HILBERT-SPACE REDUCED-RANK METHODS FOR DEEP GAUSSIAN PROCESSES

Muhammad F. Emzir[1)], *Sari Lasanen* [2)], *Zenith Purisha* [1)], *Simo Särkkä* [1)*]

[1)]Department of Electrical Engineering and Automation, Aalto University
[2)] School of Engineering Science, LUT University

ABSTRACT

A deep Gaussian process is a hierarchy of Gaussian processes where the process at each level is Gaussian given the process on the next level. In this paper, we recast special deep Gaussian processes as solutions of stochastic partial differential equations (SPDEs). Each of these SPDEs has parameters which are functions of the solutions to other SPDEs. To avoid solving SPDEs explicitly, we transform the SPDEs to finite-dimensional objects by truncating the underlying Hilbert space expansion. We then use a Markov chain Monte Carlo technique designed for function spaces to sample its posterior distribution. For a one-dimensional signal example, we show that the regression can offer discontinuity detection and smoothness constraints, which are competing with each other.

***Index Terms*—** Deep Gaussian processes, Gaussian process, Bayesian inference, Markov chain Monte Carlo

1. INTRODUCTION

Among many tools developed recently for both machine learning and statistical inference, Gaussian processes have proven to be very successful [1–3]. The main reason for this is the simplicity and flexibility of Gaussian processes. Namely, only mean and covariance functions are needed to fully characterize Gaussian processes, which are sufficiently flexible to perform many regression and classification tasks [4, 5].

There are a vast amount of studies on Gaussian processes and their applications where the process of interest is assumed to be stationary [6]. The stationary Gaussian processes have uniform spatial behavior. To model variable spatial behaviour, the covariance structure needs to be tuned appropriately. As a result, the stationary Gaussian processes fail in the cases where the smoothness of the target varies spatially in unexpected ways [5, 7]. There have been a number of proposals to increase flexibility of the non-stationary Gaussian processes (see e.g., [4, 8–10]). In this article, we use deep Gaussian processes or Gaussian processes with hyperprior [11] to tackle this problem. In many spatial inference problems, it is often useful to directly work from the infinite-dimensional space. For this particular problem, there are two approaches to construct a Bayesian inference algorithm. The first approach is to discretize the problem (e.g., via grid partitions or finite element meshes) then a Bayesian inference method is applied to the finite-dimensional setting [12]. The second approach, recently proposed in [13, 14], is to directly apply Bayesian inference to the infinite-dimensional problem, and afterward, apply a discretization method. This approach is possible through realizing that the posterior and prior probability distribution can be related via the Radon-Nikodym derivative, which can be generalized to function spaces.

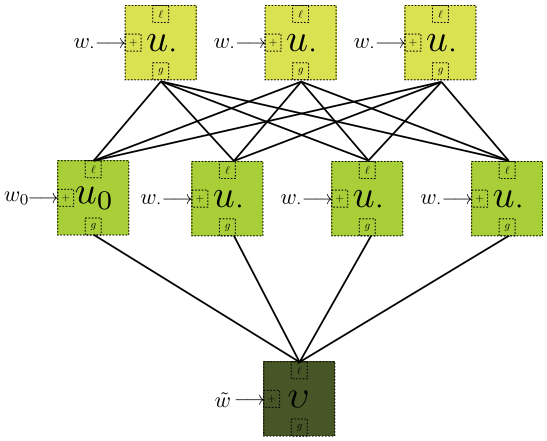


Fig. 1: Illustration of an interconnected Gaussian processes. Each node (field) has an independent white noise input field and a length-scale parameter ℓ. The length-scale ℓ is obtained as a function evaluated on the fields above it. These Gaussian fields are approximated in finite-dimensional Hilbert space using $\mathbf{L}(u_0, \cdots,)\mathbf{u}_N = \mathbf{w}_N$ (see (8)).

The contribution of this article is to present a computationally efficient Hilbert-space method for Bayesian inference in deep Gaussian process models. We follow the approach recently described in [10], where the Gaussian processes are represented as stochastic partial differential equations (SPDEs), in which the length-scale parameters depend on the solution of SPDEs for the layer above. However, in-

*This work is supported by the Academy Finland grant numbers 313708 and 314474.

