Hilbert space methods to approximate Gaussian processes using Stan

ARTICLE HISTORY
Compiled February 27, 2019

ABSTRACT

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

Contents

1. Introduction

Gaussian processes (GPs) are natural (Paul: what does "natural" imply and as compared to what?) and flexible non-parametric models to define probability distributions over multi-dimensional functions [Neal, 1997, Rasmussen and Williams, 2006]. This allows applying probabilistic inference to estimate functional relationships between one or more covariates and a single response variable. The defining feature of GPs is that the function values are assumed to jointly follow a multivariate normal distribution, which is characterized primarily by a covariance function, sometimes also referred to as covariance kernel. 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. Several types of covariance functions with varying properties can be used, and they may even be combined for further increased flexibility. GPs are often referred to as non-parametric models because the number of parameters is not fixed, but rather increases with the number of data points, which allows to adapt model complexity to the data. As such, the term 'non-parametric' does not imply having no parameters but actually having a lot of them. 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 [Sarkka et al., 2013], as well as Bayesian optimization and probabilistic numerics [Briol et al., 2015, Hennig et al., 2015, Roberts, 2010].

One of the main limitations of exact GPs is that computational demands and memory requirements scale as $O(n^3)$ and $O(n^2)$, respectively, with the number n of observations in the data used to fit the model. This effectively limits their application to rather small data sets of a few thousand observations at most. This problem becomes especially severe when performing full Bayesian inference via sampling methods, where in each sampling step we need to invert the Gram matrix of the covariance function, usually through Cholesky factorization. To alleviate these computational demands, several approximate methods have been proposed, which we will briefly review in the following.

Sparse GPs are based on building low-rank approximations of the covariance matrix of the complete data by means of selecting a subset m of training points on which to base computation. They reduce the dimension of the posterior distribution thereby reducing the memory requirements to $O(nm)$ and computational complexity to $O(nm^2)$ for some $m < n$. This approach still requires Cholesky factorization although now based on a smaller covariance matrix. While computational complexity is reduced considerably, ensuring convergence to the exact GP model is not necessarily straightforward (Paul: this may need a citation). As such, reducing the amount of data while maintaining the correct posterior distribution is not a simple procedure. 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]. (Gabi: Could you choose between one of these two set of sentences in different colors?) More recent developments in the context of sparse GPs include points-based sparse approximation methods [Wilson and Nickisch, 2015], which have been further developed by [Bui et al., 2017] to perform approximations at inference time rather than at modeling time. New developments in the context of sparse GPs include a structured kernel interpolation (SKI) method [Wilson and Nickisch, 2015]. More recently [Bui et al., 2017] revisit the inducing points-based sparse

approximation methods to perform approximations at inference time rather than at modeling time.

(Paul: This whole paragraph is unclear to me. Are all of these approaches approximations via splines, or are the splines just special cases of a more general method? In any case, I think the paragraph needs substantial revision). Other global approach make use of the spectral analysis and series expansions of Gaussian processes [Adler, 1981, Cramér and Leadbetter, 2013, Loève, 1977, Trees, 1968]. 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, but they do not generally have correspondence with GP models, and they do not have the flexibility of choosing different covariance functions as model structure. The Sparse Spectrum GP is based on a sparse approximation to the frequency domain representation of a GP [Gal and Turner, 2015,?, Lázaro Gredilla, 2010, QuiÃ±onero-Candela et al., 2010]. Recently [Hensman et al., 2017] presented a variational Fourier feature approximation for Gaussian processes that was derived for the Matern class of kernels. Random Fourier Features [Rahimi and Recht, 2008, 2009] is a method for approximating kernels.

Yet another approach to approximate GPs – on which we will focus in the present paper – was proposed by Solin and Särkkä [2018] who suggested to use reduced-rank approximations of the covariance kernel. This method decomposes the kernel into basis functions which represent the GP as a linear model and thus makes its estimation considerably faster. In principal, we are free in the choice of basis functions, but the Laplace eigenfunctions were found to be particularily appealing [Solin and Särkkä, 2018]. Not only can they be computed analytically, but they are also independent of the particular choice of the covariance kernel including the hyperparameters. The latter comes with several advantages when performing inference via algorithms that require differentiation of the posterior density (Paul: cite something here). In a fully Bayesian framework based on sampling methods, the proposed approximate GP model has a computational complexity of just $O(nm + m)$, with $m << n$ in every sampling step. The posterior distribution of the reduced-rank GP approximation is of dimension m and thus considerably smaller than in the case of exact GPs. This not only lowers the memory requirements and computational complexity but also reduces the correlations between latent values and the corresponding hyperparameters of the GP, which can help to further improve sampling efficiency.

While Solin and Särkkä [2018] fully developed the mathematical theory behind reduced-rank approximations of GPs, they did not put much effort in analyzing 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. However, for practical implementations of approximate GPs, the latter factors are critical and thus require further investigation. In the present paper, we will fill this gap by providing practical recommendations for the choice of these factors based on the recognized relations among them. What is more, we provide intuitive visualizations of these relations that will help users to improve performance and save computation time, while at the same time maintaining close approximations of exact GPs.

Although there are several GP specific software packages available to date (Paul: cite those. Gabi: This full paragraph comes from Aki. I think he refers to some Splines methods and maybe others approaches as well), 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 a specific GP models, but instead are interested in how GPs can be easily used as modular components in probabilistic programming

frameworks such as Stan [Carpenter et al., 2017]. To this end, we want to make full Bayesian inference over Gaussian process models accessible to a broader audience by making them easy, fast and save to apply.

The remainder of the paper is structered 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.

(Paul: The content and the purpose of the following sentences is not fully clear to me. Do we need them in the Introduction or can they be removed, or put into the next section? Gabi: In this paragraph I tried to state two differences I found in the literature between our method and the Sparse Spectrum GP and the Splines method. But I not know where locate them.) 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.

2. Method

2.1. Gaussian process regression

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 every 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 prior distributions over function spaces for non-parametric regression in a Bayesian setting. Consider a data set $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$, where y_n is a noisy observation of an unknown function f at $\mathbf{x}_n \in \mathbb{R}^D$, i.e. $y_n = f(\mathbf{x}_n) + \epsilon_n$. Our goal is to predict the value of the function $f^* = f(\mathbf{x}^*)$ evaluated at a new input point $\mathbf{x}^* \in \mathbb{R}^D$. That is, we want to obtain the predictive distribution $p(f^*|\mathcal{D})$ of f^* conditioned on the data $\mathcal{D}$.

We assume a Gaussian process prior for $f \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, 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,

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

The mean and covariance functions completely characterize the Gaussian process prior, and represent how the random functions behave on average and how the different points in the input space co-vary with respect to each other, respectively. The Gaussian process prior can be interpreted such as any finite number of function values $\mathbf{f} =$

$\{f(\mathbf{x}_n)\}_{n=1}^N$ is multivariate Gaussian,

$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, K),$$

where $\boldsymbol{\mu} = \{\mu(\mathbf{x}_n)\}_{n=1}^N$ is the multivariate mean and K the covariance matrix, where the element $K_{i,j}$ are formed by evalating the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ at input values $\mathbf{x}_i$ and $\mathbf{x}_j$. The joint distribution of $\mathbf{f}$ and f^* is also a multivariate Gaussian as,

$$p(\mathbf{f}, \mathbf{f}') = \mathcal{N}\left(\left[\begin{array}{c} \mathbf{f} \\ \mathbf{f}' \end{array}\right] | 0, \left[\begin{array}{cc} K_{\mathbf{f},\mathbf{f}} & K_{\mathbf{f},f^*} \\ K_{f^*,\mathbf{f}} & K_{f^*,f^*} \end{array}\right]\right),$$

where $K_{\mathbf{f},f^*}$ defines the covariances between $\mathbf{f}$ and f^*, and K_{f^*,f^*} defines the variance of f^*. By using the conditioning properties of a multivariate Gaussian, we can compute analytically the predictive distribution for f^* given $\mathbf{f}$,

$$p(f^*|\mathbf{f}) = \mathcal{N}(f^*|K_{f^*,\mathbf{f}}K_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}, K_{f^*,f^*} - K_{f^*,\mathbf{f}}K_{\mathbf{f},\mathbf{f}}^{-1}K_{\mathbf{f},f^*})$$

