})}{\sqrt{\pi}\ell} \left(\frac{1}{\ell^2} + \boldsymbol{\omega}^2\right)^{-\frac{D+1}{2}}$$

<table>
<tr>
<td>$\ell \in \mathbb{R}$</td>
<td>$\begin{aligned} k(\boldsymbol{\tau}, \ell) &= k(\boldsymbol{x} - \boldsymbol{x}', \ell) \\ &= \exp\left(-\frac{\boldsymbol{\tau}}{\ell}\right) \\ &= \exp\left(-\frac{(r_1, r_2)}{\ell}\right) \\ &\quad \text{(using dot product?)} \\ &= \exp\left(-\sum_{i=1}^{2} \frac{r_i}{\ell}\right) \end{aligned}$</td>
<td>$\begin{aligned} D &= 2 \\ S_{1/2}(\boldsymbol{\omega}, \ell) &= \frac{2\pi}{\ell}\left(\frac{1}{\ell^2} + \boldsymbol{\omega}^2\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \ell^2\boldsymbol{\omega}^2\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \ell^2(s_1, s_2)^\top(s_1, s_2)\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \ell^2(s_1^2 + s_2^2)\right)^{-\frac{3}{2}} \\ &= 2\pi\ell^2\left(1 + \sum_{i=1}^{D} \ell^2 s_i^2\right)^{-\frac{3}{2}} \end{aligned}$</td>
<td rowspan="3">-ISOTROPIC

-SEPARABLE:
$k(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\ell})$
$= k(\lvert x_1 - x_1'\rvert, \ell_1)k(\lvert x_2 - x_2'\rvert, \ell_2)$

IT SHOULD BE SEPARABLE IN THE SPECTRAL DENSITY AS WELL?
$S_{1/2}(\boldsymbol{\omega}, \ell)$ should be equal to $S_{1/2}(s_1, \ell)S_{1/2}(s_2, \ell)$</td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$</td>
<td>$\begin{aligned} k(\boldsymbol{\tau}, \boldsymbol{\ell}) &= \exp\left(-\frac{\boldsymbol{\tau}}{\boldsymbol{\ell}}\right) \\ &= \exp\left(-\frac{(r_1, r_2)}{(\ell_1, \ell_2)}\right) \\ &\quad \text{(using dot product?)} \\ &= \exp\left(-\sum_{i=1}^{2} \frac{r_i}{\ell_i}\right) \end{aligned}$</td>
<td>$S_{1/2}(\boldsymbol{\omega}, \boldsymbol{\ell}) = 2\pi\ell_1\ell_2\left(1 + \sum_{i=1}^{D} \ell_i^2 s_i^2\right)^{-\frac{3}{2}}$</td>
</tr>
<tr>
<td>$\boldsymbol{\ell} \in \mathbb{R}^2$
Separable kernel</td>
<td>$\begin{aligned} &k(\lvert x_1 - x_1'\rvert, \ell_1)k(\lvert x_2 - x_2'\rvert, \ell_2) \\ &= \exp\left(-\frac{r_1}{\ell_1}\right)\exp\left(-\frac{r_2}{\ell_2}\right) \\ &= \exp\left(-\sum_{i=1}^{2} \frac{r_i}{\ell_i}\right) \end{aligned}$</td>
<td>$\begin{aligned} &S_{1/2}(s_1, \ell_1)S_{1/2}(s_2, \ell_2) \\ &= \frac{2}{\ell_1}\left(\frac{1}{\ell_1^2} + s_1^2\right)^{-1} \cdot \frac{2}{\ell_2}\left(\frac{1}{\ell_2^2} + s_2^2\right)^{-1} \\ &= 4\ell_1\ell_2\left(1 + \ell_1^2 s_1^2\right)^{-1}\left(1 + \ell_2^2 s_2^2\right)^{-1} \end{aligned}$</td>
</tr>
</table>

G.3. Matern($\nu = 3/2$) covariance function (k) and spectral density (S)

