On the impact of prior distributions on efficiency of sparse Gaussian process regression

Esmaeilbeigi, M., Chatrabgoun, O., Daneshkhah, A. & Shafa, M

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink: Esmaeilbeigi, M., Chatrabgoun, O., Daneshkhah, A. et al. On the impact of prior distributions on efficiency of sparse Gaussian process regression. Engineering with Computers (2022). https://doi.org/10.1007/s00366-022-01686-7

DOI 10.1007/s00366-022-01686-7
ISSN 0177-0667
ESSN 1435-5663

Publisher: Springer

The final publication is available at Springer via http://dx.doi.org/10.1007/s00366-022-01686-7

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

On the impact of prior distributions on efficiency of sparse Gaussian process regression

Mohsen Esmaeilbeigi[a,*], Omid Chatrabgoun[a,b], Alireza Daneshkhah[c], Maryam Shafa[a]

[a] *Faculty of Mathematical Sciences and Statistics, Malayer University, Malayer, 65719-95863, Iran*
[b] *School of Computing, Electronics and Mathematics, Coventry University, Coventry, CV1 5FB, United Kingdom*
[c] *Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry, CV1 5FB, United Kingdom*

Abstract

Gaussian process regression (GPR) is a kernel-based learning model, which unfortunately suffers from computational intractability for irregular domain and large datasets due to the full kernel matrix. In this paper, we propose a novel method to produce a sparse kernel matrix by using the compact support radial kernels (CSRKs) to efficiently learn the GPR from large datasets. The CSRKs can effectively avoid the ill-conditioned and full kernel matrix during GPR training and prediction, consequently reducing computational costs and memory requirements. In practice, the interest in CSRKs waned slightly as it became evident that, there is a trade-off principle (conflict between accuracy and sparsity) for compactly supported kernels. Hence, when using kernels with compact support, during GPR training, the main focus will be on providing a high level of accuracy. In this case, the advantage of achieving a sparse covariance matrix for CSRKs will almost disappear, as we will see in the numerical results. This trade-off has led authors to search for an "optimal" value of the scale parameter. Accordingly, by selecting the suitable priors on the kernel hyperparameters, and simply estimating the hyperparameters using a modified version of the maximum likelihood estimation (MLE), the GPR model derived from the CSRKs yields maximal accuracy while still maintaining a sparse covariance matrix. In fact, in GPR training, modified version of the MLE will be proportional to the product of MLE and a given suitable prior distribution for the hyperparameters that provides an efficient method for learning. The misspecification of prior distributions and their impact on the predictability of the sparse GPR models are also comprehensively investigated using several empirical studies. The proposed new approach is applied to some irregular domains with noisy test functions in 2D data sets in a comparative study. We finally investigate the effect of prior on the predictability of GPR models based on the real dataset. The derived results suggest the proposed method leads to more sparsity and well-conditioned kernel matrices in all cases.

Keywords:
Compact support radial kernels, Gaussian process, Hyperparameter, Maximum likelihood estimation, Priors.

1. Introduction

Gaussian Processes (GP) is a generic supervised learning method designed to solve a wide range of probabilistic Machine Learning, including classification, regression, probabilistic forecasting, uncertainty quantification, etc [1, 2, 3, 4]. The Gaussian processes regression (GPR) has been proven to be a powerful and effective method for non-linear regression problems due to many desirable properties such as simple to

*Corresponding author
Email addresses: m.esmaeilbeigi@malayeru.ac.ir (Mohsen Esmaeilbeigi), o.chatrabgoun@malayeru.ac.ir (Omid Chatrabgoun), ali.daneshkhah@coventry.ac.uk (Alireza Daneshkhah), maryam.shafa@stu.malayeru.ac.ir (Maryam Shafa)

implement, flexibility and fully probabilistic models [5, 6, 7, 8]. Unlike other kernel-based approaches such as support vector machines (SVMs) [9, 10], GPR can quantify the uncertainty of predictions due to their explicit probabilistic formulation. In addition, GPR has been proposed as an appropriate alternative for supervised neural networks in non-linear regression and classification [11].

