Hilbert space methods to approximate Gaussian processes using Stan

ARTICLE HISTORY
Compiled February 19, 2019

ABSTRACT

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

Contents

1. Introduction

1.1. Motivation

Efficient computation in Gaussian process (GP) models is of broad interest across machine learning and statistics, with applications 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], Bayesian optimization and probabilistic numerics, [Briol et al., 2015, Hennig et al., 2015, Roberts, 2010] , and many others.

One of the main limitations of Gaussian processes (GPs) [Rasmussen and Williams, 2006] is that computational demands and memory requirements scale as $O(n^3)$ and $O(n^2)$ respectively, effectively limiting a direct implementation. This limits the applicability of GPs to a number of training samples n at most a few thousand cases. The computational requirements arise because in solving the GP regression problem we need to invert the $n \times n$ Gram matrix $\mathbf{K}$ of the covariance function $k(\mathbf{x}, \mathbf{x}')$, which is an $O(n^3)$ operation in general. This problem becomes especially important when using sampling methods for full Bayesian inference, when in each sampling step we need to invert the Gram matrix, usually through Cholesky factorization. Then, approximate methods that will alleviate these computational demands are needed. Sparse GPs, which are approximate methods to fit GP models, do reduce the dimension of the posterior, but it still requires Cholesky factorizations. They are basically based on reducing the amount of data used to fit the model, and convergence to the full GP model is not straight forward. Furthermore, reducing the amount of data while maintaining the posterior densities is not a simple procedure.

We want to make full Bayesian inference over Gaussian process models accessible to a broader audience by making them fast and easy to apply. If we can find a method to decompose the GP kernel into basis functions, we can represent the GP as a linear model, which makes it faster to fit. The set of Laplace eigenfunctions considered in this paper forms such a basis and we can compute them analytically. Furthermore, they are independent of the choice of kernel (including the hyperparameters), which means that we do not need to autodiff through the kernel and the Cholesky. We are going to achieve this by making it possible to use GPs in a probabilistic programming language like Stan.

Well-known splines models are a sort of basis functions expansion models 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.

1.2. 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.

1.3. 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.

1.4. 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.

1.4.1. 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].

1.4.2. 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 specic 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.

2. Gaussian process prior model

Gaussian processes (GP) are natural and flexible non-parametric models to define probability distributions over multi-dimensional functions [**?**]. This enables to apply probabilistic inference directly to functions. As a continuous function is a uncontable infite collection of values, a GP model can be seen as a probability model over an infinite collection of random varibales. However, formally, the defining feature of a GP model is that the joint distribution of function values at any finite number of input values is a multivariate normal distribution.

...

...

...

A stationary covariance function is a function of $x - x'$, so we can write $k(x, x') = k(x - x')$. Isotropic covariance functions are those that are function of the distance between observations, $k(x, x') = k(|x - x'|) = k(r)$. The Matérn family with half-integers is the class of isotropic covariance functions probably most commonly used, which are

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

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 multidimensional case considering multidimensional inputs $\mathbf{r}$ and the multidimensional length-scale $\boldsymbol{\ell}$.

The spectral density functions associated with the univariate Matérn covariance functions written above are

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

where variable w is in the frequency domain.

3. Hilbert space-based approximate Gaussian process prior model

3.1. Method

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):

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

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,

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

If we truncate the sum in (5) 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. (4) 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.

3.2. 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. We assume the same number J of basis functions terms in the series expansion for every input dimension, however it is straight forward to generalize this to different number of term series J_d for every dimension.

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 J^D. Let $\mathbb{S} \in \mathrm{I\!R}^{J^D \times D}$ 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 $\mathbb{S}_{j\cdot}$. Let $j = \{1, \cdots, J^D\}$,

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

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 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

$\mathbb{S} \in \mathrm{I\!R}^{12\times 3}$ 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

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{J^D} S_\theta^* \left(\sqrt{\lambda_j^*}\right) \phi_j^*(\mathbf{x})\phi_j^*(\mathbf{x}'), \tag{3}$$

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,

$$f(\mathbf{x}) \approx \sum_j \left(S_\theta^*(\sqrt{\boldsymbol{\lambda_j^*}})\right)^{1/2} \phi_j^*(\mathbf{x})\beta_j, \tag{4}$$

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.

3.3. Details in computational complexity

• 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.