The joint distribution of observations $\mathbf{y} = \{y_n\}_{n=1}^N$ and function values $\mathbf{f}$ and f^*, $p(\mathbf{y}, \mathbf{f}, f^*)$, is the product of the conditional distribution for $\mathbf{y}$ given $\mathbf{f}$ and the joint distribution for $\mathbf{f}$ and f^*,

$$p(\mathbf{y}, \mathbf{f}, f^*) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, f^*)$$

Integrating the joint distribution $p(\mathbf{y}, \mathbf{f}, f^*)$ over the uncertainty of $\mathbf{f}$ and then conditioning on the observations $\mathbf{y}$ following the Bayes's rule yields the posterior predictive distribution for f^* given the observations $\mathbf{y}$,

$$p(f^*|\mathbf{y}) = \frac{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, f^*)\mathrm{d}\mathbf{f}}{p(\mathbf{y})}$$

If the observational model $p(\mathbf{y}|\mathbf{f})$ is Gaussian, the integrals over $\mathbf{f}$ can be computed analytically yielding,

$$p(f^*|\mathbf{y}) = \mathcal{N}(f^*|K_{f^*,\mathbf{f}}(K_{\mathbf{f},\mathbf{f}} + \sigma^2 I)^{-1}\mathbf{f}, K_{f^*,f^*} - K_{f^*,\mathbf{f}}(K_{\mathbf{f},\mathbf{f}} + \sigma^2 I)^{-1}K_{\mathbf{f},f^*})$$

In most modelling problems, however, the observational model is not Gaussian (e.g. binomial or multinomial classification, Poisson regression, and many other modelling problems) and the posterior predictive distribution $p(f^*|\mathbf{y})$ is analytically untractable and can not be obtained in a closed-form expression. Approximate inference have to be used to tackle the intractable inference, tipically based on sampling methods such as MCMC [Brooks et al., 2011] or analytical Gaussian approximations such as Laplace approximation ([Rasmussen and Williams, 2006, Williams and Barber, 1998], expectation propagation [Minka, 2001], or variational methods [Csató et al., 2000, Gibbs and MacKay, 2000]. Furthermore, Gaussian processes in a direct implementation scales cubically in the number of sample points, N, which effectively limits the modelling inference for medium and large datasets. In the introduction section in this paper several methods tipically used for dealing with the computational complexity has been commented.

In this paper we proposed an approximate model for Gaussian processes based on basis functions expansion which basically turns the GP prior model into a linear model. This linear GP model representation makes inference considerably faster and can be used as latent function in non-Gaussian observational models allowing modelling flexibility.

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 $\mathbf{x}-\mathbf{x}'$, such that it can be written $k(\mathbf{x},\mathbf{x}') = k(\mathbf{x}-\mathbf{x}')$, which means that the covariance is invariant to translations. Isotropic covariance functions are those that are function of the distance between observations, $k(\mathbf{x},\mathbf{x}') = k(||\mathbf{x}-\mathbf{x}'||) = k(\mathbf{r})$, which means that the covariacne is both translation and rotation invariant. The Matérn family with half-integers is the class of isotropic covariance functions probably most commonly used, which are

$$\begin{aligned}
k_{\nu=\infty}(\mathbf{r}) &= \sigma^2 \exp(-\frac{1}{2}\mathbf{r}/\ell^2), \\
k_{\nu=\frac{1}{2}}(\mathbf{r}) &= \sigma^2 exp(-\mathbf{r}/\ell), \\
k_{\nu=\frac{3}{2}}(\mathbf{r}) &= \sigma^2(1+\sqrt{3}\mathbf{r}/\ell \exp(-\sqrt{3}\mathbf{r}/\ell), \\
k_{\nu=\frac{5}{2}}(\mathbf{r}) &= \sigma^2(1+\sqrt{5}\mathbf{r}/\ell + \frac{5}{3}\mathbf{r}^2/\ell^2)\exp(-\sqrt{5}\mathbf{r}/\ell),
\end{aligned}$$

where ν is the order the kernel, and the ℓ and σ are the lengt-scale and magnitud, respectively, of the kernel. The particular case where $\nu = \infty$ is commonly known as squared exponential (exponentiated quadratic) covariance function, and that with $\nu = 1/2$ as exponential covariance function. These covariance functions can be easily generalized to the use of multidimensional length-scale $\boldsymbol{\ell}$, which 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 covariance functions written above are

$$\begin{aligned}
S_{\nu=\infty}(\mathbf{w}) &= \sigma^2\sqrt{2\pi}\cdot\ell\cdot\exp(-0.5\ell^2\mathbf{w}^2), \\
S_{\nu=\frac{1}{2}}(\mathbf{w}) &= 2\sigma^2\frac{1}{\ell}(\frac{1}{\ell^2}+\omega^2)^{-1}, \\
S_{\nu=\frac{3}{2}}(\mathbf{w}) &= 4\sigma^2\frac{\sqrt{3}}{\ell}^3(\frac{3}{\ell^2}+\omega^2)^{-2}, \\
S_{\nu=\frac{5}{2}}(\mathbf{w}) &= \frac{16}{3}\sigma^2\frac{\sqrt{3}}{\ell}^5(\frac{5}{\ell^2}+\omega^2)^{-3},
\end{aligned}$$

where variable $\mathbf{w}$ is in the frequency domain.

2.2. Hilbert space-based approximate Gaussian process model

The approximate Gaussian process method, developed by Solin and Särkkä [2018] and implemented in this paper, lay on the basis of 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 compact subsets $\Omega \subset \mathbb{R}^D$, and with some boundary condition.

First, we focus on the unidimensional case of the input space, such as $\Omega \in \{-L, L\} \subset \mathbb{R}$, where L is some positive real value.

The approximate method leads to the approximation of the covariance function between two input values $\{x, x'\} \in \{-L, L\}$ as

$$k(x, x') \approx \sum_j S_\theta(\sqrt{\lambda_j})\phi_j(x)\phi_j(x'), \tag{1}$$

where S_θ is the spectral density of the stationary covariance function k and θ the set of hyperparameters of k; a stationary covariance function can be equivalently represented in terms of the spectral density [Rasmussen and Williams, 2006], then the spectral density is a function of the hyperparameters θ. $\{\lambda_j\}_{j=1}^{\infty}$ 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):

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

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 eq. (2) are sinusoidal functions,

$$\begin{aligned} \lambda_j &= \left(\frac{j\pi}{2L}\right)^2, \\ \phi_j(x) &= \sqrt{\frac{1}{L}}\sin\left(\sqrt{\lambda_j}(x + L)\right). \end{aligned} \tag{3}$$

If we truncate the sum in (1) up to J ($j = 1, \cdots, J$), the approximate covariance function can be represented as

$$k(x, x') \approx \boldsymbol{\phi}(x)^\intercal \Delta \boldsymbol{\phi}(x'),$$

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

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

Thus, the Gram matrix K of the covariance function k for a collection of observations $i = 1, \cdots, N$ and collection of input values $\{x^i\}_{i=1}^N \in \mathbb{R}^N$ can be represented as follows