978-1-7281-0824-7/19/$31.00 ©2019 IEEE

stead of using a grid partition, we will apply Galerkin methods. Using this approach, we can avoid evaluating SPDEs forward problem via finite difference equations. This can be considered as a compromise between the accuracy and the computational complexity. Our approach is also related to the one described in [15], where Gaussian processes with stationary covariance functions are approximated in finite-dimensional Hilbert spaces. Such approach relies on the fact that the covariance function of a stationary Gaussian field is expandable via Mercer's theorem [3, Theorem 4.2]. The Karhunen-Loève theorem [16] guarantees the existence of orthogonal basis expansion of square-integrable process. However, when generalized to the hyperprior settings, the arising nonlinearities are computationally problematic.

The outline of the article is the following. In Section 2, we present a formulation of deep Gaussian processes regression problem via Hilbert space expansion. In Section 3, we then propose a Markov chain Monte Carlo algorithm to sample from these Fourier coefficients posterior distributions. In Section 4 we show an application of the proposed method on a one-dimensional inference problem and finally Section 5 concludes the article.

2. HILBERT-SPACE APPROXIMATIONS

Consider Gaussian process regression with a real-valued random processes/fields $\upsilon(x) : \Omega \to \mathbb{R}, d \leq 3$ on an open bounded domain $\Omega \subset \mathbb{R}^d$. Specifically, $\upsilon \in \mathscr{H} \coloneqq \mathscr{L}^2(\Omega)$ [17]. To carry out the regression, a set of measurements is taken (either direct or indirect in multiple locations). In this article, the measurement is assumed to be a linear operation on υ corrupted with additive noises; that is, $y_k = H_k\upsilon(x) + e_k$, where H_k is a linear operator, and e_k is a zero-mean white noise with a diagonal covariance matrix $\mathbf{E}$.

We denote the measurements, noises, and the linear operators as $y = [y_1, \cdots, y_m]^\top, e = [e_1, \cdots, e_m]^\top, H = [H_1, \cdots, H_m]^\top$ respectively. In the Bayesian framework, the estimation problem is equivalent to determining the a posteriori distribution of υ given the measurements y. The Gaussian regression approach for this problem starts with assuming that υ is a Gaussian field with a certain mean (assumed zero for simplicity) and covariance function $C(x, x')$:

$$\upsilon(x) \backsim \mathcal{GP}(0, C(x, x')).$$

It is well known that the prior choice plays a significant role in determining the posterior and should be carefully selected. In the case when $C(x, x')$ is a Matérn covariance function the Gaussian field υ can be generated from a stochastic partial differential equation of the form [10, 18]

$$(\ell^2\Delta - 1)^{\alpha/2}\,\upsilon(x) = \sqrt{\beta\ell^d}\,w(x), \tag{1}$$

where $\alpha = \nu + d/2$, d is the dimension of the space, ν is a smoothness parameter, $w(x)$ is a white noise on $\mathbb{R}^d$, ℓ is the length-scale constant of the Matérn covariance function C, and β is a normalization constant, given by

$$\beta = \sigma^2\frac{2^d\pi^{d/2}\Gamma(\alpha)}{\Gamma(\nu)}.$$

To obtain a non-stationary field, we modify (1) so that the length-scale ℓ is modeled via another Gaussian field u with Matèrn covariance function. Namely, we select $\ell(x) = g(u(x))$, where g is a smooth positive function $g : \mathbb{R} \to \mathbb{R}_+$. Introducing the spatially varying length-scale $\ell(x)$ into (1), we obtain

$$(\ell^2(x)\Delta - 1)^{\alpha/2}\,\upsilon(x) = \sqrt{\beta\ell(x)^d}\,w(x). \tag{2}$$

From now on we will fix the value $\alpha = 2$ so that (2) becomes an elliptic partial differential equation. We consider the Dirichlet boundary condition. Other boundary condition can used via procedure given in [19]. With this boundary condition, the Laplace operator Δ on the domain Ω is a self-adjoint operator on the Hilbert space $\mathscr{H}$. The Laplacian can be expressed as

$$-\Delta\upsilon = \sum_{i=-\infty}^{\infty}\lambda_i\,\langle\upsilon, \phi_i\rangle\,\phi_i, \tag{3a}$$