GPR as a kernel-based non-parametric method, is hugely relied on selecting the appropriate kernel function [12], and efficiently estimating its hyperparameters. The kernel functions represent our assumptions about the function we wish to learn and define the closeness and similarity between data points [7]. In other words, GPR inherits the existed properties in the used kernels [6]. There are many different kernel types with various features and properties which one can be selected depending on the purposes of study [13]. For instance, the radial kernel is more suitable for predicting the future. This kernel is a real valued function whose output depends exclusively on the distance of its input from some origins. There are two general scenarios for selecting the radial kernels: global and compact support. Since all the training data is required for prediction the future using GPR, global support radial kernels (GSRKs) have some computational limitations due to producing a full and dense presentation for ill-conditioned kernel (or kernel matrix). The size of dense kernel matrix would then increase quadratically with the size of training set. Moreover, updating the Bayesian posterior distribution given the new data-points is also computationally expensive. These computational burdens are due to this fact that the kernel matrix derived using GSRKs is commonly computationally expensive to invert, and thus to use for prediction tasks. To cope with these issues, many researchers have been proposed different ways on using sparse kernel matrix to speed up the computations and reduce the memory requirements [14, 15, 16, 17, 18, 19]. In all of these strategies to increase the computational speed and to reduce memory requirements, lack of modelling accuracy can be seen. One of these methods which is not very relevant to the rest creates a down-dates kernel matrix by removing certain rows and columns corresponding to the discarded training data set [18]. This model is not very efficient, since valuable information will be lost by removing certain rows and columns. Other methods make structural assumptions, such as assuming the kernel matrix to be block diagonal, whence the GPR can be decomposed into a number of smaller GPR [20]. Another algorithm to yield sparse GPR is proposed in [21] which is based on updating the Cholesky factor of the kernel matrix. In this paper, we propose a novel method to efficiently derive sparse kernel matrix by using *compact support radial kernels (CSRKs)*. We then demonstrate the nice properties of these kernels and how they can improve efficiency of the corresponding GPR for a wide range of applications including the problems involved with high dimensional data and complicated geometries which would be very challenging under the GSRK. It should also noted that the proposed method in this paper unlike the other methods mentioned above directly deals with the kernel matrix itself in a very easy to implement manner, which turns out to be also well-conditioned. Once a radial kernel such as CSRK is selected for a GPR, the unknown hyperparameters of the kernel need to be estimated from the training data [6]. However, the Monte Carlo (MC) methods can be used to implement GPR even without estimating hyperparameters as discussed in [22, 23, 24, 25], but the common approach is to estimate the hyperparameters by means of MLE [11, 26, 27, 28] due to the high computational cost of MC methods. The MLE makes optimal use of the information contained in the data [29]. In addition, a trade-off between the model complexity and model fit can be automatically incorporated due to a nice property of the marginal likelihood [30].

If we use the CSRKs in GPR, then the main difference to the global support structure is that now the kernel matrix can be made sparse by scaling the support of the CSRK appropriately. In fact, only the entries in the kernel matrix corresponding to nodes lying closer than λ (whose support is $[0; \lambda]$, with $\lambda = 1/\epsilon$) to a given CSRK center are non-zero, leading to a sparse kernel matrix. Indeed, the ϵ determines the size of the support and, consequently, the sparsity in the kernel matrix. In practice, the interest in CSRKs waned slightly as it became evident that, to obtain good accuracy, the ϵ is decreased such that the overlap distance λ should cover most nodes in the point set. So, there is a trade-off principle (conflict between accuracy and sparsity) for CSRKs. Hence, when using CSRKs, during GPR training, the main focus will be on providing a high level of accuracy. In this case, the advantage of achieving a sparse covariance matrix for CSRKs will almost disappear, as we will see in the numerical results. This trade-off has led authors to search for an "optimal" value of the scale parameter ϵ, i.e., a value that yields maximal accuracy while still maintaining a sparse covariance matrix. Accordingly, by selecting a suitable prior on the kernel hyperparameters, and

simply estimating the hyperparameters using a modified version of the MLE, during GPR training, in addition to providing the desired accuracy, an appropriate quantity of sparsity will also be obtained. This modified version of the MLE will be proportional to the product of MLE and a given prior distribution for the hyperparameters. Here, the MLE ensures that the obtained model has maximum agreement with the training data, and the suitable prior distribution guarantees the achievement of a sparse covariance matrix. Accordingly, the final sparse kernel matrix by CSRKs can be efficiently derived from the modified version of MLE, and available data. We will investigate and analyze this subject well in the numerical results. It is therefore interesting to know how prior distributions for hyperparameters affect the performance of GPR-based CSRKs.