• Then, the operations in this whole linear model will scale $O(nm)$ and computed in every step of the HMC sampling method.

• 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.

3.4. 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

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

The approximation to the covariance function will become exact in the limit when the number of basis functions approach infinity $J = 1, \ldots, j, \ldots, \infty$. The number of basis functions can be truncated that the difference between the covariance function and the approximation be less than a threshold ϵ,

$$\int k(\tau)d(\tau) - \int \sum_j S_\theta(\sqrt{\lambda_j})\phi_j(\tau)\phi_j(0)d(\tau) < \epsilon. \tag{5}$$

The finite number of basis functions in the approximation to satisfy eq. (5) will depend on the smoothness or roughness of the function, that is, on the lengthscale ℓ. The approximation will also depends on the box size L which will affect the performance specially near the boundaries.

To illustrate the effects of these two factors, number of basis functions J and box size L, on the performance of the approximation, let's consider a simulated dataset of 75 noisy observations from a GP prior with lengthscale $\ell = 0.3$ and magnitud $\alpha = 1$. The additive noise added to the GP prior simulations is $\sigma = 0.2$. A regular GP model and different approximate GP models with different choices for the number of basis function J and box size L are fitted on this dataset.

For a fixed L, let's suppose it is set to a right choice, Figure **??** shows the effect of the number of basis functions in approaching the posterior mean and the covariance function. The number of basis functions affects the non-linearity or roughness (or smoothness) of the posteriors. Fewer basis functions implies smoother covariance functions and consequently smoother posteriors. The more roughness the process is, the more basis functions are needed.

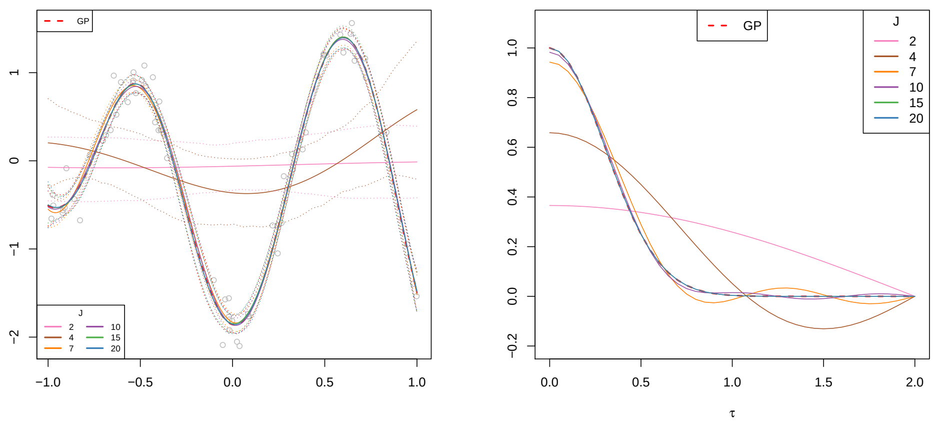

Figure 1. (left) Posterior distributions for different number of basis functions J. (right) Covariance function for different number of basis functions J.

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 the box size L in approaching the posterior mean and the covariance function. The box size mainly affects the approximation near the boundaries of the function, and affects the covariances at long distances.

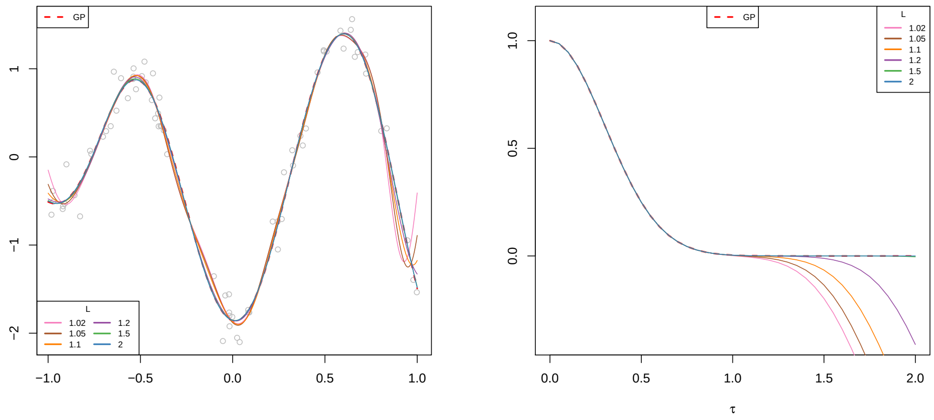

Figure 2. (left) Posterior means for different different values of the box size L. (right) Covariance functions for different values of the box size L.

