Know Your Boundaries: Constraining Gaussian Processes by Variational Harmonic Features

Arno Solin
arno.solin@aalto.fi
Department of Computer Science
Aalto University

Manon Kok
m.kok-1@tudelft.nl
Delft Center for Systems and Control
Delft University of Technology

Abstract

Gaussian processes (GPs) provide a powerful framework for extrapolation, interpolation, and noise removal in regression and classification. This paper considers constraining GPs to arbitrarily-shaped domains with boundary conditions. We solve a Fourier-like generalised harmonic feature representation of the GP prior in the domain of interest, which both constrains the GP and attains a low-rank representation that is used for speeding up inference. The method scales as $\mathcal{O}(nm^2)$ in prediction and $\mathcal{O}(m^3)$ in hyperparameter learning for regression, where n is the number of data points and m the number of features. Furthermore, we make use of the variational approach to allow the method to deal with non-Gaussian likelihoods. The experiments cover both simulated and empirical data in which the boundary conditions allow for inclusion of additional physical information.

1 INTRODUCTION

Gaussian processes (GPs, [24]) provide a widely applicable framework for probabilistic inference, where the knowledge from a measurement model (likelihood) and prior knowledge about latent functions can effectively be combined. GPs are used, for example, in robotics [3, 5, 7], spatial data analysis [16, 25], and signal processing [6, 13, 22]. Many applications call for including further information about the area or domain of interest, such as boundary conditions along some arbitrarily-shaped boundaries. Such constraints are often encountered in EEG/MEG brain data analysis (signals are constrained to the cortical surface), in simultaneous localisation and mapping (SLAM) in robotics (*e.g.*, robots constrained by walls), in spatial data analysis (*e.g.*, fish constrained to the sea), or other problems that clearly obey spatial boundaries.

Currently, the standard toolset for GPs does not include many ways for imposing constraints from arbitrarily-shaped boundaries. One way to include this information is by introducing artificial measurements with low or zero measurement noise along the boundary. Contrary to such an artificial approach, we present a method that naturally incorporates the presence of boundary conditions for arbitrarily-shaped domains.

In previous work, box-type boundaries are typically seen as a by-product of the approximation for allowing powerful representations in terms of Fourier features [8, 9, 12, 14, 21, 26]. Typically the boundaries are chosen 'far enough' from the data not to affect the results, and the main interest is in speeding up the prohibitive $\mathcal{O}(n^3)$ scaling. We also address the computational burden, but consider the presence of boundary conditions as a useful feature rather than an unfortunate necessity. One of the reasons that we can exploit the presence of boundaries is that using our method it is possible to define boundary conditions along any shape, as illustrated in Fig. 1, rather than only along rectangular or spherical domains as in previous work. Another line of previous work are time-series motivated methods (*e.g.*, [29]) and the schemes included in the R-INLA package [25]; for example [16] considers a powerful tesselation approach and addresses the inference with FEM solvers.

In GPs, apart from addressing the prohibitive computational scaling, another source of approximations are non-Gaussian likelihoods, which do not allow for closed-form inference. In this work, we use the variational approach [28] to allow for inference in non-Gaussian likelihoods. Similarly as in [9, 26], we focus on problems with low-dimensional inputs.

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

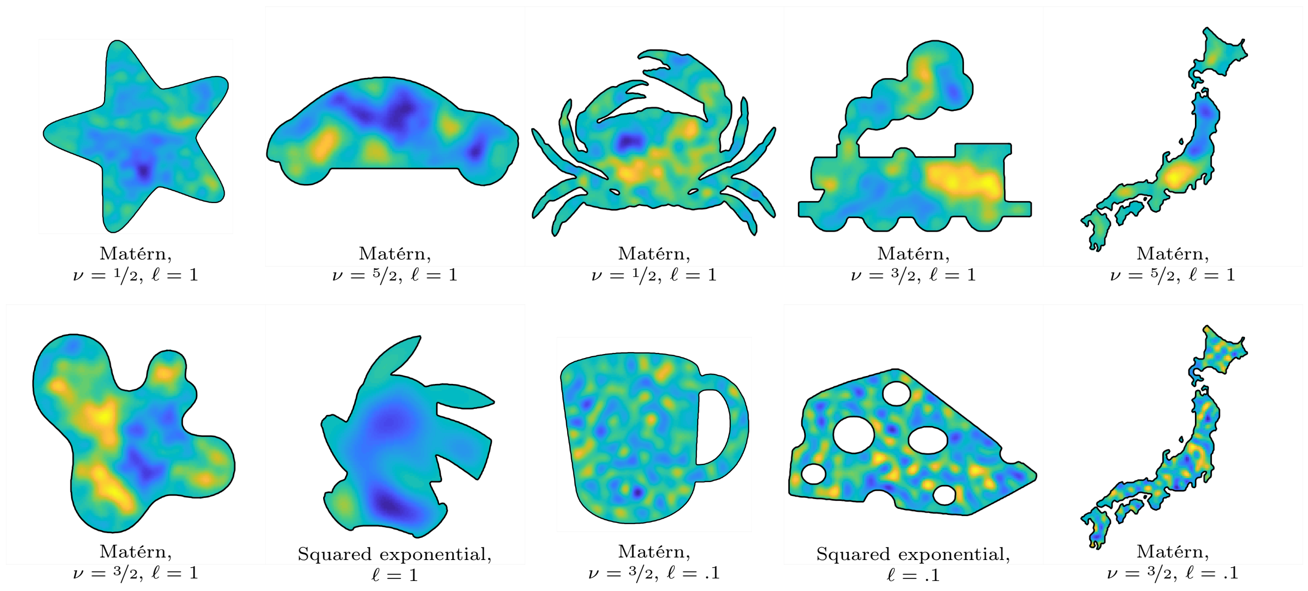


Figure 1: Random draws from Gaussian process priors constrained to 2D domains of various shapes. The process goes to zero at the boundary (black line). The approach allows for non-convex and disconnected spaces. For each domain, a random draw from a GP is shown and the assigned covariance function is shown next to the domain. The scales are arbitrary and the color map is the same as in Fig. 3.

The contributions of this paper are two-fold: *(i)* We propose a novel approach to constrain GPs to arbitrarily-shaped domains using generalised harmonic features by extending the Hilbert space GP from [26]. *(ii)* We leverage the variational Fourier features approach from [9] to allow for inference in non-Gaussian likelihoods in the constrained domains.