$$\langle\phi_i, \phi_j\rangle = \delta_{ij}, \tag{3b}$$

where $\{\phi_i\}$ is a complete set orthonormal eigenfunctions of Δ, δ_{ij} is the Kronecker delta function, and $\lambda_i > 0$, where $\lim_{i\to\infty}\lambda_i = \infty$, [20, §6.5]. Observe that, in the sense of (2), $d \leq 3$, $\ell(x) \coloneqq g(u(x)) \in \mathscr{L}^\infty(\Omega)$ is a multiplication operator acting point-wise, that is, $(\ell\upsilon)(x) = \ell(x)\upsilon(x), \forall x \in \Omega$ [19]. Next, we specify finite-dimensional Hilbert subspace $\mathscr{H}_N$ of $\mathscr{H}$ as the span of $\{\phi_{-N}, \cdots, \phi_0, \cdots, \phi_N\}$. Denote by u_N the finite dimensional projection of u onto $\mathscr{H}_N$. Since we assume that the length-scale is always greater than zero, (2) can be written in terms of $\kappa(u) = 1/\ell$ as

$$\frac{1}{\sqrt{\beta}}(\kappa(u)^{-\nu}\Delta - \kappa(u)^{2-\nu})\,\upsilon(x) = w(x). \tag{4}$$

We will seek a weak solution to (4), $\mathscr{H}_N \ni \tilde{\upsilon} = \sum \upsilon_i\phi_i$, such that for any $h \in \mathscr{H}_N$,

$$\frac{1}{\sqrt{\beta}}\left\langle(\kappa(u)^{-\nu}\Delta - \kappa(u)^{2-\nu})\tilde{\upsilon}, h\right\rangle = \langle w, h\rangle. \tag{5}$$

We also assume that $u \in \mathscr{H}_N$. Let us examine a periodic boundary condition on $\mathbb{R}^d$, where we assume that Ω is a d dimensional box with length 1. Periodic boundary condition translate this problem to a d-dimensional torus. Within this boundary condition, it is useful to consider $\mathscr{H}$ as a complex Hilbert space, so that we can set $\phi_i = \exp\big(\imath\, c_d\, x^\top k(i)\big)$ as Fourier complex basis functions for d dimensions, for some constant c_d and a multi-index $k(i)$ which is unique for every

i. The multi-index k can be selected such that $k(i+j) = k(i) + k(j), \forall -N \leq i, j \leq N$ (see lexicographic order, [21, Appendix A, Chapter I]). Under the periodic boundary condition, we have

$$\begin{aligned}&\langle u\ \phi_i, \phi_j\rangle \\ &= \int_\Omega \sum_{m=-\infty}^{\infty} \exp\big(\imath\, c_d\, x^\top (k(m)+k(i)-k(j))\big) u_{(m)} dx \\ &= u_{(j-i)}.\end{aligned} \tag{6}$$

Let us denote with $\mathbf{u}_N = [u_{(-N)} \cdots u_{(N)}]^\top$, and $M_N(u)$ the matrix representation of the multiplication operator $u(x)$ on $\mathscr{H}_N$. The previous discussion allow us to write

$$M_N(u) \coloneqq T(\big[0 \cdots 0\ u_{(-N)} \cdots u_{(N)}\ 0 \cdots 0\big]^\top). \tag{7}$$

In (7), T is a Toeplitz matrix. By definition $M_N(u) = M_N(u_N)$. Since $\{\phi_i\}$ are orthonormal, (5) is equivalent to the following equation,

$$\begin{aligned}&\mathbf{L}(u)\mathbf{v}_N = \mathbf{w}_N, \\ &\mathbf{L}(u) \coloneqq \frac{1}{\sqrt{\beta}}\big(M_N(\kappa(u)^{-\nu})\mathbf{D} - M_N(\kappa(u)^{2-\nu})\big),\end{aligned} \tag{8}$$

where $\mathbf{D}$ is $(2N+1)$ diagonal matrix, $\mathbf{v}_N, \mathbf{w}_N$ are $\mathbb{C}^{2N+1}$ vectors, $\mathbf{D}_{i,i} = -\lambda_i$, and $\mathbf{v}_{(i)} = \langle \upsilon, \phi_i\rangle, \mathbf{w}_{(i)} = \langle w, \phi_i\rangle$. Notice that since the white noise field $w(x)$ is assumed to be real, $\mathbf{w}_N$ comprised of $2N+1$ entries, where $w_{(-i)} = \overline{w_{(i)}}$, and $w_{(0)}$ is real, and $w_{(i)} \sim N(0,1)$ are independent.