Therefore, the approximation to the covariance function depends on the number of basis functions J used in the series expansion and the value for boundary condition or box size L. Additionally, there exit a relation among the number of basis functions, the box size and the lengthscale or roughness of the process on the performance of the approximation. Figure 3 collects how these three factors relate each other in order to achieve a good approximation to the covariance function of the regular GP using eq. (5). On the X-axis of this plot is placed the lengthscale of the process normalized by the half-range of the data. The countour lines gather the box size which is also normalized by the half-range of the data, to which we refer as the boundary factor c, where $L = c \cdot \text{range}(x)/2$. And the Y-axis gathers the number of basis functions. Thus, this figure give us the minimum number of basis functions needed to achieve a good approximation in terms of satisfying eq. (5), for a certain GP process with lengthscale ℓ and given a fixed boundary factor c. Alternatively, this figure could also be read as the maximum boundary factor c that we should use given a number of basis functions, for certain GP process with certain lengthscale. And also this figure could also be read as the maximum roughness in a process that can be fitted properly given a number of basis functions and a box size. Finally, this figure also gives us the minimum value for the boundary factor c which is suitable for certain roughness or number of basis functions. This figure basically collects the behaviour of the approximate GP model in function of the lengthscale, number of basis functions and boundary condition:

- As the lengthscale or smoothness of the process increase, the boundary factor and the number of basis functions needed for the approach decrease.
- There is a minimum boundary factor to achieve a good approach in function of the lengthscale (The contour lines have an end in function of the lengthscale).
- The lower boundary factor, the fewer basis functions, and the lower lengthscales suitable to be fitted.
- This plot serves as a diagnosis tool in the sense that if the estimated lengthscale is lower than that minimun generated by the figure for the chosen number of basis functions and boundary factor means that we would need to increase the number of basis functions or decrease the boundary factor.
- On the other hand, if the lengthscale is bigger than that generated by the figure then it means that the fit should be good.

Figure 3 is useful for the user to know the suitable values for the number of basis functions J and the boundary factor c to approach a 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, in the sense that if the estimated lengthscale is lower than the minimum lengthscale able to fit with the actual number of basis functions and box size, indicates that the number of basis functions needs to be increased or the box size decreased.

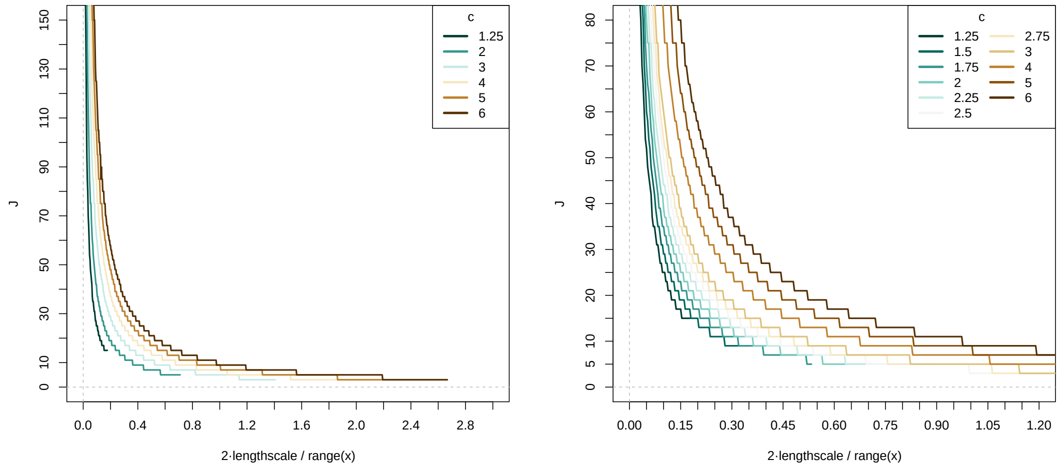

Figure 3. Relation between the minumum number of basis functions J, the lengthscale normalized by the half-range of the data ($\frac{\ell}{\frac{\text{range}(x)}{2}}$), and the boundary condition factor c ($c = \frac{L}{\frac{\text{range}(x)}{2}}$).