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

where $\Phi \in \mathbb{R}^{N \times J}$ is the matrix of eigenfunctions $\phi_j(x^i)$,

$$\Phi = \begin{bmatrix} \phi_1(x^1) & \cdots & \phi_J(x^1) \\ \vdots & \ddots & \vdots \\ \phi_1(x^N) & \cdots & \phi_J(x^N) \end{bmatrix}.$$

Now, the Gaussian process prior for the function f can be re-defined as

$$\mathbf{f} \sim \mathrm{GP}(0, \Phi \Delta \Phi^\intercal).$$

This equivalently leads to a linear representation of the function f,

$$f(x) \approx \sum_j \left(S_\theta(\sqrt{\lambda_j}) \right)^{1/2} \phi_j(x) \beta_j,$$

where $\beta_j \sim \mathrm{N}(0, 1)$ is a random variable with zero mean and unitary variance. That is, the function f can be approximated with a finite basis function expansion (eigenfunctions ϕ_j of the Laplace operator), scaled by the squared root of the spectral density as a function of the eigenvalues λ_j. The eigenfunctions ϕ_j do not depend on the kernel hyperparameters θ, the only dependence on θ is through the spectral density S_θ. The eigenvalues λ_j are monotonically increasing with j and the spectral density goes rapidly to zero for bounded covariance functions. Therefore, it is expected a good approximation in eq. (6) for a finite number, J, of terms in the series as long as the inputs values x^i are not near the boundaries of the domain $[-L, L]$, where the Laplacian was taken to be zero.

The computational cost of this approximate model scales $O(n \cdot J + J)$, where n is the number of observations and J the number of basis functions.

2.3. Generalization to multidimensional input space

We are going to generalize the results from the previous section 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 corresponds to the total number of D-tuples over J, which is $\prod_{d=1}^{D} J_d$, where J_d is the number of basis function for the dimension d. Let $\mathbb{S} \in \mathbb{R}^{\prod_{d=1}^{D} J_d \times D}$ be the set of D-tuples elements. Each new eigenfunction ϕ_j^* correspond to the product of the univariate eigenfunctions whose indices corresponds to the the elements of the D-tuple $\mathbb{S}_{j\cdot}$. And each new eigenvalue λ_j^* is a D-vector whose elements are the univariate eigenvalues whose indices correspond to the elements of the D-tuple

$\mathbb{S}_j$.. Let $j = \{1, \cdots, J^D\}$,

$$\begin{aligned} \phi_j^*(\mathbf{x}) &= \prod_{d=1}^{D} \phi_{\mathbb{S}_{jd}}(\mathbf{x}_d) = \prod_{d=1}^{D} \sqrt{\frac{1}{L_d}} \sin\left(\sqrt{\lambda_{\mathbb{S}_{jd}}}(\mathbf{x}_d + L_d)\right) \\ \lambda_j^* &= \{\lambda_{\mathbb{S}_{jd}}\}_{d=1}^{D} = \{\left(\frac{\pi \mathbb{S}_{jd}}{2L_d}\right)^2\}_{d=1}^{D} \end{aligned} \tag{4}$$

As an example we show the matrix $\mathbb{S}$ for a *three*-dimensional input vector $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ ($D = 3$) and $J_1 = 2$, $J_2 = 2$ and $J_3 = 3$ eigenfunctions and eigenvalues for the first, second and third dimension, 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

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

The approximate covariance function is represented as

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{J^D} S_\theta^* \left(\sqrt{\lambda_j^*}\right) \phi_j^*(\mathbf{x}) \phi_j^*(\mathbf{x}'), \tag{5}$$

where S_θ^* is the spectral density of the D-dimensional covariance function. The D-dimensional spectral density functions associated with the Matérn covariance functions considered in the previous section are

$$\begin{aligned} S_{\nu=\infty}^*(\mathbf{w}) &= \sigma^2 \sqrt{2\pi}^D \prod_{d=1}^{D} l_d \cdot \exp\left(-0.5 \sum_{d=1}^{D} l_d^2 \mathbf{w}_d^2\right), \\ S_\nu^*(\mathbf{w}) &= \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu) l^{2\nu}} \left(\frac{2\nu}{l^2} + 4\pi^2 \mathbf{w}^2\right)^{(\nu + D/2)}, \end{aligned}$$

where the D-vector $\mathbf{w}$ is in the frequency domain. We can represent the approximate series expansion of the function f in the multidimensional input space as,

$$f(\mathbf{x}) \approx \sum_j \left(S_\theta^*(\sqrt{\boldsymbol{\lambda}_{\boldsymbol{j}}^*})\right)^{1/2} \phi_j^*(\mathbf{x}) \beta_j, \tag{6}$$

where $\beta_j \sim \mathrm{N}(0, 1)$ is a random variable with zero mean and unitary variance. The

computational cost in a multidimensional setting scales $O((n+1)\cdot\prod_{d=1}^{D} J_d)$, where n is the number of observations and J_d is the number of basis functions considered in the dimension d.

2.4. Learning hyperparameters and model inference

• It has an attractive computational cost as this basically turns the regular GP model into a lineal 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 differentation 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 covariane 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 mod-

els the lengthscale hyperparameter tend to be fixed and then the fit is covered by the magnitud parameter. In that sense, Splines models tend to loose the useful interpretation of the lengthscale parameter.

• In addition, the computation of the automatic differentation 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 differentation 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 covariane 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

Implementation and performance of the approximate GP model depends on the number of basis functions and box size settings, which are also related each other. The time of computation in the approximation mainly depends on the number of basis functions. At the same time, the appropriate values for the number of basis functions and box size depend on the non-linear functional effects of the function to be learned. In GP functions these non-linear functional effects are characterized by the lengthscale of the covariance function.

In this section we analyze the rule and relationships of the number of basis functions and the box size on approximation and their effects on the learned functions.

We also analyze these two factors in relation to the lengthscale of the functions to be learned. We provide a useful graph that retains how these factors, number of basis fucntions, box size and lengthscale, relate each other in relation to the performance of the model. This graph basically provides the recommended values for these factors in order to get a close approximation. This graph also allows us to diagnose an actual fit.

In this section we also make a comparison between regular GP model and approximate GP model fitted for different dataset with different "wigglyness". The comparison is made through comparing their estimated lengthscales.

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

The approximation of the covariance function is a series expansion of eigenfunctions and eigenvalues of the Laplace operator in a given domain Ω, e.g. in a 1-D input space $x \in \Omega = [-L, L]$:

$$k(\tau) \approx \sum_j S_\theta(\sqrt{\lambda_j})\phi_j(\tau)\phi_j(0),$$

where j is the index for the eigenfunctions and eigenvalues, and $\tau = x - x'$. The

eigenvalues λ_j and eigenfunctions ϕ_j are those in equation (3) for the unidimensional case and those in equation (4) for the multidimensional case.

The approximation to the covariance function will become exact in the limit when the number of basis functions approach infinity, $j = 1, \cdots, \infty$. The number of basis functions can be truncated at some finite positive value J such that the difference between the densities of the exact and approximate covariance functions be less than a threshold ϵ,

$$\int k(\tau)d(\tau) - \int \sum_{j}^{J} S_\theta(\sqrt{\lambda_j})\phi_j(\tau)\phi_j(0)d(\tau) < \epsilon. \tag{7}$$

The finite number J of basis functions in the approximation needed to satisfy eq. (7) will depend on the non-linear effects of the function to be learned, that is, on its lengthscale ℓ. The approximation will also depends on the box size L (equations (3) and (4)) which will affect the performance specially near the boundaries. The box size will also have an effect on the number of basis functions needed in the approximation. Note that we refer to the box size as the value L of the Dirichelt boundary condition in eq. 2. The box size L is set as an extension of the desired predicting input domain Ψ. Considering a zero-mean input domain $\Psi = [-S, S] \in \mathbb{R}$, the box size is defined as,

$$L = c \cdot S \tag{8}$$

where S represents the half-range of the input space, and c ($c \geq 1$) is the proportional extension factor. From now on we are going to refer to this proportional factor c as the boundary factor of the approximation. The boundary factor can also be regarded as the box size L normalized by the half-range of the input space S.