The paper is structured as follows. Sec. 2 goes through the necessary background in GPs and variational inference. Sec. 3 delivers the methodology for the harmonic feature decomposition and the required inference formulation. In Sec. 4 we consider both simulated data, a standard classification test data set, and a new empirical example of a constrained Log-Gaussian Cox process task. Finally, we conclude with a discussion.

2 BACKGROUND

Gaussian processes (GPs, [24]) are typically used as non-parametric priors over unknown functions, $f(\mathbf{x})$, and connected through a likelihood to some observations y. For a set of input–output pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, we can write the model as

$$f(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')), \quad \mathbf{y} \,|\, \mathbf{f} \sim \prod_{i=1}^{n} p(y_i \,|\, f(\mathbf{x}_i)). \tag{1}$$

Here, the likelihood factorises over the observations. Furthermore, $m(\mathbf{x})$ denotes an arbitrary prior mean function and $\kappa(\mathbf{x}, \mathbf{x}')$ a covariance (kernel) function. Without loss of generality we will here forth restrict the presentation to zero-mean GP priors.

A covariance function is said to be stationary if it can be written as $\kappa(\mathbf{x}, \mathbf{x}') \triangleq \kappa(\mathbf{x} - \mathbf{x}') = \kappa(\mathbf{r})$. Examples of stationary covariance functions are the squared exponential (SE) and Matérn kernels given by

$$\kappa_{\mathrm{SE}}(\mathbf{r}) = \sigma_{\mathrm{f}}^2 \exp\left(-\frac{\|\mathbf{r}\|^2}{2\ell^2}\right), \tag{2}$$

$$\kappa_{\mathrm{Mat.}}(\mathbf{r}) = \sigma_{\mathrm{f}}^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{r}\|}{\ell}\right)^{\nu} \mathrm{K}_{\nu}\left(\frac{\sqrt{2\nu}\|\mathbf{r}\|}{\ell}\right), \tag{3}$$

where σ_{f}^2 and ℓ are magnitude and lengthscale (hyper) parameters, respectively. $\mathrm{K}_{\nu}(\cdot)$ is the modified Bessel function. Compared to the squared exponential kernel, the Matérn kernel has an additional smoothness parameter ν. Characteristics for the various degrees of smoothness, can be seen in Fig. 1.

GP models have two drawbacks. The first is the prohibitive computational complexity which scales cubically in the number of data points. The other is the fact that the solution to a GP model is only available in closed form if the likelihood is Gaussian. In this special case $y_i = f(\mathbf{x}_i) + \varepsilon_i, \varepsilon_i \sim \mathrm{N}(0, \sigma_{\mathrm{n}}^2)$, the predictive mean and variance of the model functions at an unseen test input $\mathbf{x}_\star$ can be written out in closed form as

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}_\star)] &= \mathbf{k}_\star^{\mathsf{T}} (\mathbf{K} + \sigma_{\mathrm{n}}^2 \mathbf{I}_n)^{-1} \mathbf{y}, \\ \mathbb{V}[f(\mathbf{x}_\star)] &= k(\mathbf{x}_\star, \mathbf{x}_\star) - \mathbf{k}_\star^{\mathsf{T}} (\mathbf{K} + \sigma_{\mathrm{n}}^2 \mathbf{I}_n)^{-1} \mathbf{k}_\star, \end{aligned} \tag{4}$$

where $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{k}_{\star,i} = \kappa(\mathbf{x}_\star, \mathbf{x}_i)$, $i, j = 1, 2, \ldots, n$. Maximising the log marginal likelihood is typically employed for finding suitable hyperparameter values. That means minimising the negative log marginal likelihood w.r.t. $\boldsymbol{\theta}$:

$$\begin{aligned} -\log p(\mathbf{y} \mid \boldsymbol{\theta}, \mathcal{D}) = \frac{1}{2} \log |\mathbf{K}_{\boldsymbol{\theta}} + \sigma_\mathrm{n}^2 \, \mathbf{I}_n| \\ + \frac{1}{2} \mathbf{y}^\mathsf{T} (\mathbf{K}_{\boldsymbol{\theta}} + \sigma_\mathrm{n}^2 \, \mathbf{I}_n)^{-1} \mathbf{y} + \frac{n}{2} \log(2\pi). \end{aligned} \quad (5)$$

Both Eqs. (4) and (5) show the problem with naïve GP regression as both the calculation of the determinant and the matrix inverse scale as $\mathcal{O}(n^3)$.

2.1 Fourier Bases for GPs

A strategy to deal with the issue of computational complexity in GPs, is to, instead of working with the full covariance matrix, work with an approximation of it. If an eigenvalue decomposition of the covariance matrix would be available, a reduced-rank approximation $\mathbf{K} \approx \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^\mathsf{T}$ could be made, where $\boldsymbol{\Lambda}$ is a diagonal matrix of the leading m eigenvalues of $\mathbf{K}$ and $\boldsymbol{\Phi}$ is the matrix of the corresponding orthonormal eigenvectors. However, computing the eigenvalue decomposition is also of computational complexity $\mathcal{O}(n^3)$. One way to obtain an approximate eigenvalue decomposition in the case of stationary kernels has been presented in [26]. It solves the eigendecomposition of the Laplace operator subject to Dirichlet boundary conditions on a certain domain Ω as

$$\begin{cases} -\nabla^2 \phi_j(\mathbf{x}) = \lambda_j^2 \phi_j(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \quad \phi_j(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \quad (6)$$

The eigenfunctions ϕ_j and eigenvalues λ_j of the Laplace operator can be computed in closed-form for certain shapes of regular domains such as rectangles, circles, and spheres, see [26]. The approximation of the covariance function now relies on the Fourier duality of spectral densities and covariance functions, known as the Wiener–Khintchin theorem:

$$s(\boldsymbol{\omega}) = \mathcal{F}\{k(\mathbf{r})\} \quad = \int \kappa(\mathbf{r}) e^{-\mathrm{i}\, \boldsymbol{\omega}^\mathsf{T} \mathbf{r}} \, \mathrm{d}\mathbf{r}, \quad (7)$$

$$\kappa(\mathbf{r}) = \mathcal{F}^{-1}\{s(\boldsymbol{\omega})\} = \frac{1}{(2\pi)^d} \int s(\boldsymbol{\omega}) e^{\mathrm{i}\, \boldsymbol{\omega}^\mathsf{T} \mathbf{r}} \, \mathrm{d}\boldsymbol{\omega}. \quad (8)$$