The right-side plots in Figures 1 and 2 are computed using the lengthscale and magnitud values of the true GP prior. Now, we focus on the estimated lengthscale and magnitud parameters after fitting the model, either the regular GP model fit and the approximate GP model fit. Figure 4 shows the posteriors distributions and the covariance functions of the approximate GP model and the regular GP model, obtained after fitting the data. Figure 5 shows the standarized root mean square error both models in function of the number of basis functions and the boundary factor. Figure 6 shows the evolution of the GP hyperparameters, lengthscale and magnitud, in function of the number of basis functions and boundary factor. Following Figure 5 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 accurate enough. This same conclusion can be roughly seen in the right-side plots in Figure **??**. This conclusions also agree with Figure 3 from which we can deduce that for a process with a normalized lengthscale of 0.3, 10 basis functions and a normalized box size (boundary factor) of 1.5 would be enough to get a good approximation following eq. (5).

From Figure 5 it can also be drawn the behaviour in terms of performance of the approximate GP model in function of the number of basis functions and the boundary factor. Two main behaviours can be deduced from Figure 5, which can also be easily

seen from Figure 3:
• as the boundary factor increases, more basis functions are needed,
• as fewer basis functions are used, the boundary factor must decrease.

Figure 6 shows the estimated lengthscale and magnitud parameter as function of the number of basis functions and the boundary factor. The same conclusion drawn before can be drawn form this figure where with 15 basis functions and a boundary factor of 1.5 the estimated lengthscale of the approximate GP model approach the estimeted lengthscale of the regular GP model. From this figure it can be drawn that exists a minumum value for the boundary factor under this never a good approach will be achieved. This conclusion also appears in Figure 3 the contour lines of the boundary factor have an end at certain lengthscales, which means that exist a minimum boundary factor depending on the lengthscale of the process.