During the implementation, we approximate $M_N(\kappa(u)^\gamma)$ with $M_N(\kappa(u_N)^\gamma)$, $\gamma \in \mathbb{R}$; hence $\mathbf{L}(u) \approx \mathbf{L}(\mathbf{u}_N)$. By (6), the entries of $M_N(\kappa(u)^\gamma)$ are given by $\langle \kappa(u)^\gamma, \phi_j\rangle$. Calculation of $M_N(\kappa(u_N)^\gamma)$ can be done fast by using fast Fourier transform algorithms [22]. Using the same notation, we can generalize further to multilayered deep Gaussian fields with zero means. Finally, we can write the measurement as

$$y_k = \sum_{i=0} \langle \phi_i, H_k\rangle \langle \upsilon, \phi_i\rangle + e_k.$$

3. BAYESIAN INFERENCE ALGORITHM

Let us denote by $\mu_0(\upsilon) = \mathbb{P}(\upsilon)$ and $\mu_y = \mathbb{P}(\upsilon|y)$ the prior and the posterior distribution respectively. We assume that the measurement noise $\mathbf{e} \in \mathbb{R}^m$ is finite-dimensional with density function $\pi(\mathbf{e})$ that is Gaussian, with zero mean and covariance E. The likelihood density is given by $\pi(y - H\upsilon)$. The posterior distribution μ_y will be absolutely continuous with respect to the prior μ_0, and hence, Radon-Nikodym theorem leads us to

$$\begin{aligned}&\frac{\mathrm{d}\mu_y}{\mathrm{d}\mu_0} = \frac{1}{Z}\exp(-\Phi(\upsilon, y)), \\ &\Phi(\upsilon, y) \coloneqq \frac{1}{2}\Big\|E^{-1/2}(y - H\upsilon)\Big\|^2, \\ &Z \coloneqq \int_{\mathscr{H}} \exp(-\Phi(\upsilon, y)) d\mu_0(\upsilon).\end{aligned}$$

Consider the case with one hyperprior layer, that is, the field of interest υ has a length-scale ℓ as function of u. The field u is independent of υ and has a Gaussian distribution with Matèrn covariance function. The posterior probability density of the Fourier expansion coefficients $\mathbf{u}$:

$$\begin{aligned}p(\mathbf{u}_N, \mathbf{v}_N|\mathbf{y}) \propto& p(\mathbf{y}|\mathbf{u}_N, \mathbf{v}_N) p(\mathbf{u}_N, \mathbf{v}_N) \\ \propto& p(\mathbf{y}|\mathbf{u}_N, \mathbf{v}_N) p(\mathbf{v}_N|\mathbf{u}_N) p(\mathbf{u}_N).\end{aligned} \tag{9}$$

Having defined the posterior probability density, our next task is to sample from this probability distribution. In particular, our algorithm will construct a Markov chain of $\left(\mathbf{u}_N^{(k)}, \mathbf{v}_N^{(k)}\right)$ with transition probabilities

- $\mathbf{v}_N^{(k+1)} \sim p_{(\mathbf{u}_N^{(k)}, \mathbf{y})}(\mathbf{v}_N^{(k)}, \cdot)$,
- $\mathbf{u}_N^{(k+1)} \sim q_{(\mathbf{v}_N^{(k+1)}, \mathbf{y})}(\mathbf{u}_N^{(k)}, \cdot)$.

We discretize υ as in Section 2. The prior distribution of the coefficients $\mathbf{v}_N$ is given by (8) where u_N has Fourier coefficients $\mathbf{L}_0^{-1}\tilde{\mathbf{w}}$, $\mathbf{L}_0 = 1/\sqrt{\beta_0}(\mathbf{D}\kappa_0^{-\nu} - \kappa_0^{2-\nu}\mathbf{I})$. Observe that $\mathbf{u}_N \sim N(0, \left(\mathbf{L}_0^\top \mathbf{L}_0\right)^{-1})$. Since the hyperprior is stationary, then its covariance matrix is equals to $\mathcal{C}_u = \left(\mathbf{L}_0^\top \mathbf{L}_0\right)^{-1}$, with the m-th diagonal entry given by

$$\mathcal{C}_{u(m)} = \frac{\kappa_0^{2\nu}\beta_0}{\left(\left(c_d m\right)^2 + \kappa_0^2\right)^2}. \tag{10}$$