For a naive illustration of the effects of these two factors, number of basis functions J and box size L, on the performance of the approximation, a set of noisy observations are simulated from a GP model with lengthscale $\ell = 0.3$ and magnitud $\alpha = 1$ for input vales within the input domain with half-range $S = 1$. Approximate GP models with different number of basis functions and box sizes L ($L = c \cdot S$, with $c \geq 1$) are fitted to this data. The lengthscale and magnitud parameters are considered fix to the true values of the generative GP model. Figures 1 and 2 ilustrate the individual effects of the number of basis functions and the box size, respectively, on the approximation of the covariance function. On the one hand, for a fixed L, let's suppose it is set to a right choice, Figure 1 shows the effect of changing the number of basis functions in approaching the posterior mean and the covariance function. It can be seen how the number of basis functions affects the non-linear effects of the posteriors. Fewer basis functions implies smoother covariance functions and consequently smoother posteriors. The more "wigglyness" of the function, the more basis functions are needed. On the other hand, and similarly, for a fixed number of basis functions J, let's suppose it is set to a right choice, Figure 2 shows the effect of changing the box size L in approaching the posterior mean and the covariance function. It can be seen how the box size mainly affects the approximation near the boundaries of the function and affects the covariances at long distances.

In the previous Figures 1 and 2 we focused on illustrating the individual effects of the number of basis functions and the box size. Furthermore, for a clearer illustration

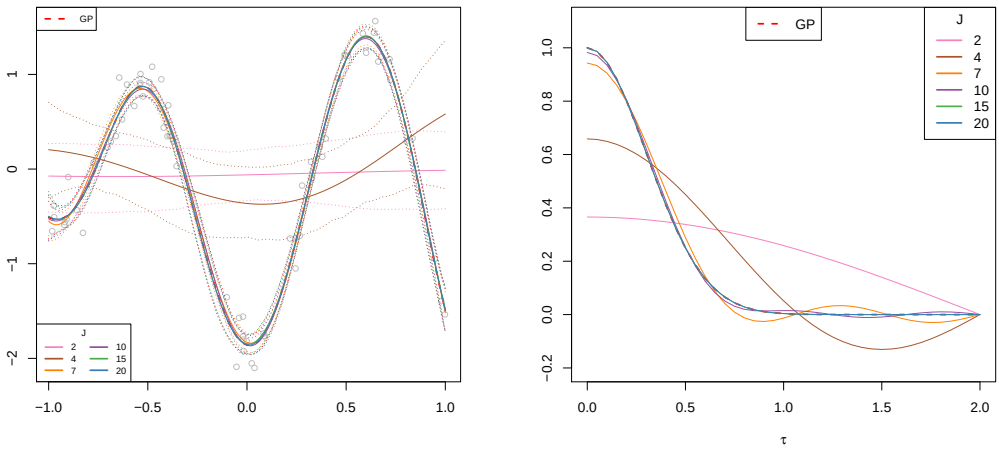

Figure 1. (left) Posterior distributions for different number of basis functions J. (right) Covariance function for different number of basis functions J.

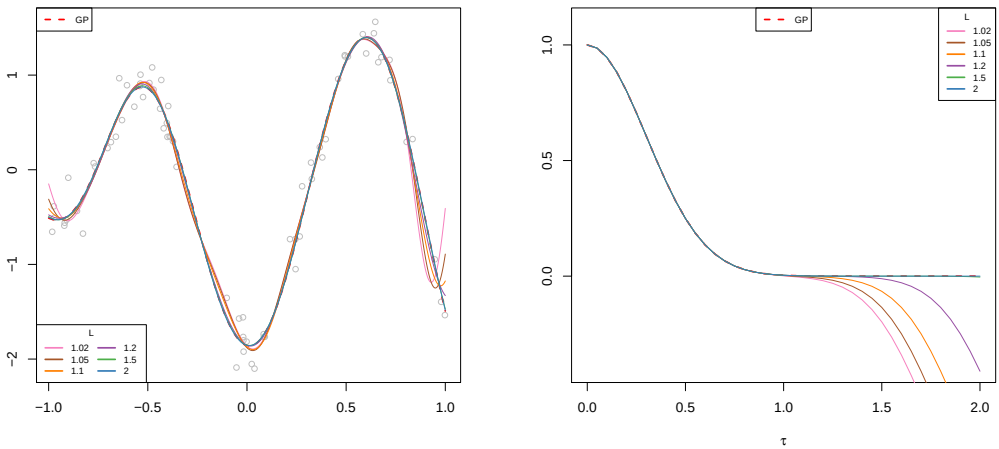

Figure 2. (left) Posterior means for different different values of the box size L. (right) Covariance functions for different values of the box size L.

of these effects, the lengthscale and magnitud parameters in the approximate GP model were fixed to their true values. Now we are going to focus on analyzing the interaction effects between these two factors, number of basis functions and the boundary factor, on the performance of the approximation. The lengthscale and magnitud parameters will not be fixed rather they will be estimated in both regular and approximate GP models. Figure 3 shows the posteriors means and the covariance functions obtained after fitting the data, for different number of basis functions and boundary factors. Figure 4 shows the root mean square error (RMSE) of the approximate GP model over the regular GP model as a function of the number of basis functions and the boundary factors. Figure 5 shows the lengthscale and magnitud parameters for the regular GP model and for the approximate models as a function of the number of basis functions and boundary factors. Looking at the RMSEs in Figure 4, it can be drawn that the optimal choice in terms of precision and computations would be 15 basis functions and a boundary factor between 1.5 and 2.5. The choice of 10 basis functions and a boundary factor of 1.5 could be an accurate enough choice. This same conclusion can be also be intuitively seen in the posterior means and covariance functions plots in Figure 3.

From Figure 4, a behaviour in terms of performance of the approximate GP model as a function of the number of basis functions and the boundary factor, can be deduced:
• as the boundary factor increases, more basis functions are needed,
• as fewer basis functions are used, the boundary factor must decrease.

Similar conclusions can be stated from results of Figure 5, This Figure shows the estimated lengthscale and magnitud parameter as function of the number of basis functions and the boundary factor. In addition to the above conclusions, it can be easily seen that exists a minimum value for the boundary factor under which a close approximation will never be achieved.

Additionally, there exit a relation of the number of basis functions J and the boundary factor c with the lengthscale ℓ of the function. Figure 6 collects how these three factors relate each other in relation to the performance of the approximation evaluated using eq. (7). On the X-axis of this plot is placed the lengthscale of the process normalized by the half-range of the data ℓ/S. The countour lines gather the boundary factor c. And the Y-axis represents the number of basis functions. Thus, this figure give us, for a certain GP function with lengthscale ℓ and given a fixed boundary factor c, the minimum number of basis functions needed to achieve a close approximation in terms of satisfying eq. (7). Alternatively, this figure could also be read as the minimum boundary factor c that we should use given a number of basis functions, for certain GP function with certain lengthscale. And also this figure could also be read as the minimum functional lengthscales that can be closely approximated given a number of basis functions and a boundary factor. This figure basically collects the behaviour of the approximate GP model as a function of the lengthscale, number of basis functions and boundary factor:

• As the lengthscale of the function increases, the boundary factor and the number of basis functions needed for the approximation decrease.