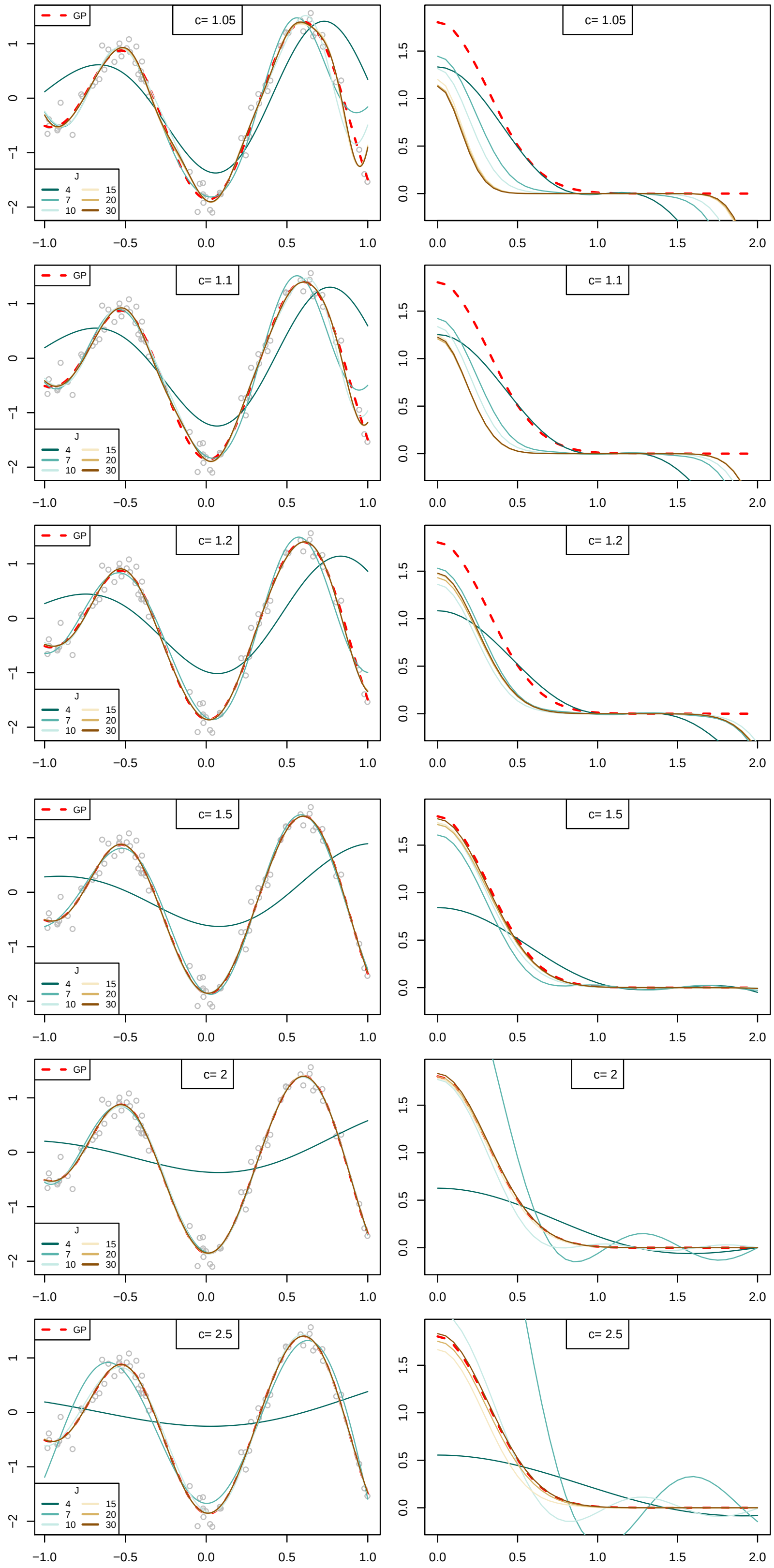

Figure 4. Posterior distributions and covariance functions of the proposed approximate GP models and the exact GP model in function of the number of basis functions J and the boundary factor c.

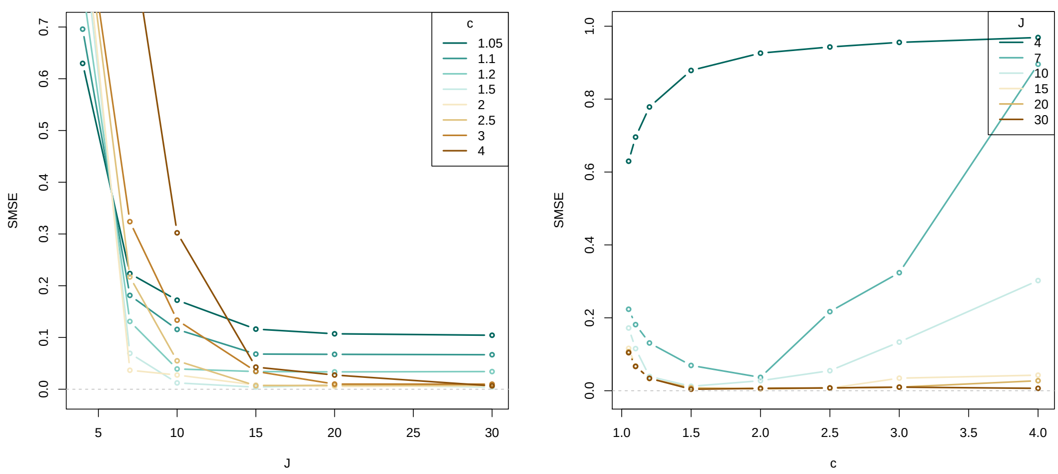

Figure 5. Standarized root mean square error (SRMSE) of the proposed approximate GP models and the exact GP model. (left) SRMSE against the number of basis functions J for different values of the boundary factor c. (right) SRMSE against the boundary factor c for different values of the number of basis functions J.