This theorem relies on Bochner's theorem [2], which tells us that any continuous positive definite function, such as a covariance function, can be represented as the Fourier transform of a positive measure. If this measure has a density, it is known as the spectral density $s(\omega) \triangleq s(\|\boldsymbol{\omega}\|)$ of the covariance function. Under our convention of the Fourier transform, the spectral density corresponding to the squared exponential and Matérn covariance functions are

$$s_\mathrm{SE}(\omega) \quad = \sigma_\mathrm{f}^2 (2\pi\ell^2)^{d/2} \exp\left(-\frac{1}{2}\omega^2 \ell^2\right), \quad (9)$$

$$s_\mathrm{Mat.}(\omega) = \sigma_\mathrm{f}^2 \frac{\Gamma(\nu + d/2)}{\Gamma(\nu)} \frac{2^d \pi^{d/2} (2\nu)^\nu}{\ell^{2\nu}} \left(\frac{2\nu}{\ell^2} + \omega^2\right)^{-\frac{2\nu+d}{2}} \quad (10)$$

where d is the dimensionality of the input $\mathbf{x}$. The covariance function can now be approximated as [26]

$$\kappa(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{m} s(\lambda_j) \, \phi_j(\mathbf{x}) \, \phi_j(\mathbf{x}') = \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^\mathsf{T}, \quad (11)$$

where

$$\boldsymbol{\Phi}_i = \big(\phi_1(\mathbf{x}_i), \phi_2(\mathbf{x}_i), \ldots, \phi_m(\mathbf{x}_i)\big), \quad (12)$$

$$\boldsymbol{\Lambda} = \operatorname{diag}\left(s(\lambda_1), s(\lambda_2), \ldots, s(\lambda_m)\right), \quad (13)$$

for $i = 1, 2, \ldots, n$. Similarly, we define $\boldsymbol{\Phi}_\star$ as vectors evaluated at the prediction input location $\mathbf{x}_\star$. For Gaussian likelihoods, the prediction Eqs. (4) can now be rewritten in terms of the approximation as

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}_\star)] &\approx \boldsymbol{\Phi}_\star (\boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi} + \sigma_\mathrm{n}^2 \boldsymbol{\Lambda}^{-1})^{-1} \boldsymbol{\Phi}^\mathsf{T} \mathbf{y}, \\ \mathbb{V}[f(\mathbf{x}_\star)] &\approx \sigma_\mathrm{n}^2 \, \boldsymbol{\Phi}_\star (\boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi} + \sigma_\mathrm{n}^2 \boldsymbol{\Lambda}^{-1})^{-1} \boldsymbol{\Phi}_\star^\mathsf{T}. \end{aligned} \quad (14)$$

For this model, the expression for evaluating the negative log marginal likelihood function for hyperparameter optimisation can be written as

$$\begin{aligned} -\log p(\mathbf{y} \mid \boldsymbol{\theta}, \mathcal{D}) &= \frac{1}{2}(n - m) \log \sigma_\mathrm{n}^2 + \frac{1}{2} \sum_{j=1}^{m} [\boldsymbol{\Lambda}_{\boldsymbol{\theta}}]_{j,j} \\ &+ \frac{1}{2} \log |\sigma_\mathrm{n}^2 \boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} + \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi}| + \frac{n}{2} \log(2\pi) \\ &+ \frac{1}{2\sigma_\mathrm{n}^2} \big[\mathbf{y}^\mathsf{T} \mathbf{y} - \mathbf{y}^\mathsf{T} \boldsymbol{\Phi} (\sigma_\mathrm{n}^2 \boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} + \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi} \mathbf{y}\big], \end{aligned} \quad (15)$$

where the only remaining dependency on the covariance function hyperparameters is in the diagonal matrix $\boldsymbol{\Lambda}$ defined through the spectral density.

Since the basis functions $\boldsymbol{\Phi}$ do not depend on the hyperparameters, the product $\boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi}$ can be computed once, at order $\mathcal{O}(nm^2)$, after which evaluating the likelihood is only of $\mathcal{O}(m^3)$. However, the previous work in [26] does not deal with non-Gaussian likelihoods. We will therefore review some standard methods to do this in Sec. 2.2. In Sec. 3 we will then extend on the approach presented in this section and present a method that allows us to also deal with non-Gaussian likelihoods.

2.2 Variational Gaussian Processes for General Likelihoods

Non-Gaussian likelihoods can be handled using variational approaches (see, *e.g.*, [4]) in which the posterior is approximated by selecting the optimal distribution from a fixed family. Optimality is usually defined through the Kullback–Leibler divergence defined as (with slight abuse of notation, see also [18])

$$\begin{aligned}&\mathrm{KL}[q(f(\mathbf{x}))\,\|\,p(f(\mathbf{x})\,|\,\mathbf{y})] = \\ &\qquad \mathbb{E}_{q(f(\mathbf{x}))}\left[\log q(f(\mathbf{x})) - \log p(f(\mathbf{x})\,|\,\mathbf{y})\right]. \qquad (16)\end{aligned}$$

The variational GP approach has extensively been used in combination with inducing inputs and has recently also been combined with Fourier-based approximations [9]. In the following, we will give a short overview required for the next section, more information can for instance be found in [17]. To find a family of approximating distributions $q(f(\mathbf{x}))$, typically a set of inducing (pseudo) inputs $\mathbf{Z} = \{\mathbf{z}_j\}_{j=1}^m$ is introduced, where $m < n$. The values of the function at $\mathbf{Z}$ are denoted by $\mathbf{u} = \{\mathbf{u}_j\}_{j=1}^m$. Since $f(\mathbf{x})$ and $\mathbf{u}$ are jointly Gaussian, it is possible to write the function $f(\mathbf{x})$ conditioned on the values $\mathbf{u}$ as

$$\begin{aligned}f(\mathbf{x})\,|\,\mathbf{u} \sim \mathrm{GP}\big(&\mathbf{k}_\mathrm{u}(\mathbf{x})^\mathsf{T}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{u}, \\ &\kappa(\mathbf{x},\mathbf{x}') - \mathbf{k}_\mathrm{u}(\mathbf{x})^\mathsf{T}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{k}_\mathrm{u}(\mathbf{x}')\big). \qquad (17)\end{aligned}$$