As a summary, in this paper, we propose a novel method to produce a sparse kernel matrix by using the compact support radial kernels (CSRKs) to efficiently learn the GPR from large datasets. Then, we provide the first empirical study of the impact of the prior distributions on the hyperparameter estimation and the performance of GPR-based CSRKs, for some commonly used kernels in GPR modeling.

It should be noted that, selecting the best prior or best kernel is not subject of this paper and as the results will show, once a CSRK is chosen, the impact of different priors for the initial hyperparameters on the performance of GPR prediction is investigated. However, this paper aims to demonstrate the capabilities and improved efficiency of the prior-CSRKs method (GPR-based CSRKs by a prior distribution on hyperparameters) presented in this paper for GPR problems with large datasets and complicated geometries.

The layout of the article is as follows. In Section 2, we briefly review GPR model, and the MLE method to estimate the hyperparameters. In Section 3, we attempt to compare the main nature of the GSRKs and CSRKs. Then, we explain prior-MLE or modified version of MLE to estimate hyperparameter of the GP model with CSRKs in Section 4. Section 5 provides an overview of the computational complexity of Cholesky decomposition for sparse kernel matrices. Section 6 presents the results of our extended numerical experiments concerning GPR-based GSRKs and CSRKs. In addition, in Section 7, we investigate the influence of prior on the predictability of GPR models based on the real dataset. Finally, Section 8 is devoted to brief conclusions.

2. Gaussian process regression review

A GP is a collection of random variables with the property that the joint distribution of any of its subset is consistent joint Gaussian distribution. In fact, for $N \in \mathbb{N}$ and $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$, the vector of random variables $f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), ..., f(\boldsymbol{x}_N)$ is (multivariate) Gaussian. As a Gaussian distribution is specified by mean and covariance, a GP is also completely defined by mean and covariance function or kernel. Therefore, for any mean function μ and kernel K, there exists a GP $f(\boldsymbol{x})$ such that $\mathbb{E}[f(\boldsymbol{x})] = \mu(\boldsymbol{x})$ and $Cov(f(\boldsymbol{x}_s), f(\boldsymbol{x}_t)) = K(\boldsymbol{x}_s, \boldsymbol{x}_t)$. It denotes by $f \sim GP(\mu, K)$.
To define a probabilistic regression based on GP given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i), i = 1, \cdots, N\}$ of N pairs of (vectorial) inputs $\boldsymbol{x}_i$ and noisy (real, scalar) outputs y_i, consider a regression problem

$$y_i = f(\boldsymbol{x}_i) + \zeta_i, \qquad where \qquad \zeta_i \sim \mathcal{N}\left(0, \sigma_{\boldsymbol{\zeta}}^2\right),$$

it yields that the collection of functions $\{f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_N)\}$ follow a multivariate Gaussian distribution such that

$$\boldsymbol{f} = [f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_N)]^T \sim \mathcal{N}(\mu, \mathsf{K}),$$

where $\mu = [\mu(x_1), ..., \mu(x_N)]^T$ is the mean vector and K is the $N \times N$ covariance matrix of which the (i, j)-th element $\mathsf{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Also, we can compute the predictive distribution of the function values $f(\boldsymbol{x}^*)$ (hereafter f^*) or noisy y^* at test locations $\boldsymbol{x}^*$. To predict the function values $\boldsymbol{f}^* = [f_1^*, ..., f_M^*]^T$ at the test locations $\boldsymbol{X}^* = [\boldsymbol{x}_1^*, ..., \boldsymbol{x}_M^*]^T$, the joint distribution of training observations $\boldsymbol{y}$ and predictive targets $\boldsymbol{f}^*$ are given by