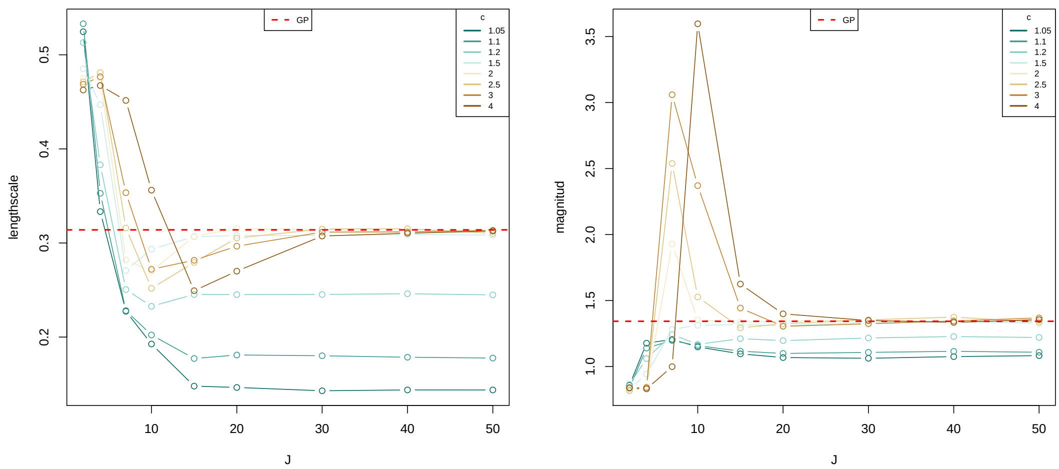

Figure 6. (left) Estimated lengthscales against the number of basis functions J for different values of the boundary factor c. (right) Estimated GP magnitud the number of basis functions J for different values of the boundary factor c.

3.5. Comparative between true lengthscales and estimated lengthscales

We fit the exact and the approximate GP model on different datasets drawn from a squared exponential GP model with different lengthscales. We fit the approximate GP model using different number of basis functions.

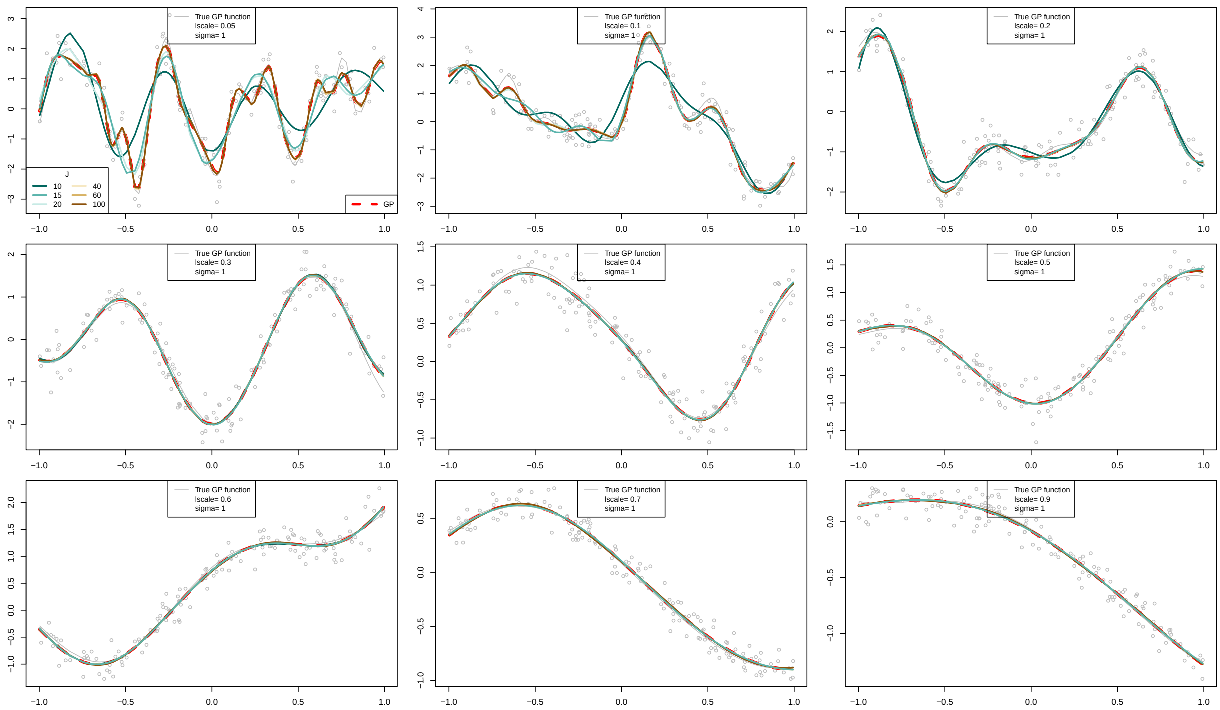

Figure 7. Posterior means of the exact and approximate GP models fitted over different datasets with different lengthscales.

Figure 8 compares the true lengthscales to the estimated lengthscales for the exact GP and the approximate GP. The black line defines the minimun lengthscale that the approximate GP model can fit for each case.