The joint approximation for the posterior is $q(\mathbf{u})\,q(f(\mathbf{x})\,|\,\mathbf{u})$. We therefore must also choose some approximate posterior distribution $q(\mathbf{u})$, whose exact form will depend on the likelihood $p(\mathbf{y}\,|\,f(\mathbf{x}))$. This can be done by minimising the Kullback–Leibler divergence (16). Expanding the true posterior using Bayes' rule, (16) can be written as

$$\begin{aligned}&\mathrm{KL}\left[q(f(\mathbf{x}))\,\|\,p(f(\mathbf{x})\,|\,\mathbf{y})\right] \\ &= -\mathbb{E}_{q(f(\mathbf{x}))}\left[\log\frac{p(\mathbf{y}\,|\,f(\mathbf{x}))\,p(f(\mathbf{x}))}{q(f(\mathbf{x}))}\right] + \log p(\mathbf{y}) \\ &\triangleq -\mathrm{ELBO} + \log p(\mathbf{y}). \qquad (18)\end{aligned}$$

Minimising the Kullback–Leibler objective is therefore equivalent to maximising the Evidence Lower Bound (ELBO).

We factor $p(f(\mathbf{x})) = p(\mathbf{u})\,p(\mathbf{f}\,|\,\mathbf{u})\,p(f\,|\,\mathbf{f},\mathbf{u})$ with

$$\begin{aligned}p(\mathbf{u}) &= \mathrm{N}\left(\mathbf{0},\,\mathbf{K}_\mathrm{uu}\right), \\ p(\mathbf{f}\,|\,\mathbf{u}) &= \mathrm{N}\left(\mathbf{K}_\mathrm{fu}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{u},\,\mathbf{K}_\mathrm{ff} - \mathbf{K}_\mathrm{fu}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{K}_\mathrm{fu}^\mathsf{T}\right), \\ p(f(\mathbf{x})\,|\,\mathbf{f},\mathbf{u}) &= \mathrm{GP}\left(m^\star(\mathbf{x}),\,\kappa^\star(\mathbf{x},\mathbf{x}')\right), \qquad (19)\end{aligned}$$

where $m^\star(\mathbf{x})$ and $k^\star(\mathbf{x},\mathbf{x}')$ are the usual Gaussian process conditional mean and variance, conditioned on both $\mathbf{f}$ and $\mathbf{u}$. The approximate posterior process can be factored similarly as

$$\begin{aligned}q(\mathbf{f}\,|\,\mathbf{u}) &= \mathrm{N}\left(\mathbf{K}_\mathrm{fu}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{u},\,\mathbf{K}_\mathrm{ff} - \mathbf{K}_\mathrm{fu}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{K}_\mathrm{fu}^\mathsf{T}\right), \\ q(f(\mathbf{x})\,|\,\mathbf{f},\mathbf{u}) &= \mathrm{GP}\left(m^\star(\mathbf{x}),\,\kappa^\star(\mathbf{x},\mathbf{x}')\right). \qquad (20)\end{aligned}$$

Since the two processes $p(f(\mathbf{x}))$ and $q(f(\mathbf{x}))$ are the same except for $p(\mathbf{u})$ and $q(\mathbf{u})$, the ELBO (18) simplifies to

$$\begin{aligned}\mathrm{ELBO} = \mathbb{E}_{q(\mathbf{u})q(\mathbf{f}\,|\,\mathbf{u})}&\left[\log p(\mathbf{y}\,|\,\mathbf{f})\right] \\ &- \mathbb{E}_{q(\mathbf{u})}\left[\log\frac{q(\mathbf{u})}{p(\mathbf{u})}\right]. \qquad (21)\end{aligned}$$

It is possible to show that the distribution $\hat{q}(\mathbf{u})$ that maximises the ELBO is given by

$$\log\hat{q}(\mathbf{u}) = \mathbb{E}_{q(\mathbf{f}\,|\,\mathbf{u})}\left[\log p(\mathbf{y}\,|\,\mathbf{f})\right] + \log p(\mathbf{u}) + \mathrm{const}. \qquad (22)$$

This distribution is intractable for general likelihoods, but can be approximated using a Gaussian distribution [11]. Maximising the ELBO then corresponds to optimising over the mean and the covariance of this Gaussian. If the approximating Gaussian distribution $q(\mathbf{u})$ has mean $\mathbf{m}$ and covariance $\mathbf{\Sigma}$, then the entire approximating process is a GP, with

$$\begin{aligned}q(f(\mathbf{x})) &= \int q(\mathbf{u})\,q(f(\mathbf{x})\,|\,\mathbf{u})\,\mathrm{d}\mathbf{u} \\ &= \mathrm{GP}\big(\mathbf{k}_\mathrm{u}(\mathbf{x})^\mathsf{T}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{m},\,\kappa(\mathbf{x},\mathbf{x}') + \\ &\qquad \mathbf{k}_\mathrm{u}(\mathbf{x})^\mathsf{T}(\mathbf{K}_\mathrm{uu}^{-1}\mathbf{\Sigma}\mathbf{K}_\mathrm{uu}^{-1} - \mathbf{K}_\mathrm{uu}^{-1})\mathbf{k}_\mathrm{u}(\mathbf{x}')\big). \qquad (23)\end{aligned}$$

For the special case where the data-likelihood $p(y_i\,|\,f(\mathbf{x}_i))$ is Gaussian with noise-variance σ_n^2, the optimal distribution for $q(\mathbf{u})$ is given by [10]

$$\begin{aligned}\hat{q}(\mathbf{u}) &= \mathrm{N}\left(\hat{\mathbf{m}},\,\hat{\mathbf{\Sigma}}\right), \\ \hat{\mathbf{\Sigma}} &= [\mathbf{K}_\mathrm{uu}^{-1} + \sigma_\mathrm{n}^{-2}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{K}_\mathrm{uf}\mathbf{K}_\mathrm{uf}^\mathsf{T}\mathbf{K}_\mathrm{uu}^{-1}]^{-1}, \qquad (24) \\ \hat{\mathbf{m}} &= \sigma_\mathrm{n}^{-2}\hat{\mathbf{\Sigma}}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{K}_\mathrm{uf}\mathbf{y},\end{aligned}$$

and the ELBO at this optimal point is

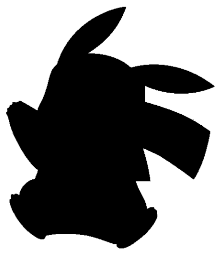
$$\begin{aligned}\mathrm{ELBO}\,(q) = \log\mathrm{N}&\left(\mathbf{y}\,|\,0,\,\mathbf{K}_\mathrm{fu}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{K}_\mathrm{uf} + \sigma_\mathrm{n}^2\mathbf{I}\right) \\ &- \frac{1}{2}\sigma_\mathrm{n}^{-2}\mathrm{tr}\left(\mathbf{K}_\mathrm{ff} - \mathbf{K}_\mathrm{fu}\mathbf{K}_\mathrm{uu}^{-1}\mathbf{K}_\mathrm{uf}\right). \qquad (25)\end{aligned}$$