3.1. Sampling

In this section, we give the algorithm for drawing samples from the posterior distribution of $(\mathbf{v}_N, \mathbf{u}_N)$. The sampling technique has to be carefully designed as problem considered here is originally given in a function space, which is infinite-dimensional. As was the case in the grid partitioning technique, traditional Markov chain Monte Carlo (MCMC) algorithms suffer from slow mixing times upon Fourier-basis refinement, when the approximation in Section 2 becomes more accurate. These methods dictate to reduce the MCMC step size s which becomes computationally expensive. There are several MCMC algorithms designed specifically to deal with an infinite-dimensional problem so that mixing time will, up to some extent, be almost independent with the increase of dimension [23–25]. From the samples of $\mathbf{u}_N$, we can calculate estimates for $\ell^N = \kappa(u^N)^{-1}$. The algorithm is as follows:

1. Initiate $\mathbf{v}_N^{(0)}$ and $\mathbf{u}_N^{(0)}$ and set the value of s.
2. For $k = 1 \ldots K$:
   (a) Update $\mathbf{v}_N^{(k)}$ given fixed $\mathbf{u}_N^{(k-1)}$, draw $\mathbf{e} \sim \mathcal{N}(0, \mathbf{I}_m)$, $\mathbf{w}_N \sim \mathcal{N}(0, \mathbf{I}_{2N+1})$, and sample from the Gaussian posterior of $\mathbf{v}_N$ given $(\mathbf{u}_N, \mathbf{y})$:

$$\mathbf{v}_N^{(k)} = \begin{pmatrix}\mathbf{E}^{-1/2}\mathbf{H} \\ \mathbf{L}(\mathbf{u}_N^{(k-1)})\end{pmatrix}^\dagger \left(\begin{pmatrix}\mathbf{E}^{-1/2}\mathbf{y} \\ 0\end{pmatrix} + \begin{pmatrix}\mathbf{e} \\ \mathbf{w}_N\end{pmatrix}\right),$$

where † denotes the matrix pseudoinverse. Observe that the prior of $\mathbf{v}_N$ is represented here as $\mathbf{L}(\mathbf{u}_N^{(k-1)})\mathbf{v}_N^{(k)} = \mathbf{w}_N$ [10].

(b) Update $\mathbf{u}_N^{(k)}$ using preconditioned Crank-Nicholson algorithm (pCN) [23, 24] given fixed $\mathbf{v}_N^{(k)}$.

i. Draw a candidate sample $\tilde{\mathbf{u}}$ from $q(\cdot|\mathbf{u}_N^{(k-1)})$ using pCN.

$$\tilde{\mathbf{u}} = \sqrt{1-s^2}\mathbf{u}_N^{(k-1)} + s\sigma_{\mathbf{u}} \cdot \tilde{\mathbf{w}}.$$

The vector $\sigma_{\mathbf{u}}$ has entries equal to inverse diagonal elements of $\mathbf{L}_0$.

ii. Accept with probability

$$p = \min\left(1, \tilde{p}\right), \tag{11}$$

$$\tilde{p} = \frac{p\left(\tilde{\mathbf{u}}|\mathbf{v}_N^{(k)}, \mathbf{u}_N^{(k-1)}\right) q\left(\mathbf{u}_N^{(k-1)}|\tilde{\mathbf{u}}\right)}{p\left(\mathbf{u}_N^{(k-1)}|\mathbf{v}_N^{(k)}, \mathbf{u}_N^{(k-1)}\right) q\left(\tilde{\mathbf{u}}|\mathbf{u}_N^{(k-1)}\right)}.$$

iii. If accepted, we set $\mathbf{u}_N^{(k)} = \tilde{\mathbf{u}}$. Otherwise we set $\mathbf{u}_N^{(k)} = \mathbf{u}_N^{(k-1)}$.

We aim at acceptance ratio between 25-50 %, which is obtained by tuning s. Implementing an automatic s correction and parallel computation for algorithm above is also possible [26, 27].