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\boldsymbol{X}) \\ \mu(\boldsymbol{X}^*) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_{\boldsymbol{\zeta}}^2 I & K(\boldsymbol{X}^*, \boldsymbol{X}) \\ K(\boldsymbol{X}^*, \boldsymbol{X}) & K(\boldsymbol{X}^*, \boldsymbol{X}^*) \end{bmatrix} \right),$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^T$, $\mu(\boldsymbol{X}) = \mu, \mu(\boldsymbol{X}^*) = [\mu(\boldsymbol{x}_1^*), ..., \mu(\boldsymbol{x}_M^*)]^T$, $K(\boldsymbol{X}, \boldsymbol{X}) = \mathsf{K}$, $K(\boldsymbol{X}^*, \boldsymbol{X}) = \mathsf{K}_*$ is an $M \times N$ matrix of which the (i, j)-th element $[K(\boldsymbol{X}^*, \boldsymbol{X})]_{ij} = K(\boldsymbol{x}_i^*, \boldsymbol{x}_j)$, and $K(\boldsymbol{X}^*, \boldsymbol{X}^*) = \mathsf{K}_{**}$ is an $M \times M$

matrix with the $(i,j)-$th element $[K(\boldsymbol{X}^*,\boldsymbol{X}^*)]_{ij}=K(\boldsymbol{x}_i^*,\boldsymbol{x}_j^*)$ and I is the identity matrix. In GPR method the mean function $\mu(\boldsymbol{X})$ is often assumed to be 0. Thus the predictive distribution is

$$p(\boldsymbol{f}_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{X}^*)=\mathcal{N}(\mathsf{K}_*^T(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)^{-1}\boldsymbol{y},\ \mathsf{K}_{**}-\mathsf{K}_*^T(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)^{-1}\mathsf{K}_*). \tag{1}$$

In fact, the main task in regression is to use the values $y_1,\cdots,y_N$ sampled at locations $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N$ in order to predict the unknown value f^* at a location $\boldsymbol{x}^*$ whose entries are given by the kernel [7]. Also, one can produce confidence intervals of the standard form

$$p(\boldsymbol{f}^*\in \mathsf{K}_*^T(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)^{-1}\boldsymbol{y}\pm z_{\frac{\alpha}{2}}\sqrt{\mathsf{K}_{**}-\mathsf{K}_*^T(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)^{-1}\mathsf{K}_*})=1-\alpha,$$

where $z_{\frac{\alpha}{2}}=F^{-1}(\frac{\alpha}{2})=-F^{-1}(1-\frac{\alpha}{2})$ where $100(1-\alpha)\%$ is the confidence level and F is the cumulative distribution function of the standard normal distribution, used as the critical value. This value is only dependent on the confidence level for the test. The valid kernels give rise to positive semi-definite covariance matrices. The kernel encodes our assumptions about the function we wish to learn, by defining a notion of similarity between two function values, as a function of the corresponding two inputs. Therefore, the kernel plays a crucial role in the predictive mean and variance. The kernels contain our presumptions about the function we wish to learn and define the closeness and similarity between data points. Therefore, the choice of kernel has a profound impact on the performance of a GPR model, just as activation function, learning rate can affect the result of a neural network [12]. In fact, GPR inherits the existed properties in the used kernels. Many different types of kernels were featured in [13] such as radial, translation invariant and series kernels. In GPR, the radial kernel is a popular kernel which is used for making predictions. A radial kernel is a real valued function whose output depends exclusively on the distance of its input from some origins [31]. There are two general scenarios for selecting these radial kernels: global and compact support kernels. In the sequel, we show that the GPR method based on GSRKs is not useful for irregular domain and large data sets. The computational complexities for irregular domain and large data sets are produced by full and dense kernel matrix, and also to invert kernel matrix and to use in prediction. To cope with these issues, we investigate to show that the GPR based on CSRKs can produce a sparse kernel matrix versus a dense kernel matrix.

The kernels usually contain unknown hyperparameters (such as the length-scale, signal variance, and noise variance) need to be inferred from the data in GPR models. As the posterior distribution over the hyperparameters is generally difficult to obtain, full Bayesian inference of the hyperparameters is generally not used. There are recent efforts to make hyperparameter estimation fully Bayesian which are promising to result in more robust estimates by additionally providing uncertainty estimates for the obtained hyperparameters [32]. One common criterion for estimating the hyperparameters in GPR is usually computed by MLE. In other words, MLE maximizes the agreement between the observed data and the model. Following the GP assumption, the MLE is given as

$$p(\boldsymbol{y}|\,\boldsymbol{X},\epsilon)=\mathcal{N}(0,\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I),$$

where ϵ is the collection of the unknown hyperparameters. Therefore, the negative log MLE is

$$\mathcal{L}(\epsilon)=-\log p(\boldsymbol{y}|\,\boldsymbol{X},\epsilon)=\frac{1}{2}\boldsymbol{y}^T(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)^{-1}\boldsymbol{y}+\frac{1}{2}\log\ \mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I\ +\frac{N}{2}\log 2\pi. \tag{2}$$