3 METHODS

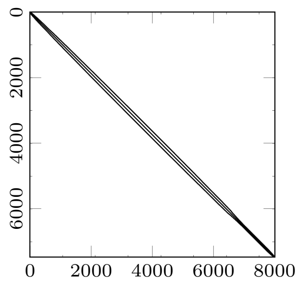
In this paper, the main interest is in including boundary conditions in the standard GP model (1) resulting in

$$\begin{aligned}&f(\mathbf{x}) \sim \mathrm{GP}(0,\kappa(\mathbf{x},\mathbf{x}')), \\ &\text{s.t. } f(\mathbf{x}) = 0, \quad \mathbf{x}\in\partial\Omega, \qquad (26) \\ &\mathbf{y}\,|\,\mathbf{f} \sim \prod_{i=1}^n p(y_i\,|\,f(\mathbf{x}_i)),\end{aligned}$$

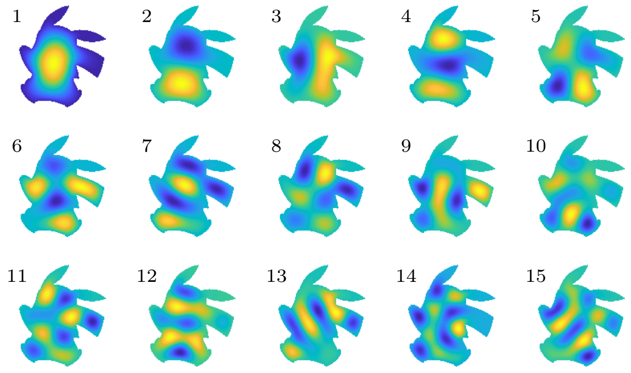
where $\Omega \subset \mathbb{R}^d$ is the domain of interest and $\partial\Omega$ denotes its boundary. This means that *a priori* we assume the GP to be generated by a GP prior with stationary covariance function $\kappa(\mathbf{x},\mathbf{x}')$ with the additional



(a) Domain



(b) Sparse stencil matrix



(c) Harmonic basis functions

Figure 2: Example domain for which we can numerically compute the harmonic basis functions using the sparse stencil matrix.

constraints of smoothly attaining the given boundary constraint. Thus the effective covariance function in the domain-aware GP prior is highly non-stationary.

Note that without loss of generality we can change the boundary constraint to be constant. Other boundary conditions, such as Neumann conditions (derivative going to zero at boundary) can also be included. Since we no longer assume a rectangular or spherical domain, it is no longer possible, like in Sec. 2.1, to compute the eigendecomposition of the Laplace operator in closed form. In this section we will now first discuss how to compute the harmonic features numerically. We will then discuss how the variational approach from Sec. 2.2 can be used to handle non-Gaussian likelihoods.

3.1 Computing the Harmonic Features

Instead of computing the eigendecomposition of the Laplace operator in closed form as in Sec. 2.1, we solve the eigendecomposition numerically. We first turn our domain of interest into a grid mask (*e.g.*, in Fig. 2 we use a 162×162 grid). We then approximate the Laplacian using a 9-point finite difference approximation with step size h (determined by the physical size of the domain) as

$$\begin{aligned}
&-\nabla^2 u(x_1,x_2) \\
&\approx \tfrac{1}{h^2}\Big[\tfrac{2}{3}u(x_1+h,x_2)+\tfrac{2}{3}u(x_1-h,x_2)+\tfrac{2}{3}u(x_1,x_2+h) \\
&+\tfrac{2}{3}u(x_1,x_2-h)+\tfrac{1}{3}u(x_1+h,x_2+h)+\tfrac{1}{3}u(x_1+h,x_2-h) \\
&+\tfrac{1}{3}u(x_1-h,x_2+h)+\tfrac{1}{3}u(x_1-h,x_2-h)-\tfrac{10}{3}u(x_1,x_2)\Big]. \qquad (27)
\end{aligned}$$

This operation can be written as an operation by a sparse *stencil* matrix $\mathbf{S}$. By letting this stencil matrix only work on locations that are inside the domain Ω, the boundary conditions can naturally be included. Let us consider an irregularly-shaped domain Ω, for instance the one displayed in Fig. 2a. This results in a stencil matrix (of size $162^2 \times 162^2$) with the sparsity pattern displayed in Fig. 2b. The number of non-zero entries in this stencil matrix is 65,596 which corresponds to ~1‰ of the elements.

We form the stencil corresponding to the domain of interest and solve the m largest, real eigenvalues λ_j^2 and the corresponding eigenvectors $\phi_j(\mathbf{x})$ of the stencil matrix using a Krylov–Schur algorithm [15, 27]. The implementation is part of ARPACK (https://www.caam.rice.edu/software/ARPACK/) and callable in MATLAB as `eigs` and in Python through *scipy.sparse.linalg.eigs*. The first 25 harmonic basis functions of the example domain from Fig. 2a are shown in Fig. 2c.