• There is a minimum boundary factor under which a close approximation never is going to be achieved (The contour lines have an end in function of the lengthscale). This statement match with one of the behaviour recognized in Figure 5 where were a

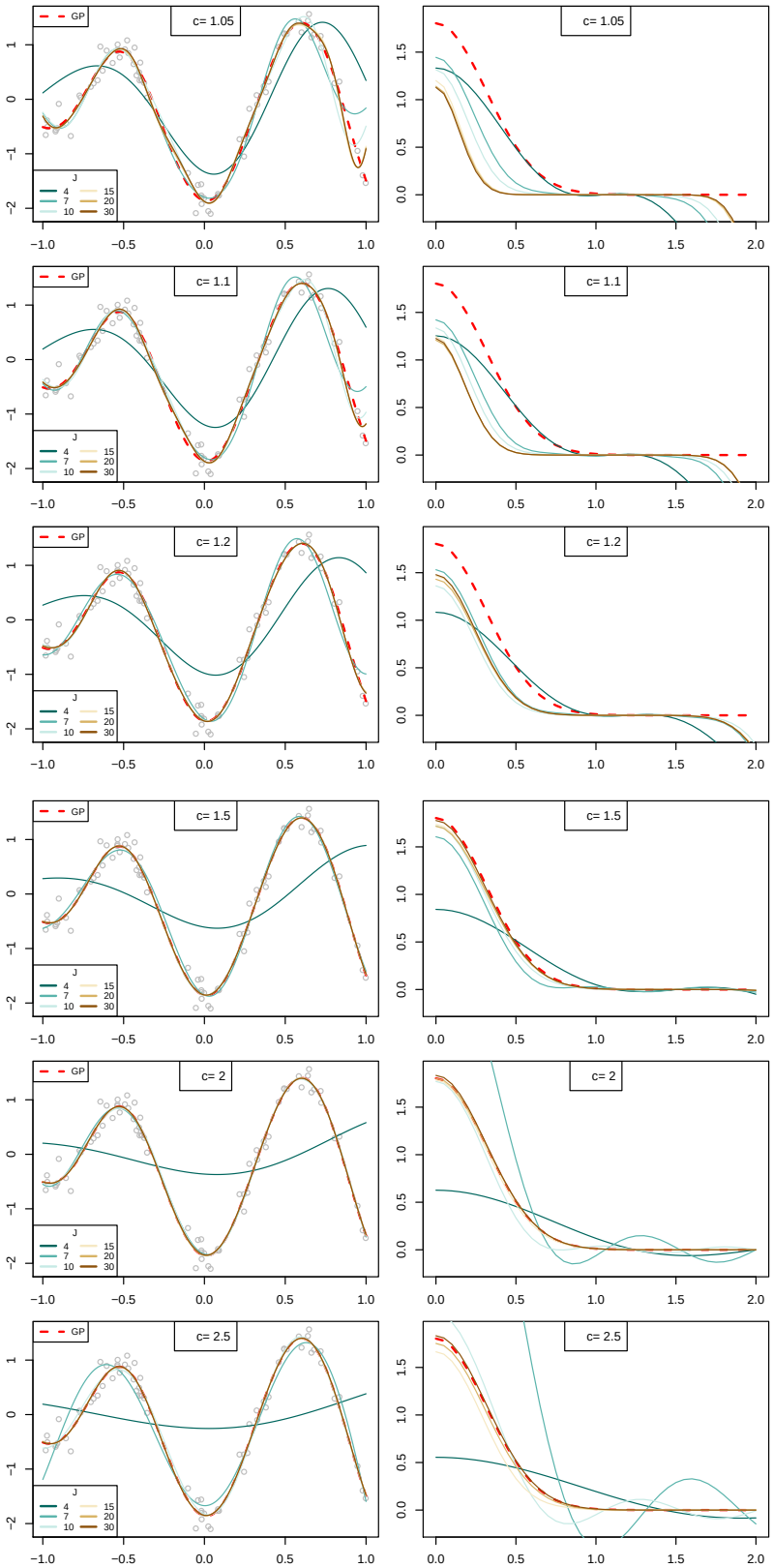

Figure 3. Posterior distributions and covariance functions of the proposed approximate GP models and the exact GP model in function of the number of basis functions J and the boundary factor c.

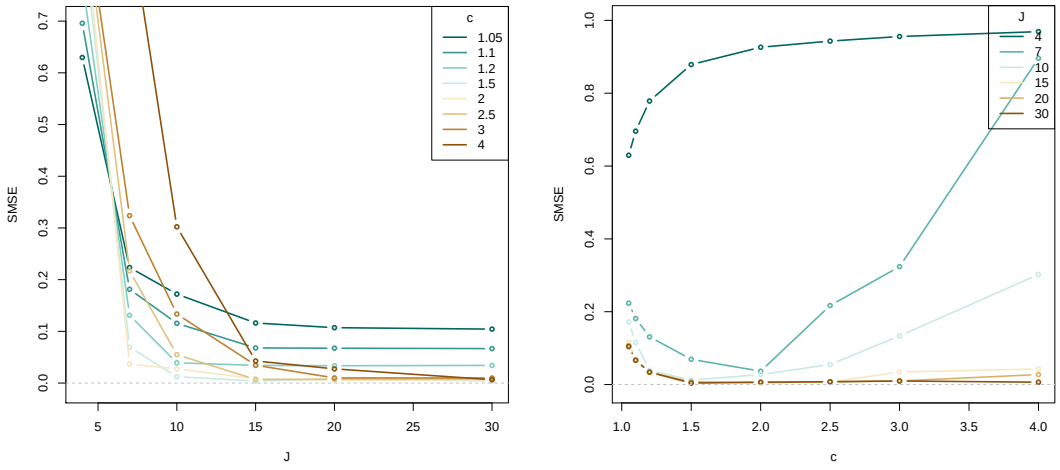

Figure 4. Standarized root mean square error (SRMSE) of the proposed approximate GP models and the exact GP model. (left) SRMSE against the number of basis functions J for different values of the boundary factor c. (right) SRMSE against the boundary factor c for different values of the number of basis functions J.

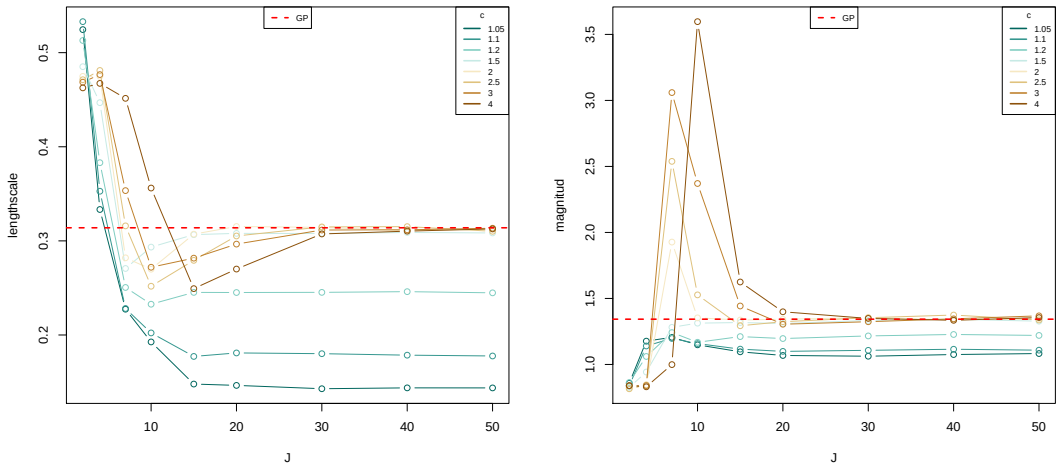

Figure 5. (left) Estimated lengthscales against the number of basis functions J for different values of the boundary factor c. (right) Estimated GP magnitud the number of basis functions J for different values of the boundary factor c.

minimum value for the boundary factor under which a close approximation will never be achieved.