To set the hyperparameters by minimizing (2), we seek gradient the partial derivatives of (2) w.r.t. the hyperparameters:

$$\begin{aligned}\tfrac{\partial}{\partial\epsilon}\mathcal{L}(\epsilon)=&-\tfrac{1}{2}\boldsymbol{y}^T(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)^{-1}\tfrac{\partial(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)}{\partial\epsilon}(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)^{-1}\boldsymbol{y}\\&+\tfrac{1}{2}tr\left(\left(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I\right)^{-1}\tfrac{\partial(\mathsf{K}+\sigma_{\boldsymbol{\zeta}}^2 I)}{\partial\epsilon}\right).\end{aligned}$$

Here, (2) can normally be used through a numerical optimization algorithm in MATLAB such as Conjugate Gradient to find optimal hyperparameter.

3. GPR-based compact support radial kernels

From the equation (1), the kernel plays an important role in the predictive mean and covariance. The choice of kernel is based on assumptions such as smoothness and likely patterns to be expected in the data [7]. In fact, the GPR models are non-parametric kernel-based probabilistic models, where the choice of kernel has a profound impact on the performance of a GPR model. The main advantages of kernel-based methods lie in their simplicity and their effectiveness in dealing with high-dimensional problems with complicated geometries since no mesh generation is needed. Here based on the [13], we consider a domain $\Omega \subseteq \mathbb{R}^d$, and call a function $K : \Omega \times \Omega \longrightarrow \mathbb{R}$ as a kernel. Also, a kernel K is symmetric, if $K(\boldsymbol{x}, \boldsymbol{z}) = K(\boldsymbol{z}, \boldsymbol{x})$ holds for all $\boldsymbol{x}, \boldsymbol{z} \in \Omega^d$. A kernel $K : \Omega \times \Omega \longrightarrow \mathbb{R}$ is called to be radial if there exists a univariate function $k : [0, \infty) \longrightarrow \mathbb{R}$ such that $K(\boldsymbol{x}, \boldsymbol{z}) = k(r)$, where $r = ||\boldsymbol{x} - \boldsymbol{z}||$ and $||\cdot||$ denotes the Euclidean distance. In Table 1 we report a list of some well-known radial kernels with their orders of smoothness and support, where ϵ a positive constant which is known as the shape parameter, $(\cdot)_+$ denotes the truncated power function, and $\nu \in \mathbb{N}$ [33].

Table 1: Examples of some popular radial kernels

Radial kernel	$\boldsymbol{k(r)}$	**Support**
Gaussian C^∞ (GA)	$e^{-\epsilon^2 r^2}$	GSRK
MultiQuadric C^∞ (MQ)	$(1+\epsilon^2 r^2)^{1/2}$	GSRK
Inverse MultiQuadric C^∞ (IMQ)	$(1+\epsilon^2 r^2)^{-1/2}$	GSRK
Thin Plate Spline $C^{\nu+1}$ (TPS)	$(-1)^{\nu+1} r^{2\nu} \log r$	GSRK
Matérn C^4 (M4)	$e^{-\epsilon r}(\epsilon^2 r^2 + 3\epsilon r + 3)$	GSRK
Matérn C^2 (M2)	$e^{-\epsilon r}(\epsilon r + 1)$	GSRK
Wendland C^6 (W6)	$(1-\epsilon r)_+^8 (32\epsilon^3 r^3 + 25\epsilon^2 r^2 + 8\epsilon r + 1)$	CSRK
Wendland C^4 (W4)	$(1-\epsilon r)_+^6 (35\epsilon^2 r^2 + 18\epsilon r + 3)$	CSRK
Wendland C^2 (W2)	$(1-\epsilon r)_+^4 (4\epsilon r + 1)$	CSRK

There are various radial kernels which are suitable for GPR. They can be divided in two groups: GSRKs and CSRKs. The traditional kernels are GSRKs such as MQs, IMQs and GAs. Since all the training data is required for making predictions in GPR, GSRKs have some computational limitations such that they produce full and dense kernel matrix which is ill-conditioned. We will clearly show the effects the size of training set has on the condition number (CN) of kernel matrix (and therefore the numerical stability) of our computations. Therefore, in Figure 1 we display CNs of kernel matrix as a function of the size of training set by using $M4$ kernel. In fact, the size of dense kernel matrix increases quadratically with the size of training