Note that using a Taylor expansion of the different terms in (27), it can be shown that in (27) we actually compute $-\nabla^2 u(x_1,x_2) - \frac{h^2}{12}\nabla^4 u(x_1,x_2) + \mathcal{O}(h^4)$. Ignoring these higher order terms, the eigenvalue problem that is being solved is therefore instead

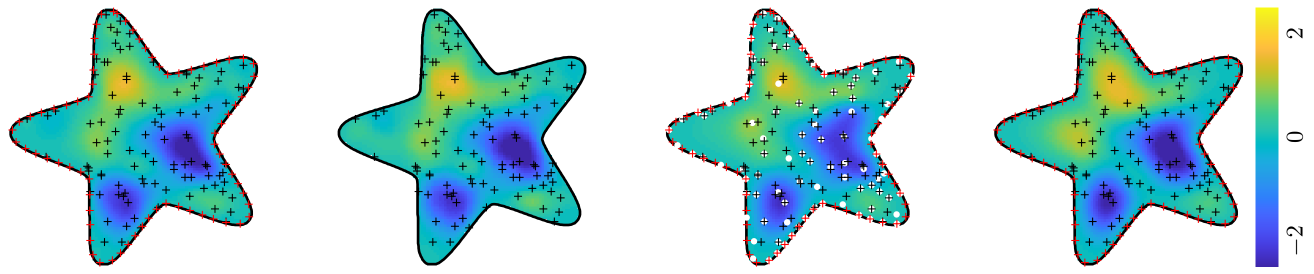
$$-\nabla^2 u(x_1,x_2) - \frac{h^2}{12}\nabla^4 u(x_1,x_2) = \lambda^2 u(x_1,x_2). \qquad (28)$$

The eigenvalues can be corrected for this as

$$\bar{\lambda}_j^2 = 2\lambda_j^2 \Big/ \sqrt{1 + \frac{\lambda_j^2 h^2}{3}} + 1\,. \qquad (29)$$

3.2 Variational Harmonic Features

The variational approach from Sec. 2.2 can be used for variational inference in a model in which the Fourier features are either computed in closed-form as in Sec. 2.1 or numerically as in Sec. 3.1. The harmonic features will hereby play a similar role as the inducing points in Sec. 2.2. Assume that our function values are defined in terms of our features as $f(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x})\mathbf{u}$ and that the prior over these features is $p(\mathbf{u}) = \mathrm{N}(\mathbf{0}, \mathbf{\Lambda})$, where $\mathbf{\Phi}(\mathbf{x})$ and $\mathbf{\Lambda}$ are defined in (12) and (13), respectively. Similar to (17), the function values and inducing inputs are then jointly Gaussian. We will now follow a similar approach to the one taken in Sec. 2.2 and approximate



(a) FULL (b) OUR (c) FITC (d) VFF

Figure 3: Example results for the regression example in a star-shaped domain. The boundary (black line) enforces the process to go to zero. The black crosses are the training inputs. In the comparison methods, the red crosses are the noise-free boundary measurements. In (c), the white dots are the inducing inputs. Even in this simple example, (c) and (d) have difficulties predicting the left-side arm.

the posterior by minimising the Kullback–Leibler divergence. Hence, we are again interested in computing the posterior $q(\mathbf{u})$ that maximises the ELBO defined in (21).

For the special case of a Gaussian likelihood, the posterior $q(\mathbf{u})$ is again given by (24). Using the relation that for our model $\mathbf{K}_{\mathrm{uu}} = \boldsymbol{\Lambda}$ and $\mathbf{K}_{\mathrm{fu}} = \boldsymbol{\Phi\Lambda}$, we can write the posterior $q(\mathbf{u})$ as

$$\hat{q}(\mathbf{u}) = \mathrm{N}(\hat{\mathbf{m}}, \hat{\boldsymbol{\Sigma}}), \qquad \hat{\boldsymbol{\Sigma}} = [\boldsymbol{\Lambda}^{-1} + \sigma_n^{-2}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}]^{-1}, \\ \hat{\mathbf{m}} = \sigma_n^{-2}\hat{\boldsymbol{\Sigma}}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\mathsf{T}\mathbf{y}. \tag{30}$$

Using the relation $f(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x})\mathbf{u}$, the posterior over the function values $\mathbf{f}$ is therefore given by

$$\mathbf{f} \sim \mathrm{N}(\hat{\mathbf{m}}_\mathrm{f}, \hat{\boldsymbol{\Sigma}}_\mathrm{f}), \quad \hat{\boldsymbol{\Sigma}}_\mathrm{f} = \boldsymbol{\Phi}[\boldsymbol{\Lambda}^{-1} + \sigma_n^{-2}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}]^{-1}\boldsymbol{\Phi}^\mathsf{T}, \\ \hat{\mathbf{m}}_\mathrm{f} = \sigma_n^{-2}\boldsymbol{\Phi}\hat{\boldsymbol{\Sigma}}_\mathrm{f}\boldsymbol{\Phi}^\mathsf{T}\mathbf{y}. \tag{31}$$

Note that these equations exactly correspond to (14) presented in Sec. 2.1 if these are used to predict on $\mathbf{x}$. Using this relation between the methods, it is now possible to use the variational inference scheme from Sec. 2.2 together with the harmonic features derived in Sec. 3.1 to do inference in the model (26) for arbitrary boundary conditions and general likelihoods.

4 EXPERIMENTS

We include three different experiments showing the properties of our method. The first example is a simulation study that compares our method to alternative solutions in a regression setup. The second example considers the *banana* classification dataset with a hard decision boundary. The third example is a new empirical example of modelling tick bite density.

The methods in Sec. 3 were implemented in both Mathworks MATLAB and Python. The codes for replicating them are available at https://github.com/AaltoML/boundary-gp. We leveraged the GP machinery available in TensorFlow [1] and GPflow [19]. GPflow was also used for the comparison methods.

4.1 Benchmarking

We consider a simulated setup where we choose a star-shaped domain of unit width and a Dirichlet boundary condition enforcing the process to be zero at the boundary. We simulate 10 data sets which obey the assumption of the process going to zero at the boundary, observe $n = 100$ uniformly random input locations in the domain (black crosses in Fig. 3), and add Gaussian noise to the measurements.

As a baseline, we solve the GP regression problem naïvely by including a number of noise-free observations along the boundary. We use 73 points along the boundary (red crosses in Fig. 3) which are considered as normal measurements in the GP regression problem, but with a noise scale of zero. We compare our approach to two general-purpose schemes for rank-reduced GP regression: The (gold-standard) *Fully independent training conditional* (FITC) method (see [23]) and *Variational Fourier features* (VFF, [9]). With the expectation that increasing the number of inducing points should improve every method, we applied 4 to 100 inducing points/features. For all models, we used a Gaussian process prior with a Matérn ($\nu = 3/2$) covariance function. In the FULL, FITC, and our method the standard non-separable covariance function was used, while VFF decomposed the covariance function to a product of two Matérns over the different input dimensions. This Kronecker structure in VFF needs less inducing features, and including too many features leads to instability in evaluation of the model, as previously discussed in [9] (thus $m = 100$ is not included for VFF). For FITC and VFF, the noise-free inputs were problematic for hyperparameter and inducing input location optimisation, resulting in numerical