G.3.1. Using norm-L2 (Euclidean distance)

$$\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{D=2};\;\; \tau_{L2} = |\boldsymbol{x} - \boldsymbol{x}'|_{L2} = \sqrt{(\boldsymbol{x} - \boldsymbol{x}')(\boldsymbol{x} - \boldsymbol{x}')} = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2} = \sqrt{r_1^2 + r_2^2} \in \mathbb{R};\;\; k(\tau_{L2}, \ell) = \left(1 + \frac{\sqrt{3}\tau}{\ell}\right) \exp\left(-\frac{\sqrt{3}\tau}{\ell}\right)$$

$$\omega_{L2} = \sqrt{s_1^2 + s_2^2} \in \mathbb{R};\;\; S_\nu(\omega_{L2}) = \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\ell^{2\nu}} \left(\frac{2\nu}{\ell^2} + \omega_{L2}^2\right)^{-(\nu + D/2)};\;\; S_{3/2}(\omega_{L2}) = \frac{2^D \pi^{D/2} \Gamma(\frac{D+3}{2}) \sqrt{3}^3}{\frac{1}{2}\sqrt{\pi}\ell^3} \left(\frac{3}{\ell^2} + \omega_{L2}^2\right)^{-\frac{D+3}{2}}$$

$\ell \in \mathbb{R}$	$k(\tau_{L2}, \ell) = k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L2}, \ell)$ $= \left(1 + \frac{\sqrt{3}\sqrt{\sum_{i=1}^2 r_i^2}}{\ell}\right) \exp\left(-\frac{\sqrt{3}\sqrt{\sum_{i=1}^2 r_i^2}}{\ell}\right)$ $= \left(1 + \sqrt{\frac{\sum_{i=1}^2 3r_i^2}{\ell^2}}\right) \exp\left(-\sqrt{\frac{\sum_{i=1}^2 3r_i^2}{\ell^2}}\right)$	$D = 2$ $S_{3/2}(\omega_{L2}, \ell) = \frac{6\pi\sqrt{3}^3}{\ell^3}\left(\frac{3}{\ell^2} + \omega^2\right)^{-\frac{5}{2}}$ $= 6\pi\sqrt{3}^3 \ell^2 \left(3 + \ell^2\omega^2\right)^{-\frac{5}{2}}$ $= 6\pi\sqrt{3}^3 \ell^2 \left(3 + \ell^2 s_1^2 + \ell^2 s_2^2\right)^{-\frac{5}{2}}$ $D = 3$ $S_{3/2}(\omega_{L2}, \ell) = 32\pi\sqrt{3}^3 \ell^3 \left(3 + \ell^2\omega^2\right)^{-3}$	-ISOTROPIC -NO SEPARABLE: $k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L2}, \boldsymbol{\ell})$ $\neq k(\lvert x_1 - x_1'\rvert, \ell_1) k(\lvert x_2 - x_2'\rvert, \ell_2)$ $S_{3/2}(\omega_{L2}, \ell) \neq S_{3/2}(s_1, \ell) S_{1/2}(s_2, \ell)$
$\boldsymbol{\ell} \in \mathbb{R}^2$	$k(\tau_{L2}, \boldsymbol{\ell}) = k(\lvert\boldsymbol{x} - \boldsymbol{x}'\rvert_{L2}, \boldsymbol{\ell})$ $= \left(1 + \sqrt{\frac{\sum_{i=1}^2 3r_i^2}{\ell_i^2}}\right) \exp\left(-\sqrt{\frac{\sum_{i=1}^2 3r_i^2}{\ell_i^2}}\right)$	$S_{3/2}(\omega_{L2}, \boldsymbol{\ell}) = 6\pi\sqrt{3}^3 \ell_1 \ell_2 \left(3 + \ell_1^2 s_1^2 + \ell_2^2 s_2^2\right)^{-\frac{5}{2}}$ $= 6\pi\sqrt{3}^3 \prod_{i=1}^D \ell_i \left(3 + \sum_{i=1}^D \ell_i^2 s_i^2\right)^{-\frac{5}{2}}$	
$\boldsymbol{\ell} \in \mathbb{R}^2$ Separable kernel	$k(\lvert x_1 - x_1'\rvert, \ell_1) k(\lvert x_2 - x_2'\rvert, \ell_2)$ $= \left(1 + \sqrt{3}\frac{r_1}{\ell_1}\right) \exp\left(-\sqrt{3}\frac{r_1}{\ell_1}\right)$ $\cdot \left(1 + \sqrt{3}\frac{r_2}{\ell_2}\right) \exp\left(-\sqrt{3}\frac{r_2}{\ell_2}\right)$ $= \left(1 + \sqrt{\frac{3r_1^2}{\ell_1^2}}\right)\left(1 + \sqrt{\frac{3r_2^2}{\ell_2^2}}\right) \exp\left(-\sum_{i=1}^2 \sqrt{\frac{3r_i^2}{\ell_i^2}}\right)$ $= \left(1 + \sum_{i=1}^2 \sqrt{\frac{3r_i^2}{\ell_i^2}} + \frac{3r_1^2 r_2^2}{\ell_1^2 \ell_2^2}\right) \exp\left(-\sum_{i=1}^2 \sqrt{\frac{3r_i^2}{\ell_i^2}}\right)$	$S_{3/2}(s_1, \ell_1) S_{3/2}(s_2, \ell_2)$ $= \frac{4\sqrt{3}^3}{\ell_1^3}\left(\frac{3}{\ell_1^2} + s_1^2\right)^{-3} \cdot \frac{4\sqrt{3}^3}{\ell_2^3}\left(\frac{3}{\ell_2^2} + s_2^2\right)^{-3}$ $= 4^2\sqrt{3}^6 \boldsymbol{\ell}^\top \boldsymbol{\ell} \left(3 + \ell_1^2 s_1^2\right)^{-2} \left(3 + \ell_2^2 s_2^2\right)^{-2}$	