• The lower the boundary factor, the fewer basis functions and the smaller lengthscales that are suitable to be fitted.

• This plot serves as a diagnosis tool in the sense that if a estimated lengthscale is lower than that minimun that the figure bring us, given certain number of basis functions and certain boundary factor, means that the number of basis functions should be increased or the boundary factor decreased. On the other hand, if the lengthscale is bigger than that generated by the figure means that the approximation should be close enough.

Figure 6 is useful for the user to know the suitable values for the number of basis functions J and the boundary factor c to approximate certain process characterized by its lengthscale. This Figure will allow us to optimize computations which depends on the number of basis functions. Starting from some guess of the lengthscale of the process and knowing the range of the input data which we are interested in, the user can do a few iterative adjustments to obtain an optimal fit. This figure also provides a diagnosis tool of the fit by comparing a actual estimated lengthscale to that the figure bring.

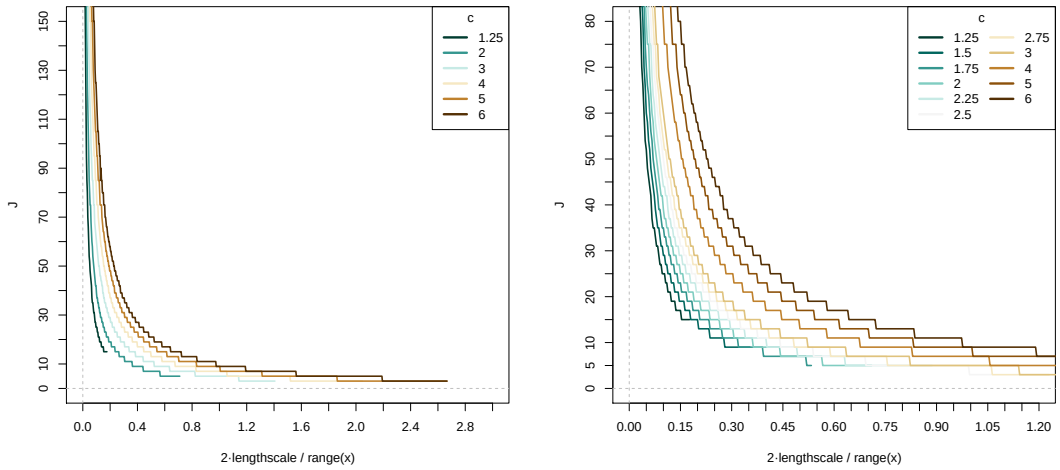

Figure 6. Relation between the minumum number of basis functions J, the lengthscale normalized by the half-range of the data ($\frac{\ell}{S}$), and the boundary condition factor c ($c = \frac{L}{S}$).

If we look back to the conclusions drawn from Figures 4 and 5 where 10 basis functions and a boundary factor of 1.5 would be enough to closely approximate a function with normalized lengthscale eaual to 0.3, we can recognize that these conclusions also matches the conclusion brought form Figure 6.

3.2. Comparative analysis between the lengthscales estimated by a regular GP model and approximate GP model

Finally, we make a comparative analysis between the lengthscales estimated by a regular GP model and approximate GP model in several datasets with different "wiggly-

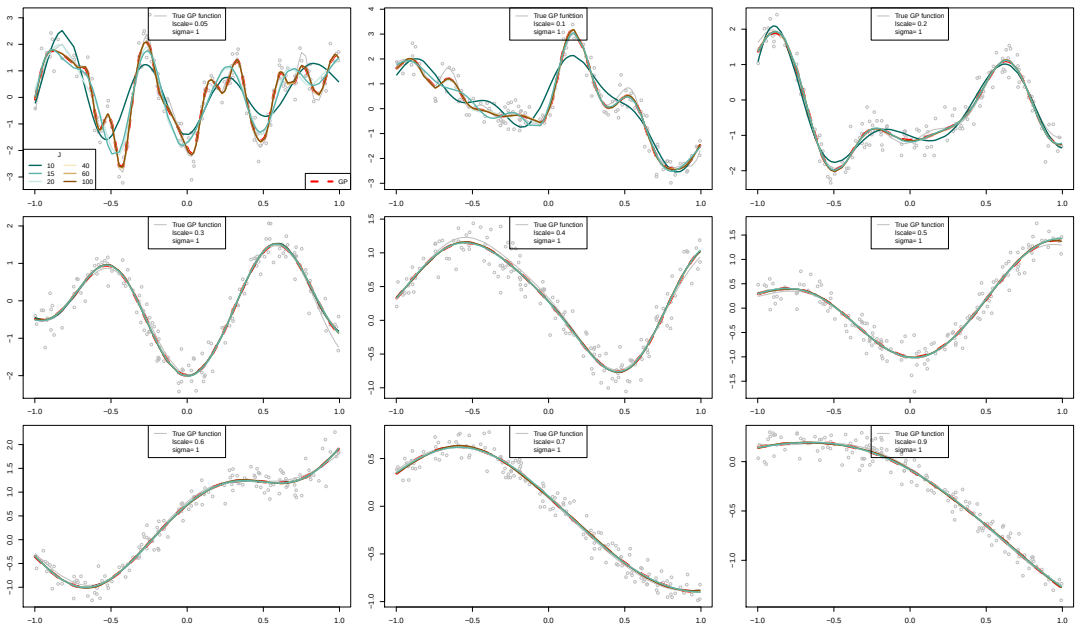

Figure 7. Posterior means of the exact and approximate GP models fitted over different datasets with different lengthscales.

ness" (or lengthscales). From this analysis we can recognize when an approximation is not close enough to the regular GP when the lengthscales are not similar. We contrast these results from these lengthscale similarities between the regular and the approximate GP model with the information brought from the Figure 6.

We fit the regular GP model and the approximate GP model on different datasets. These datasets are drawn from GP prior models with a squared exponential covariance function with different lengthscales for every dataset. The approximate GP models is fitted using different number of basis functions. The boubdary condition factor is set en ivery fit to a right and optinal choice. Figure 7 shows the posterior means of the exact GP model and the approximate GP model with different number of basis functions for every dataset with different wiggly effects (different lengthscale).

The comparative analysis between the lengthscales estimated by a regular GP model and approximate GP model in several datasets is shown in Figure 8. The X-axes of the plots represent the true lengthscales of the generative functions for every dataset scenario, and in the Y-axes are represented the estimated lengthscales for the regular GP and the approximate GP models. Different comparison are made using different number of basis functions in the approximations. Let's assume that the box size parameter was properly set in every case. The dashed black line represents the minimum lengthscales that can be closely approximate, given the number of basis functions and the boundary factor, according to Figure 6. Figure 9 shows the SRMSE of the approximate GP model over the regular GP model for the different datasets with different lengthscale scenarios and the number of basis functions used in the approximation.