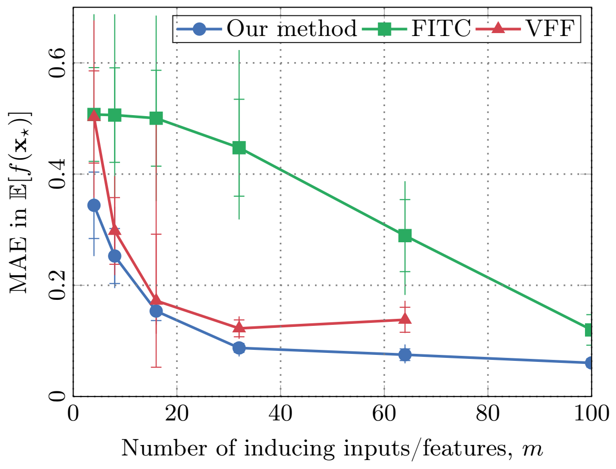


Figure 4: The effect of increasing the number of inducing inputs/features in the star-shaped domain regression study. The curves show the mean absolute error (± std/min/max) in predictive mean compared to the results of the FULL GP model with noise-free observations along the boundary.

instability. These problems were not encountered in the FULL and proposed model. For a fair comparison of representative power, we hence fixed the hyperparameters ($\sigma_\mathrm{f}^2 = 1$, $\ell = 0.1$, $\sigma_\mathrm{n}^2 = 0.1^2$) for all models and chose the inducing input locations for FITC by k-means clustering.

Fig. 3 shows the predictive mean for one of the data sets with the four different methods ($m = 64$). The naïve FULL GP with the noise-free observations clearly best agrees with the proposed method that can directly use the boundary information as part of the GP prior. FITC and VFF do a fair job but differ even visually from the results in (a)–(b). Clear differences can be seen in the blue middle part and in the left arm.

For a quantitative evaluation, we compared estimates for all ten data sets and a differing number of inducing inputs/features. Fig. 4 shows the effect of increasing the number of inducing inputs/features and reports the mean absolute error (MAE) for the predictive mean compared to the results of the FULL GP. The results show that including the boundary information directly has clear benefits. The same conclusion can be drawn from analysis of the marginal log-likelihoods and the predictive marginal variance estimates. In terms of computation time, our model is on par with VFF, with additional computational saving in evaluation of the marginal likelihood. Furthermore, the noise-free boundary measurements bring additional computational burden to the general-purpose schemes, while our model considers the boundary directly.

4.2 Banana Classification Dataset

We consider the banana classification dataset (see [11]) with a hard decision boundary. We perform variational classification using a Gaussian approximation to the posterior $q(\mathbf{u})$ and optimise the ELBO with respect to the mean and variance of the approximation. We expect that increasing the number of harmonic features leads to an improved approximation; the variational framework guarantees that more inducing variables is monotonically better [28]. However, this does not necessarily hold due to the restriction on $q(\mathbf{u})$.

Fig. 5 shows the two classes in the banana data set with red and blue markers. The pre-defined *hard decision boundary* is the solid black boundary enclosing all the data. It indicates the boundary outside of which the data does not play a role and at which there is absolute uncertainty of the correct class. We use a Bernoulli likelihood and a Matérn ($\nu = 5/2$) kernel for the GP prior. We train the hyperparameters of the model jointly with the variational approximation. Fig. 5 shows the classification model outcomes with different numbers of inducing harmonic features. The black lines are the decision boundaries, which clearly improve with the increasing number of features.

4.3 Tick Bite Density Estimation

Ticks are small arachnids, typically 3–5 mm long. They are external parasites that live by feeding on blood typically of mammals and birds. Because of this, they may carry diseases that affect humans and other animals. To monitor the spread of ticks and of the diseases they might carry, many countries ask people to report tick bites. One such platforms is https://tekenradar.nl which is an initiative of the National Institute for Public Health and the Environment and Wageningen University, and collects data about tick bites in the Netherlands. The data is accessible online, and we used data of tick bites collected by this platform during the first 9 months of 2018. The 4,446 data points are scattered over the country as shown in Fig. 6.

We use this data to model the tick density and exploit physical prior knowledge that ticks only live on land. The boundaries of our domain reflect this knowledge by assuming that the tick density is zero at sea, in the large lake in the middle of the country and in various rivers and lakes in the country. The area outside the domain in Fig. 6 is shown in white. We model the tick density using the number of ticks in a grid of roughly $n = 15{,}000$ points and a Poisson likelihood in a Log-Gaussian Cox process model [20]. We used a Matérn ($\nu = 3/2$) covariance function, and we train the hyperparameters of the model jointly with the variational approximation. The results are shown in

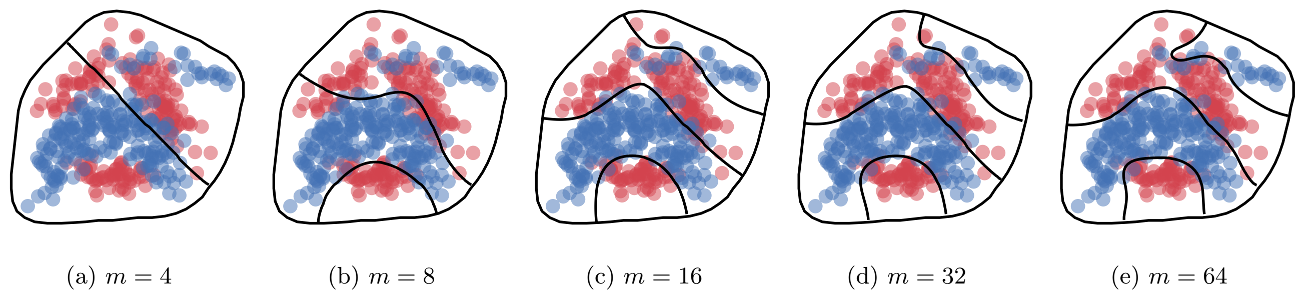


Figure 5: The effect of increasing the number of inducing features for the *banana* classification dataset with a hard decision boundary. In each pane, the coloured points represent training data and the decision boundaries are black lines. The outermost line is the pre-defined hard decision boundary.