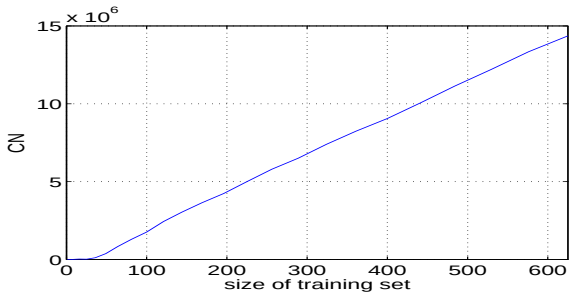


Figure 1: The effects of the size of the training set on the condition number of kernel matrix when using M4.

set with used GSRKs. The computational difficulties are generated by the Bayesian posterior update to incorporate data, and also to inverting of kernel matrix, and per test case for prediction. To cope with these issues, we show that the GPR based on CSRKs can produce sparse kernel matrix which is in contrast dense kernel matrix. The CSRKs can result in a sparse kernel matrix and effectively avoids the ill-conditioned and dense matrix and consequently reduces computational costs. As a simple consequence of a theorem of Schoenberg, a CSRK can not be positive definite for all space dimensions [34]. The CSRKs can be strictly positive on $\mathbb{R}^d$ only for a fixed maximal dimension d [38]. Therefore authors focus their attention on the characterization and construction of functions that are compactly supported, strictly positive definite and radial on $\mathbb{R}^d$ for some fixed dimension d.

Wendland constructs, by dimension walk, a popular family of CSRKs, expressed with a piecewise polynomial form whose degree is minimal for a given dimension space d and whose continuity is C^{2n} [35]. Wendland defines a certain integral operator $\mathcal{I}$ and show how they facilitate the construction of CSRKs.

Definition 1. *Let ϕ be such that $t \longmapsto t\phi(t) \in L_1[0,\infty)$. Then we define the integral operator $\mathcal{I}$ via*

$$(\mathcal{I}\phi)(r) = \int_r^\infty t\phi(t)dt, \quad r \geq 0.$$

The resulting function is to be interpreted as even function using even extension.

Theorem 1. *Suppose that ϕ is continuous.*

(1) $\mathcal{I}$ preserve compact support, i.e., if ϕ has compact support, then so do $\mathcal{I}\phi$.

(2) If $t \longmapsto t^{d-1}\phi(t) \in L_1[0,\infty)$ and $d \geq 3$, then $\mathcal{F}_d(\phi) = \mathcal{F}_{d-2}(\mathcal{I}\phi)$.

The operator $\mathcal{I}$ allows us to express d-variate Fourier transform as $(d-2)$-variate Fourier transform, respectively. Since according to Bochner's theorem and generalizations thereof, positive definite and radial kernels on $\mathbb{R}^d$ are characterized by a nonnegative d-variate Fourier transform we can draw the following conclusion.

Corollary 1. *Suppose $\phi \in C(\mathbb{R})$. If $t \longmapsto t^{d-1}\phi(t) \in L_1[0,\infty)$ and $d \geq 3$, then ϕ is strictly positive definite and radial on $\mathbb{R}^d$ if and only if $\mathcal{I}\phi$ is strictly positive definite and radial on $\mathbb{R}^{d-2}$.*

Proof. This follows immediately from the preceding theorem and Bochner's characterization for radial and integrable functions. □

This allows us to construct new strictly positive definite radial functions from given ones by a "dimension-walk" technique that steps through multivariate Euclidean space in even increments.

For example, Wendland constructed the most popular family of CSRKs by this technique. Wendland starts with the truncated power function

$$\phi_l(r) = (1-r)_+^l,$$

which we know to be strictly positive definite and radial on $\mathbb{R}^d$ for $l \geq \lfloor d/2 \rfloor + 1$ [38]. Then he walks through dimensions by repeatedly applying the integral operator $\mathcal{I}$.