From Figure 8 we can recognize similarities and dissimilarities between the estimation for the lengthscales of the regular GP model and the approximations, which indicates the performance of the model. Figure 9 with the SRMSE also indicates the performance of the model. We can appreciate an agreement between Figures 8 and 9 except for very small lengthscales and very long lengthscales. Figure 8 shows a corre-

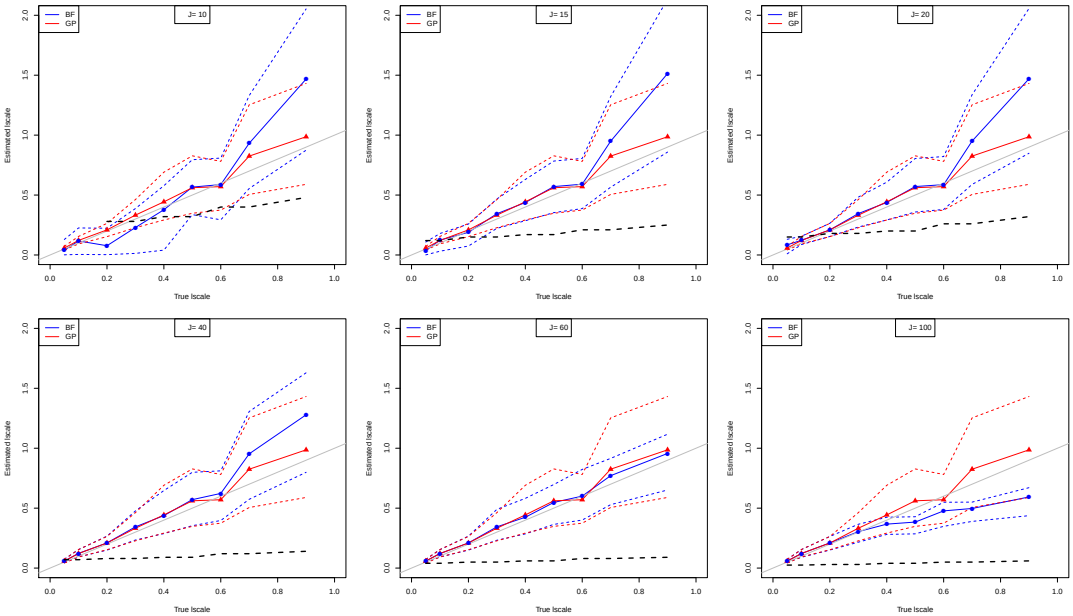

Figure 8. True lengthscales versus estimated lscales of the different realizations of datasets using different number of basis functions.

pondence between very low lengthscales when the SRMSE is large, conversely there is no correspondence between very large lengthscale and the SRMSE is close to zero. These effects can make sense because, in the case of long lengthscales these have larger variabilities obtaining similar posteriors, and in the case of low lengthscale these have lower variabilities obtaining diferent posteriors.

The dashed black line represents the minimum lengthscales that can be properly fitted by the approximate GP model according to Figure 6.

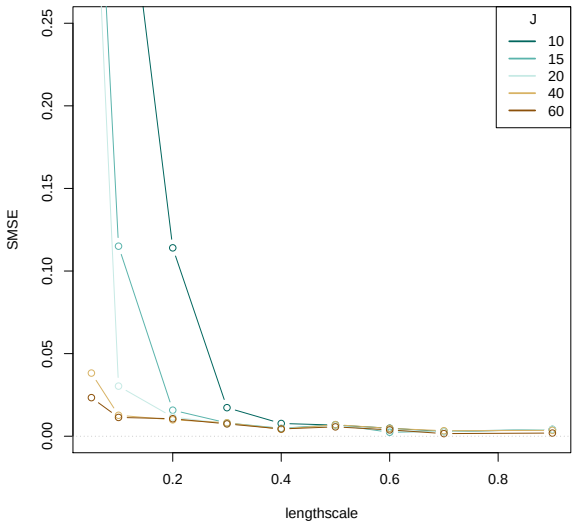

Figure 9. SMSE of the approximate GP model for the different datasets with different wiggly effects and the number of basis functions.

4. Univariate examples

4.1. Simulated data

A simulated one-dimensional dataset, with 250 observations $f(\mathbf{x}_i)$ $(i = 1, \cdots, 250)$ drawn from a Gaussian process prior with a Matern covariance function with 3/2 degrees of freedom, with inputs $\mathbf{x} \in \{-1, 1\} \subset \mathbb{R}$. The hyperparameters of this GP prior with a Matern kernel, marginal variance α and lengthscale ℓ, are set to 1 and 0.15, respectively. The additive Gaussian noise σ to these drawn points is set to 0.2. The latent Gaussian process model is as follows,

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

where $\mathbf{I}$ is the identity matrix and k the Matern covariance function in eq. (4). More efficient when using HMC than this latent GP model would be the marginalized model,

$$\mathbf{y} \sim \mathcal{N}(0, K + \sigma^2 \mathbf{I}) \tag{9}$$

where the element K_{ij} is the covariance function $k(\mathbf{x}_i, \mathbf{x}_j, \theta)$ evaluated at the inputs $\mathbf{x}_i$ and $\mathbf{x}_j$.

In the approximate GP model, the covariance function k is approximated as in eq. (1) with the spectral density in eq. (8). The latent function $f(\mathbf{x})$ is approximated as in eq. (6).

We compare the modelling performance over this simulated process of our proposed approach with the performance of a regular GP following the marginalized version in eq. (9) and a Splines-based model. For our basis function approach we use $J = 80$ basis functions and a boundary factor $c = 1.2$. For the Splines-based model we use 80 knots or basis in the model.

Figure **??** shows the posteriors distributions of the proposed model and the regular GP model, fitted over the data. Posteriors distributions of the proposed approximate GP model, of the exact GP model, and the Splines model are plotted jointly with the true GP prior from the data were drawn. The right and left extremes of the data, as well as three small regions around the middle of the data, correspond to out-of-sample or test data which have not been taking part of training the models.

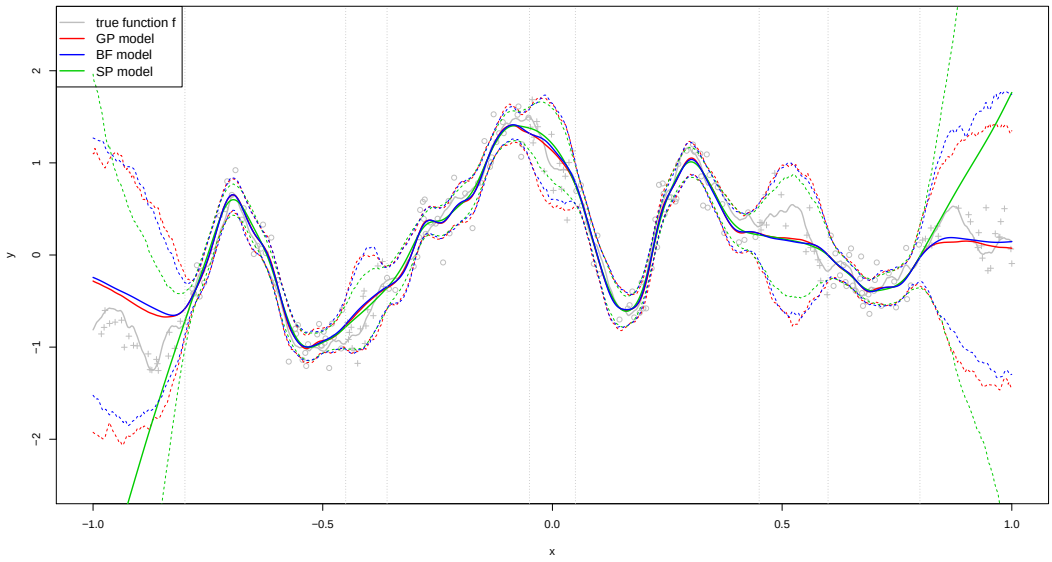

Figure 10. Posterior distributions of the proposed approximated GP model, the exact GP model, and the Splines model.

Figure 11 shows the standardized mean squared error (SMSE) over the true function for interpolation and extrapolation data. As it can be seen the splines model does not extrapolate data properly. For interpolation both models show similar performance although the approximate GP model seems to be a bit better.