Acknowledgment

References

Adler, R. J. (1981). *The geometry of random fields*, volume 62. Siam.

Akhiezer, N. and Glazman, I. (1993). Theory of linear operators in hilbert space (ungar, new york, 1963). *Vol. II* pages 121–126.

Arthur, D. (1979). Baker cth, the numerical treatment of integral equations (clarendon press; oxford university press, 1978), xiv+ 1034 pp.,£ 22–50. *Proceedings of the Edinburgh Mathematical Society* **22,** 67–67.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70,** 825–848.

Briol, F.-X., Oates, C., Girolami, M., Osborne, M. A., Sejdinovic, D., et al. (2015). Probabilistic integration: A role in statistical computation? *arXiv preprint arXiv:1512.00933* .

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of markov chain monte carlo*. CRC press.

Bui, T. D. and Turner, R. E. (2014). Tree-structured gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 2213–2221.

Bui, T. D., Yan, J., and Turner, R. E. (2017). A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research* **18,** 3649–3720.

Burt, D., Rasmussen, C. E., and van der Wilk, M. (2019). Explicit rates of convergence for sparse variational inference in Gaussian process regression. *arXiv preprint arXiv:1903.03571* .

Carlin, B. P., Gelfand, A. E., and Banerjee, S. (2014). *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software* **76,**.

Cramér, H. and Leadbetter, M. R. (2013). *Stationary and related stochastic processes: Sample function properties and their applications*. Courier Corporation.

Csató, L., Fokoué, E., Opper, M., Schottky, B., and Winther, O. (2000). Efficient approaches to gaussian process classification. In *Advances in neural information processing systems*, pages 251–257.

Cunningham, J. P., Shenoy, K. V., and Sahani, M. (2008). Fast gaussian process methods for point process intensity estimation. In *Proceedings of the 25th international conference on Machine learning*, pages 192–199. ACM.

Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence* **37,** 408–423.