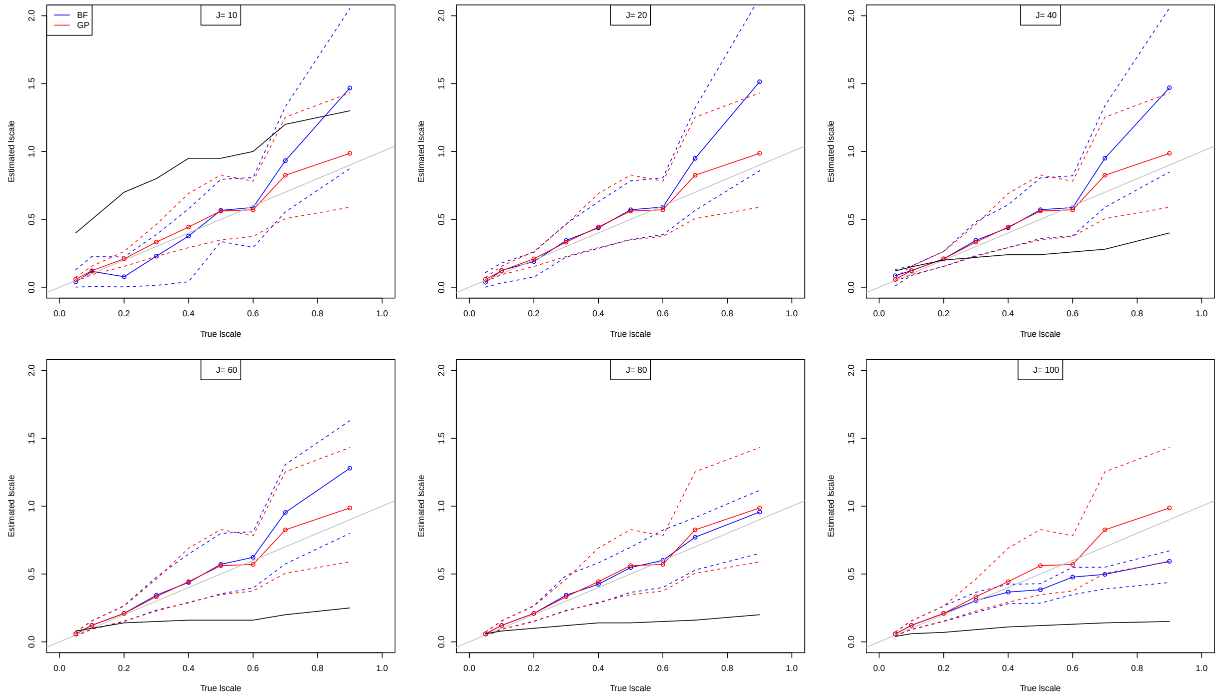

Figure 8. True lengthscales versus estimated lscales of the different realizations of datasets using different number of basis functions.

4. Univariate illustrative examples

4.1. Simulated data

A simulated one-dimensional dataset, with 250 data points drawn from a Gaussian process prior with a Matern covariance function with 3/2 degrees of freedom. The hyperparameters of this GP prior with a Matern kernel, marginal variance and lengthscale are set to 1 and 0.15, respectively. The additive Gaussian noise to the GP model is set to 0.2. We compare the modelling performance over this simulated process of our proposed approach with the performance of a regular GP 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 9 shows the posteriors distributions of the proposed model and the regular GP model, fitted over the simulated dataset. Posteriors distributions of the proposed approximation 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 simulated data correspond to out-of-sample data or test data which have not been taking part of training the models. Over this test data we can evaluate the predictive performance of every method.