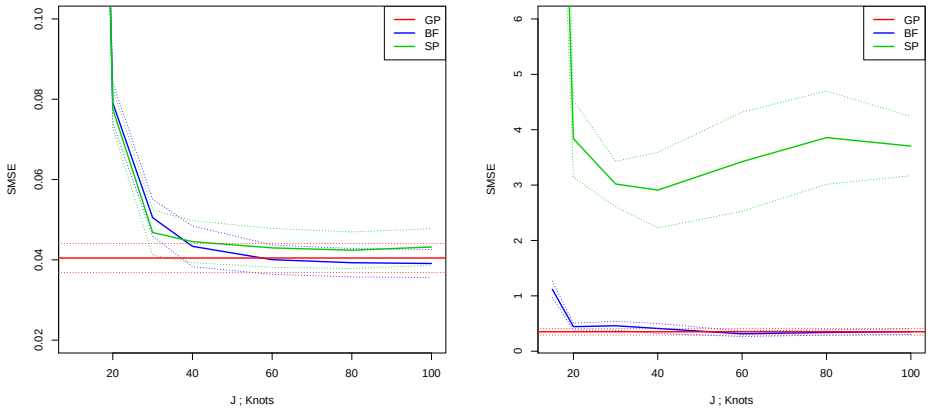

Figure 11. Standarized mean square error (SMSE) of the different methods againts the true function. (left) SMSE for interpolation. (right) SMSE for extrapolation

4.2. Gay data

This data set relates the proportion of support for same-sex marriage to the age. This proportions correspond to binomial observations per age group i. The data consist of $i = 1, \cdots, 74$ binomial observations of the amount of people y_i supporting same-sex marriage from a population n_i per age group.

The observational model for the observed number of people y_i supporting same-age marriage per age group i is a binomial model with the total number of people per age group n_i and the probability per age group p_i as model parameters. The total number of people per age group n_i is a known quantity and the goal is to estimate the same-sex support probability p_i per age group i or mean number of support people per age group $\bar{y} = p_i \cdot n_i$.

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

The probability p_i is related through the logit function to a latent Gaussian process prior distribution with the age values x_i as inputs,

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

with a squared exponetial covariance function k in eq. (2).

In the approximate GP model the covariance function is approximated as in eq. (1) with spectral density in eq. (6), and the latent function $f(\mathbf{x})$ as in eq. (6). As comparison a splines model is also fitted. Figure 12 shows the posterior distributions of the three models. In the case of the approximate GP model the posterior shown in the figure corresponds to 20 basis functions and a boundary factor of 1.5. For the spline posterior shown in the figure, it correspond to a 20 knots. Two small sets of observations have been used as test data.

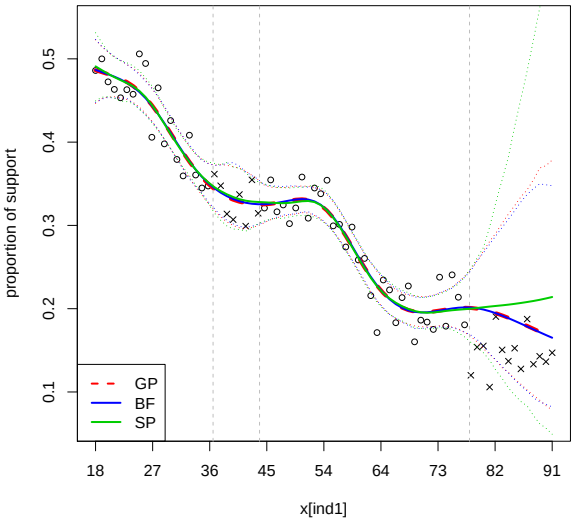

Figure 12. Posterior distributions of the proposed approximated GP model, the exact GP model, and the Splines model.

Figure 13 shows the MSE against the regular GP model of the approximate GP model for the different number of basis functions and boundary factor. The expected patterns of the approximation in function of the number of basis function and boundary factor are recognized.

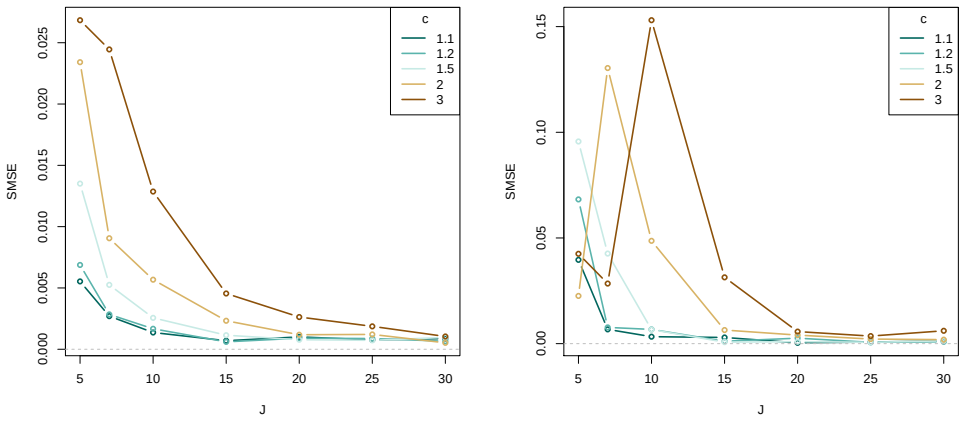

Figure 13. Mean square error (MSE) of the different methods againts the true function. (left) MSE for training data. (right) MSE for test data.

Figure 14 shows the MSE of the different models against the actual data, for training set of data and the test set of data.

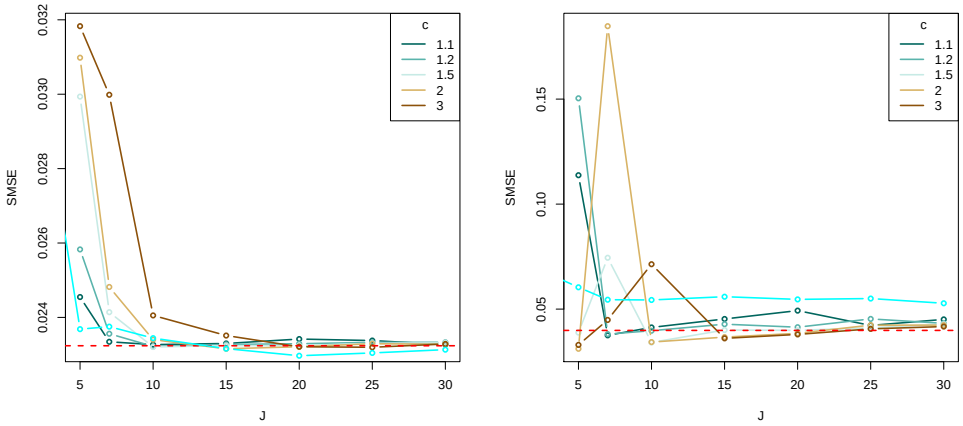

Figure 14. Mean square error (MSE) of the different methods againts the true function. (left) MSE for training data. (right) MSE for test data.

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 modications 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 degenarete prior which produces unresoanble 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´om 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, QuiÃ±onero-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 Matern 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 differentation 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 covariane 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 magnitud 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 lengthsccale of the functions.

• We make recomendations 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 $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$ can be represented as the Fourier transform of a positive finite measure ($Bochner's theorem$).

(Bochners 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

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

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:

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

Appendix E. Approximate the covariance function using Hilbert space methods

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

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

Assuming that the espectral density function $S(\cdot)$ is regular enough, then it can be

represented as a polynomial expansion:

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

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

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

then the covariance function can be represented as

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

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):

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

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 $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2\}$ ($D = 2$) 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

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

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

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.

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.

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.

QuiÃ±onero-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.

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.

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.

Sarkka, 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.

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* .

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.