With $\phi_l(r) = (1-r)_+^l$, we define Wendland CSRKs $\phi_{d,k}$ such that $\phi_{d,k} = \mathcal{I}^K \phi_{\lfloor d/2 \rfloor + k + 1}$. It turns out that the functions $\phi_{d,k}$ are all supported on $[0,1]$ and have a polynomial representation there. More precisely, Wendland CSRKs $\phi_{d,k}$ are positive definite on $\mathbb{R}^{d_0}$ for $d_0 \leq d$ and are of the form

$$\phi_{d,k} = \begin{cases} p_{d,k}(r), & 0 \leq r \leq 1, \\ 0, & r > 1, \end{cases}$$

with a univariate polynomial $p_{d,k}$ of degree $\lfloor d/2 \rfloor + 3k + 1$. They possess continuous derivatives up to order $2k$. Whereas, Wu presents another way, by convolution, to construct similar CSRK, but provides higher polynomial degree for a prescribed smoothness and dimension [36]. There are many other ways in which

one can construct compactly supported functions that are strictly positive definite and radial on $\mathbb{R}^d$. In [37] several such possibilities are described. However, Wenldland's CSRK are used in this paper because other compactly supported polynomial function that globally C^{2n} and strictly positive definite and radial on $\mathbb{R}^d$ will not have a smaller polynomial degree [38]. In this paper, the notation $\phi_{d,k}(r)$ for Wendland CSRKs is replaced by $k(r)$. Some of the most important and widely used types of Wendland's CSRKs are W2, W4 and W6, which are introduced in Table 1.

If we use the Wendland's CSRKs in GPR then the main difference to the global support structure is that now the kernel matrix can be made sparse by scaling the support of the CSRK appropriately. In fact, only the entries in the kernel matrix corresponding to nodes lying closer than λ (whose support is $[0;\lambda]$, with $\lambda = 1/\epsilon$) to a given CSRK center are non-zero, leading to a sparse kernel matrix. Indeed, the ϵ determines the size of the support and, consequently, the sparsity in kernel matrix. In practice, the interest in CSRKs waned slightly as it became evident that, in order to obtain a good accuracy, the ϵ is decreased such that the overlap distance λ should covered most nodes in the point set. So, there is a trade-off principle (conflict between accuracy and sparsity) for CSRKs. Hence, when using CSRKs (without priors on the hyperparameters), during GPR training, the main focus will be on providing a high level of accuracy. In this case, the advantage of achieving a sparse covariance matrix for CSRKs will almost disappear, as we will see in the numerical results.

This trade-off has led authors to search for an "optimal" value of the scale parameter ϵ, i.e., a value that yields maximal accuracy, while still maintaining a sparse covariance matrix. Accordingly, by selecting a suitable prior on the kernel hyperparameters, and simply estimating the hyperparameters using a modified version of the MLE, during GPR training, in addition to providing the desired accuracy, an appropriate quantity of sparsity will also be obtained. We will investigate and analyze well this subject in the numerical results and we will deal with it as a novelty of this paper.

4. Estimating hyperparameters in CSRKs-GPR by a modified version of MLE

One of the issues that affect the sparsity of the kernel matrix and accuracy of the GPR based on CSRKs (CSRKs-GPR) is the estimated hyperparameters in the used CSRKs. As mentioned earlier, the scale parameter ϵ of CSRK is the hyperparameter should be specified during CSRKs-GPR training.

According to the structure of CSRKs-GPR, depending on the support of the CSRKs, many of the entries in the kernel matrix will be zero. Indeed, the shape parameter determines the size of the support and, consequently, the sparsity in the kernel matrix. Since increasing the size of the support in order to achieve higher accuracy reduces the sparsity of the kernel matrix, there will always be a trade-off principle (conflict between accuracy and sparsity) for CSRKs. In this paper, we propose a novel method to yields maximal accuracy, while still maintaining a sparse covariance matrix by using CSRKs.

In CSRKs-GPR models, the hyperparameters involved in the CSRK need to be estimated from the training data. Although, MC methods can perform GPR without the need of estimating hyperparameters, the common approach is to estimate the hyperparameters by means of MLE due to the high computational cost of MC methods. Therefore, the hyperparameters in CSRKs-GPR with a specified CSRKs are often estimated from the data via MLE. Hence, as mentioned, during CSRKs-GPR training by minimizing (2), the main focus will be on providing a high level of accuracy (targets fitting the training data). In this case, the advantage of achieving a sparse covariance matrix for CSRKs will almost disappear, as we will see in the numerical results.