4. NUMERICAL RESULTS

In this section, we present an example of Gaussian process regression with a single layer hyperprior. Implementation of algorithms in this paper is written in Python. To gain better computational speed, some of the routines are implemented via Just in Time (JIT) compiler through Numba package [28]. Parallelization of Monte Carlo computation is implemented via Mpi4py package [29]. To show the ability of the proposed technique to address the non-parametric estimation of piecewise smooth signal, let us construct the following test signal [10].

$$\upsilon(t) = \begin{cases} \exp\left(4 - \frac{1}{2t-4t^2}\right), & x \in (0, 0.5), \\ 1, & x \in [0.7, 0.8], \\ -1, & x \in (0.8, 0.9], \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

The measurement model is,

$$y(t_k) = \upsilon(t_k) + e(k), \tag{13}$$

where, e is a white noise with zero mean and known variance. The measurement matrix $\mathbf{H}$ for (13) is given by

$$\mathbf{H} = \begin{bmatrix} \phi_{-N}(t_0) & \cdots & \phi_N(t_0) \\ \vdots & & \vdots \\ \phi_{-N}(t_m) & \cdots & \phi_N(t_m) \end{bmatrix}.$$

We notice that in the points $\{0.7, 0.8, 0.9\}$, there are discontinuities. Therefore, in order to capture this phenomenon, the length-scale of υ prior should be close to zero at these points. If we select $\ell(u) = \exp(u)$, to allow high variation near points of discontinuities, the length-scale of u should be close to zero as well. We take measurement of (13) on time grid of 2^8 points and set the standard deviation of the measurement noise to be 0.1. The proposed algorithm is tested with a different selection of $N = \{2^4, 2^5, 2^6\}$.

After estimation, we reconstruct the signal using inverse Fourier transform with a finer grid with 2^{10} points equally spaced between zero and one. In this simulation, we use the same measurement record for each run with different N. Figures 3a, 3d, and 3g, show the regression result, where the blue, red, and dotted lines are the sample means in the 95 % confidence shades, Fourier inverse of $\mathbf{v}$, and the original $\upsilon(t)$ signals respectively. We can see generally that the regression result are increasingly better when we increase N. The norms between the reference signal $\upsilon(t)$ in the finer grid and sample mean of posterior sample mean for different N are given by $5.7013, 4.0640$, and 3.2074 respectively. However, with lower Fourier basis $N = 2^4$, the discontinuities are unobservable. Only when N is increased to 2^5, the discontinuities start to appear in the regression result. Furthermore, in Figures 3b, 3e, and 3h we observe that the length-scale ℓ for different simulations is considerably lower near the point of discontinuities, where starting at $N = 2^5$ we can see that there are three different suspect discontinuities just by examining the length-scale. Lastly, Figures 3c, 3f, and 3i show the absolute value of Fourier coefficients, where blue and red lines are the posterior sample mean and the original $\mathbf{v}$. We can see that the lower frequency component of υ is reconstructed considerably close with $N = 2^4$. As expected, increasing the number of the Fourier basis functions, we can gain a closer match in the remaining Fourier components. Notice, however, although increasing N would slowly increase one step simulation time, the number of steps to achieve mixing in MCMC grows at multiple times than that; see. Fig. 2.