Diggle, P., Ribeiro, P., and Geostatistics, M.-b. (2007). Springer series in statistics.

Diggle, P. J. (2013). *Statistical analysis of spatial and spatio-temporal point patterns*. Chapman and Hall/CRC.

Furrer, E. M. and Nychka, D. W. (2007). A framework to understand the asymptotic properties of kriging and splines. *Journal of the Korean Statistical Society* **36,** 57–76.

Gal, Y. and Turner, R. (2015). Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *International Conference on Machine Learning*, pages 655–664.

Gibbs, M. N. and MacKay, D. J. (2000). Variational gaussian process classifiers. *IEEE Transactions on Neural Networks* **11,** 1458–1464.

Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proc. R. Soc. A* **471,** 20150142.

Hensman, J., Durrande, N., and Solin, A. (2017). Variational fourier features for gaussian processes. *The Journal of Machine Learning Research* **18,** 5537–5588.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835* .

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research* **14,** 1303–1347.

Kimeldorf, G. S. and Wahba, G. (1970). A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics* **41,** 495–502.

Lázaro Gredilla, M. (2010). Sparse gaussian processes for large-scale machine learning.

Loève, M. (1977). Probability theory. 1977.

Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.

Moore, D. and Russell, S. J. (2015). Gaussian process random fields. In *Advances in Neural Information Processing Systems*, pages 3357–3365.

Neal, R. M. (1997). Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026* .

Paciorek, C. J. (2007). Computational techniques for spatial logistic regression with large data sets. *Computational statistics & data analysis* **51,** 3631–3653.

Paciorek, C. J. (2007b). Bayesian smoothing with gaussian processes using fourier basis functions in the spectralgp package. *Journal of statistical software* **19,** nihpa22751.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research* **6,** 1939–1959.

Quiñonero-Candela, J., Rasmussen, C. E., Figueiras-Vidal, A. R., et al. (2010). Sparse spectrum gaussian process regression. *Journal of Machine Learning Research* **11,** 1865–1881.

Rahimi, A. and Recht, B. (2008). Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184.

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian process for machine learning*. MIT press.

Ro, L. C. and Opper, M. (2001). Sparse online gaussian processes. *Neural Comput.* **14,** 641–668.

Roberts, S. J. (2010). *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, University of Oxford.

Saatçi, Y. (2012). *Scalable inference for structured Gaussian process models*. PhD thesis, Citeseer.

Seeger, M., Williams, C., and Lawrence, N. (2003). Fast forward selection to speed up sparse gaussian process regression. In *Artificial Intelligence and Statistics 9*, number EPFL-CONF-161318.

Smola, A. J. and Bartlett, P. L. (2001). Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pages 619–625.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.

Snelson, E. and Ghahramani, Z. (2007). Local and global sparse gaussian process approximations. In *Artificial Intelligence and Statistics*, pages 524–531.

Solin, A. and Särkkä, S. (2018). Hilbert space methods for reduced-rank gaussian process regression. *arXiv preprint arXiv:1401.5508* .

Särkkä, S., Solin, A., and Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering. *IEEE Signal Processing Magazine* **30,** 51–61.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574.

Trees, H. (1968). Detection, estimation and modulation theory, vol. 1.

Urtasun, R. and Darrell, T. (2008). Sparse probabilistic regression for activity-independent human pose inference. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Vanhatalo, J., Pietiläinen, V., and Vehtari, A. (2010). Approximate inference for disease mapping with sparse gaussian processes. *Statistics in medicine* **29,** 1580–1607.

Wahba, G. (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society. Series B (Methodological)* pages 364–372.

Wahba, G. (1990). *Spline models for observational data*, volume 59. Siam.

Williams, C. K. and Barber, D. (1998). Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20,** 1342–1351.

Williams, C. K. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688.

Wilson, A. and Nickisch, H. (2015). Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65,** 95–114.

Wood, S. N. (2017). *Generalized additive models: an introduction with R*. Chapman and Hall/CRC.