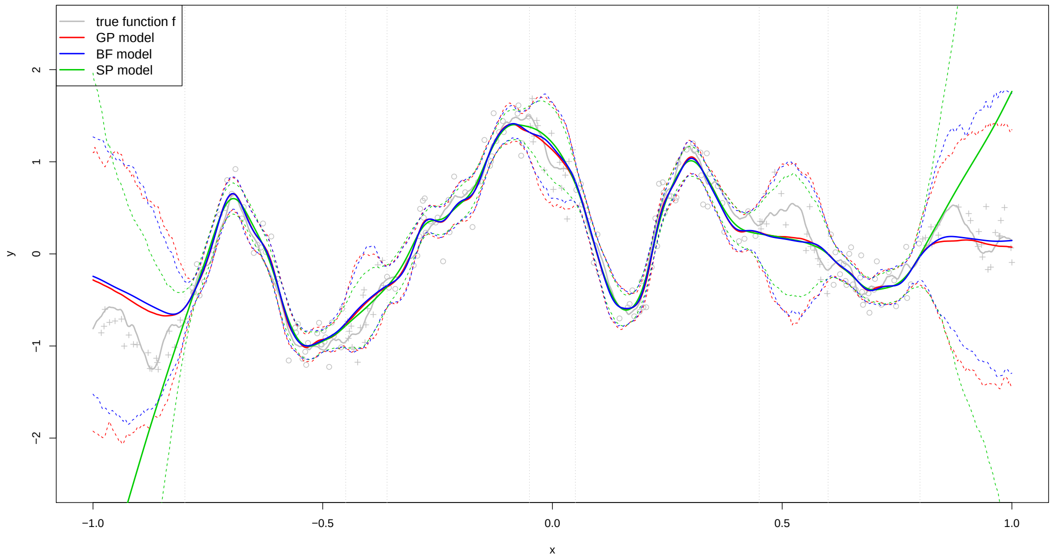

Figure 9. Posterior distributions of the proposed approximated GP model, the exact GP model, and the Splines model.

For assessing the performance of different methods we use the standardized mean squared error (SMSE) over the true function for interpolation and extrapolation data.

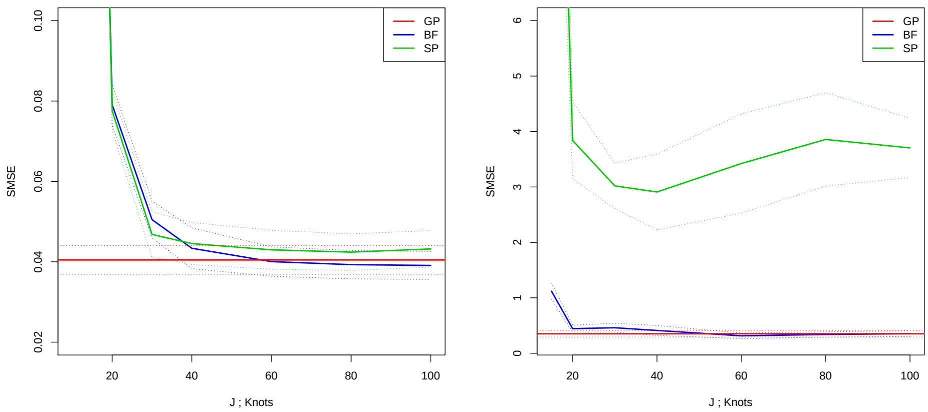

Figure 10. Standarized mean square error (SMSE) of the different methods againts the true function. (left) SMSE for interpolation. (right) SMSE for extrapolation

4.2. Case I: Gay data

4.3. Case II: Birthday data

5. Multivariate illustrative examples

5.1. Simulated data

5.2. Case III: Diabetes data

5.3. Case IV: Leukemia data

6. Novel application cases

6.1. Case V: Land use spatio-temporal classification task

Appendix A. 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

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

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:

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

Appendix B. 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:

$$\mathcal{K} f(\mathbf{x}) = \int k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') d\mathbf{x}'. \tag{B1}$$

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

$$S(\mathbf{w}) = a_0 + a_1\mathbf{w}^2 + a_2(\mathbf{w}^2)^2 + a_1(\mathbf{w}^2)^3 + \cdots \tag{B2}$$

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

$$-\nabla^2 f(\mathbf{x}) = \int k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') d\mathbf{x}', \tag{B3}$$

then the covariance function can be represented as

$$k(\mathbf{x}, \mathbf{x}') = \sum_j \lambda_j \phi_j(\mathbf{x}) \phi_j(\mathbf{x}'), \tag{B4}$$

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):

$$\begin{aligned} -\nabla^2 \phi_j(x) &= \lambda \phi_j(x), \qquad & x \in \{-L, L\} \\ \phi_j(x) &= 0, \qquad & x \notin \{-L, L\}. \end{aligned} \tag{B5}$$

a series expansion of eigenvalues and eigenfunctions

Appendix C. 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.

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* .

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.

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

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.

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.

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

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 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.