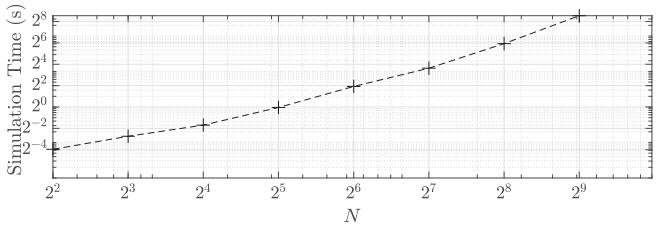
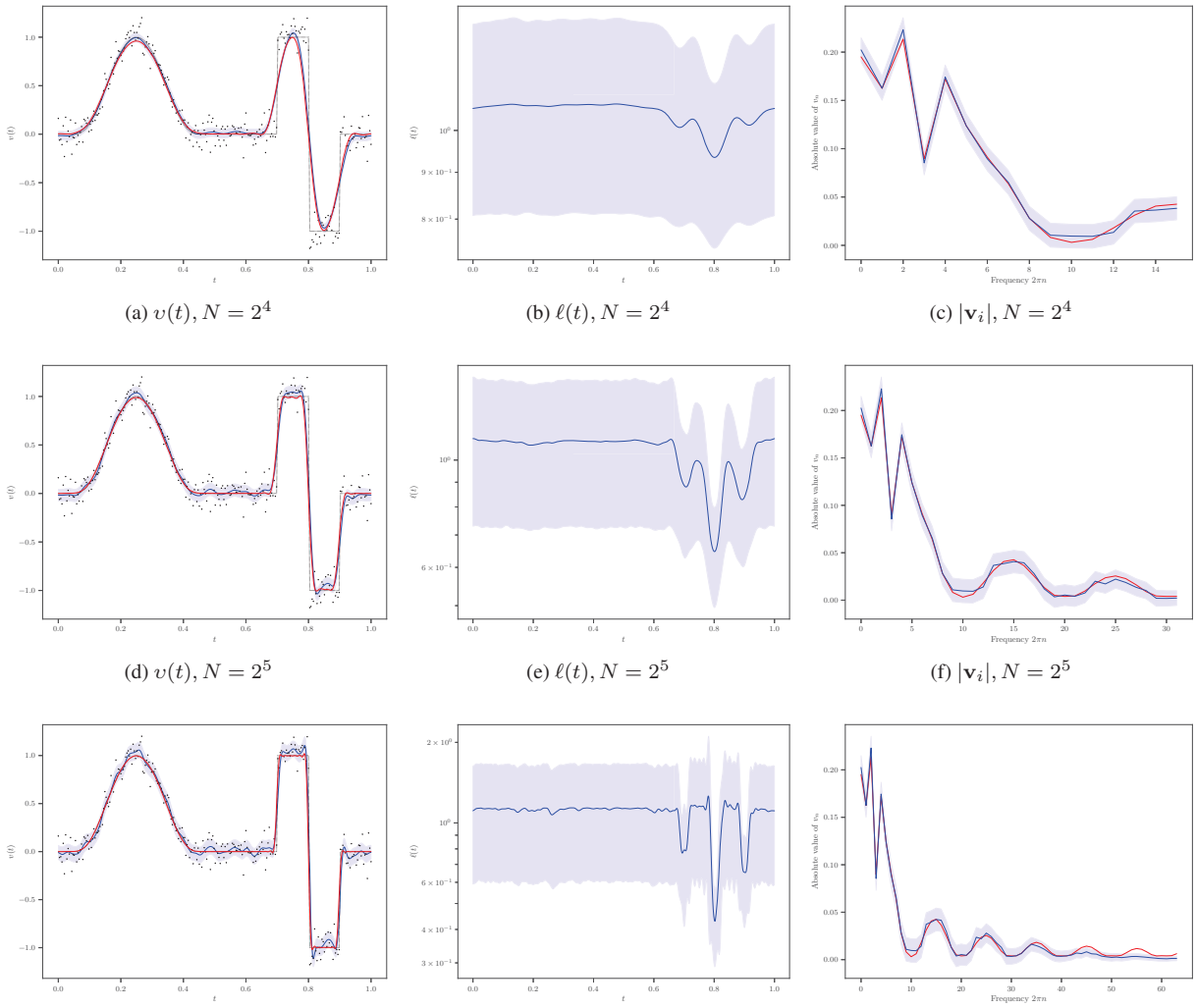


Fig. 2: Execution time of MCMC for 1000 steps for different number of basis functions N.



(a) $\upsilon(t), N = 2^4$ (b) $\ell(t), N = 2^4$ (c) $|\mathbf{v}_i|, N = 2^4$

(d) $\upsilon(t), N = 2^5$ (e) $\ell(t), N = 2^5$ (f) $|\mathbf{v}_i|, N = 2^5$

(g) $\upsilon(t), N = 2^6$ (h) $\ell(t), N = 2^6$ (i) $|\mathbf{v}_i|, N = 2^6$

Fig. 3: Simulation Result of example in Section 4.

5. CONCLUSION

In this paper, we have proposed deep Gaussian processes regression via a Hilbert space expansion. By projecting the underlying SPDEs onto a finite-dimensional space, we can avoid explicitly evaluating SPDEs forward problem. The application of the proposed method on a one-dimensional Gaussian inference problem has shown that the proposed method has an ability to offer simultaneously smooth regression and discontinuity detection. Further work would be needed to investigate the convergence behavior of the regression method upon the number of the basis functions.

6. REFERENCES

[1] D. J. MacKay, "Introduction to Gaussian processes," *NATO ASI F Comp. & Sys. Sci.*, vol. 168, pp. 133–166, 1998.

[2] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, "Combining field data and computer simulations for calibration and prediction," *SIAM J. Sci. Comp.*, vol. 26, no. 2, pp. 448–466, jan 2004.

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press Ltd, 2005.

[4] C. J. Paciorek, "Non-stationary Gaussian processes for regression and spatial modelling," Ph.D. dissertation, 2003.