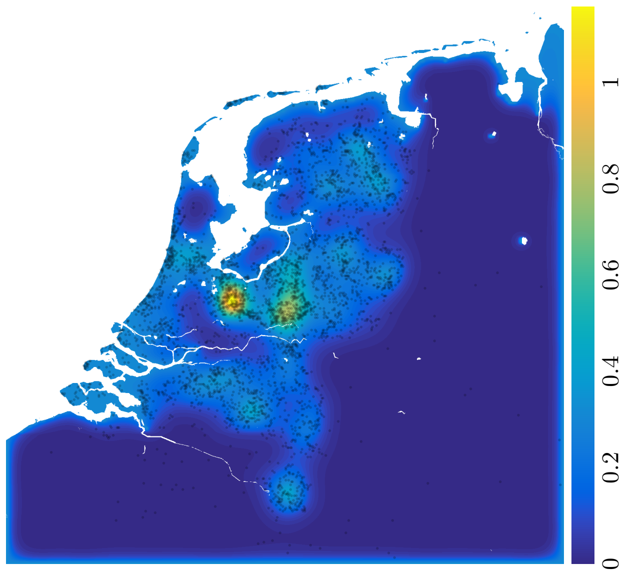


Figure 6: Predicted density of tick bites per square kilometre around the Netherlands. The map is 400 km wide and the sea, rivers, and lakes (in white) are constraining the domain. The data (reported tick bites) are shown by the black dots, and they are modelled as a Log-Gaussian Cox process (Poisson likelihood).

Fig. 6, where the colour indicates the intensity tick bites per square kilometre. As can be seen, this density explains the data well. For the eigendecomposition we used $m = 256$ and a grid size of 200×200.

5 DISCUSSION AND CONCLUSION

We have considered the problem of including physical knowledge about spatial constraints in GP inference. A numerical computation of the harmonic features of the GP prior allows us to specify boundary conditions on arbitrary shapes while at the same time attaining a low-rank representation that is used for speeding up inference. By approximating the posterior using variational inference, it is possible to use our method for non-Gaussian likelihoods.

We have illustrated the efficacy of our method using both simulated and experimental data, with both Gaussian and non-Gaussian likelihoods. For the proposed method, there is a fixed setup cost for each new kind of domain, after which the method is as fast as the Hilbert GP (and thus slightly faster than VFF). In Sec. 4.1 ($m = 100$) the setup cost was 4.9±0.1 s, and for the banana example (Sec. 4.2 for $m = 64$) the setup cost was 10.3±0.4 s. After the preparation cost, the computation time for the actual GP inference is fast. Evaluating the marginal likelihood (or doing GP prediction) in Sec. 4.1 takes 0.23 s when we have increased the number of observations to $n = 10{,}000$ for the numbers to make more practical sense. If the variational approximation is used, most time is spent in the optimiser—as seen in the experiment in Sec. 4.3, where the setup cost was in the range of some hundred seconds, but the optimiser ran for 25 min. This would, of course, be the same for all methods using the variational approximation.

The examples that we have presented only consider two-dimensional input domains. Higher-dimensional domains can, however, be considered in similar fashion as in [9]. Although the number of required basis functions grows exponentially in the input dimensionality d, Kronecker products and sums across the input dimensions have previously been used to address this problem. Throughout the paper we have focussed on using Dirichlet boundary conditions. The method can, however, be extended to be used with other boundary conditions, such as Neumann conditions.

The main merit of our approach is that due to its simple and straightforward formulation, it can easily be extended or used in larger-scale systems. For example, the approach can be extended to spatio-temporal analysis or be used in frameworks for simultaneous localisation and mapping (SLAM). The codes are available at https://github.com/AaltoML/boundary-gp.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[2] Akhiezer, N. I. and Glazman, I. M. (1993). *Theory of Linear Operators in Hilbert Space*. Dover, New York.

[3] Anderson, S., Barfoot, T. D., Tong, C. H., and Särkkä, S. (2015). Batch nonlinear continuous-time trajectory estimation as exactly sparse gaussian process regression. *Autonomous Robots*, 39(3):221–238.

[4] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[5] 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(2):408–423.

[6] Deisenroth, M. P., Turner, R. D., Huber, M. F., Hanebeck, U. D., and Rasmussen, C. E. (2012). Robust filtering and smoothing with Gaussian processes. *IEEE Transactions on Automatic Control*, 57(7):1865–1871.

[7] Ferris, B., Fox, D., and Lawrence, N. D. (2007). WiFi-SLAM using Gaussian process latent variable models. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 7, pages 2480–2485, Hyderabad, India.

[8] Fritz, J., Neuweiler, I., and Nowak, W. (2009). Application of FFT-based algorithms for large-scale universal kriging problems. *Mathematical Geosciences*, 41(5):509–533.

[9] Hensman, J., Durrande, N., and Solin, A. (2018). Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52.

[10] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Uncertainty in Artificial Intelligence (UAI)*, pages 282–290. AUAI Press.

[11] Hensman, J., Matthews, A. G., and Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 351–360.

[12] John, S. and Hensman, J. (2018). Large-scale Cox process inference using variational Fourier features. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2362–2370.

[13] Kok, M. and Solin, A. (2018). Scalable magnetic field SLAM in 3D using Gaussian process maps. In *Proceedings of the 21st International Conference on Information Fusion*, Cambridge, UK.

[14] Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881.

[15] Lehoucq, R. B., Sorensen, D. C., and Yang, C. (1998). *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. Siam.

[16] Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498.

[17] Matthews, A. G. (2016). *Scalable Gaussian Process Inference Using Variational Methods*. PhD thesis, University of Cambridge, Cambridge, UK.

[18] Matthews, A. G., Hensman, J., Turner, R. E., and Ghahramani, Z. (2016). On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 231–238.

[19] Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

[20] Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482.

[21] Paciorek, C. J. (2007). Bayesian smoothing with Gaussian processes using Fourier basis functions in the spectralGP package. *Journal of Statistical Software*, 19(2).

[22] Pérez-Cruz, F., Van Vaerenbergh, S., Murillo-Fuentes, J. J., Lázaro-Gredilla, M., and Santamaria, I. (2013). Gaussian processes for nonlinear signal processing: An overview of recent advances. *IEEE Signal Processing Magazine*, 30(4):40–50.

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

[24] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

[25] Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392.

[26] Solin, A. and Särkkä, S. (2014). Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*.

[27] Stewart, G. W. (2002). A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614.

[28] Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 567–574.

[29] Yang, J., Rattray, M., Penfold, C. A., and Grant, M. R. (2016). Inferring the perturbation time from biological time course data. *Bioinformatics*, 32(19):2956–2964.