[5] C. J. Paciorek and M. J. Schervish, "Nonstationary covariance functions for Gaussian process regression," in *Adv. Neur. Inf. Proc. Sys.*, 2004, pp. 273–280.

[6] M. J. Heaton, A. Datta, A. Finley, R. Furrer, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D. Hammerling, M. Katzfuss, F. Lindgren, D. W. Nychka, F. Sun, and A. Zammit-Mangion, "A case study competition among methods for analyzing large spatial data," *arXiv:1710.0501*.

[7] G.-A. Fuglstad, D. Simpson, F. Lindgren, and H. Rue, "Does non-stationary spatial data always require non-stationary random fields?" *Spat. Stat.*, vol. 14, pp. 505–531, nov 2015.

[8] A. M. Schmidt and A. O. Hagan, "Bayesian inference for non-stationary spatial covariance structure via spatial deformations," *J. Royal Stat. Soc. B*, vol. 65, no. 3, pp. 743–758, aug 2003.

[9] E. Snelson, Z. Ghahramani, and C. E. Rasmussen, "Warped Gaussian processes," in *Adv. Neur. Inf. Proc. Sys.*, 2004, pp. 337–344.

[10] L. Roininen, M. Girolami, S. Lasanen, and M. Markkanen, "Hyperpriors for Matérn fields with applications in Bayesian inversion," *Inve. Probl. & Imag.*, vol. 13, no. 1, pp. 1–29, 2019.

[11] M. M. Dunlop, M. A. Girolami, A. M. Stuart, and A. L. Teckentrup, "How deep are deep Gaussian processes?" *J. Mach. Learn. Res.*, vol. 19, no. 54, pp. 1–46, 2018.

[12] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*. Springer, 2004.

[13] A. M. Stuart, "Inverse problems: A Bayesian perspective," *Acta Numerica*, vol. 19, pp. 451–559, 2010.

[14] M. Dashti and A. M. Stuart, "The Bayesian approach to inverse problems," in *Handbook of Uncertainty Quantification*. Springer International Publishing, 2017, pp. 311–428.

[15] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *arXiv:1401.5508*, 2019.

[16] R. J. Adler, *The Geometry of Random Fields*. Society for Industrial and Applied Mathematics, Philadelphia, 2010.

[17] I. I. Gikhman and A. V. Skhorokhod, *The Theory of stochastic processes*. Springer-Verlag, 2004.

[18] F. Lindgren, H. Rue, and J. Lindström, "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach," *J. Royal Stat. Soc. B*, vol. 73, no. 4, pp. 423–498, 2011.

[19] S. Lasanen, L. Roininen, and J. M. Huttunen, "Elliptic boundary value problems with Gaussian white noise loads," *Stoch. Proc. & Appl.*, vol. 128, no. 11, pp. 3607–3627, 2018.

[20] L. C. Evans, *Partial Differential Equations*. American Mathematical Society, 2010.

[21] F. P. Greenleaf and S. Marques, *Linear Algebra*, ser. Courant Lecture Notes in Mathematics. American Mathematical Society, 2019.

[22] E. Brigham, *Fast Fourier Transform and Its Applications*. Pearson, 1988.

[23] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White, "MCMC methods for functions: Modifying old algorithms to make them faster," *Statist. Sci.*, vol. 28, no. 3, pp. 424–446, 2013.

[24] K. J. H. Law, "Proposals which speed up function-space MCMC," *J. Comp. & Appl. Math.*, vol. 262, pp. 127–138, 2014.

[25] A. Beskos, M. Girolami, S. Lan, P. E. Farrell, and A. M. Stuart, "Geometric MCMC for infinite-dimensional inverse problems," *J. Comp. Phys*, vol. 335, pp. 327–351, 2017.

[26] Y. Chen, D. Keyes, K. J. H. Law, and H. Ltaief, "Accelerated dimension-independent adaptive Metropolis," *SIAM J. Sci. Comp.*, vol. 38, no. 5, pp. S539–S565, 2016.

[27] D. J. Wilkinson, *Parallel Bayesian Computation*. Chapman & Hall/ CRC, 2006.

[28] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: a LLVM-based Python JIT compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*. ACM Press, 2015.

[29] L. Dalcín, R. Paz, M. Storti, and J. D'Elía, "MPI for Python: Performance improvements and MPI-2 extensions," *J. Par & Dist. Comp.*, vol. 68, no. 5, pp. 655–662, 2008.