In Bayesian inference, a prior distribution of an uncertain quantity is the distribution that would express one is beliefs about this quantity before some evidence is taken into account. Accordingly, by selecting a suitable prior on the kernel hyperparameters, and simply estimating the hyperparameters using a modified version of the MLE, the GPR model derived from the CSRKs yields maximal accuracy while still maintaining a sparse covariance matrix. It is usually assumed that the covariance function of a GP belongs to a certain parametric family [7] whose hyperparameters need to be estimated from data. If the hyperparameters are specified by a certain prior distribution, then the modified version of MLE by prior distribution on hyperparameters (prior-MLE) will be proportional to the product of marginal likelihood depending on the

hyperparameters of the covariance function and a given prior distribution. Here, the MLE ensures that the obtained model has maximum agreement with the training data, and the suitable prior distribution guarantees the achievement of a sparse covariance matrix.

The prior distributions that can be considered include non-informative [12] and informative [39] priors. In the cases when there is little information about the data, vague prior distributions can be selected with the intention that they should have a slight influence on the inferences. However, with a small amount of data, the use of non-informative prior may be problematic, and a vague prior distribution may be misleading on any inference made because the results are easily sensitive to the selection of prior distributions. Therefore, by incorporating some information inferred from training data, we have listed different informative prior distributions like Gamma and Gaussian priors which have been discussed in our study.

Now, assume that we are also given a certain prior distribution $p(\epsilon)$ on the hyperparameter ϵ. Then the prior-MLE is usually written as

$$p(\,\epsilon|\,\boldsymbol{X},\boldsymbol{y}) \propto e^{-\mathcal{L}(\epsilon)}p(\epsilon).$$

Therefore, the improved function in the modified version of MLE to be optimized will be as follows:

$$\mathcal{L}^{mod}(\epsilon) = -\log p(\,\epsilon|\,\boldsymbol{X},\boldsymbol{y}) = \mathcal{L}(\epsilon) - \log p(\epsilon), \tag{3}$$

Here, (3) can normally be used through a numerical optimization algorithm such as Conjugate Gradient to find optimal hyperparameter. The procedure for hyperparameter estimation is described below.

1. Randomly choose an initial hyperparameter ϵ_0 from certain prior distribution $p(\epsilon)$.
2. Numerically minimize $\mathcal{L}^{mod}(\epsilon)$ in (3) using ϵ_0 as the starting value and obtain an estimate of the hyperparameter.

As mentioned earlier in the introduction, selecting the best prior or best kernel is not subject of this paper and as the results will show, once a CSRK is chosen, the impact of different priors for the initial hyperparameters on the performance of GPR prediction is investigated. However, this paper aims to demonstrate the capabilities and improved efficiency of the prior-CSRKs method presented in this paper for GPR problems with large datasets and complicated geometries. We will conclude how prior distributions for hyperparameters affect the performance of GPR-based CSRKs such that kernel matrix is more sparse and well-conditioned. We will investigate the influence of the modified version of MLE on the hyperparameters in sparsity structure of kernel matrix for GPR-based GSRKs problems with high dimensional and complicated geometries.

5. Computational complexity review

In this paper, we propose a novel method to produce a sparse kernel matrix by using CSRKs to efficiently learn the GPR from large datasets. In practical implementation of GPR, Cholesky decomposition is usually used instead of directly inverting the kernel matrix, since it is faster and numerically more stable [7]. So, we will review a simple yet very effective scheme for applying Cholesky decomposition to large and positive definite kernel matrix that is banded or has an envelope structure [40]. This method is in widespread use and, as we shall see, it can yield enormous savings in computer time and storage space.

A kernel matrix K is banded if there is a narrow band around the main diagonal such that all of the entries of K outside of the band are zero. More precisely, if K is $N \times N$, and there is an $s \ll N$ such that $\mathsf{K}_{i,j} = 0$ whenever $|i-j| > s$, then all of the nonzero entries of K are confined to a band of $2s+1$ diagonals centered on the main diagonal. We say that K is banded with band width $2s+1$. Since we are concerned with symmetric matrices in this paper, we only need half of the band. Since $\mathsf{K}_{i,j} = 0$ whenever $i-j > s$, there is a band of s diagonals above the main diagonal that, together with the main diagonal, contains all of the nonzero entries of K. We say that K has semiband width s.

Banded positive definite kernel matrix can be solved economically because it is possible to ignore the entries that lie outside of the band. For this it is crucial that the Cholesky factor R inherits the band structure of the original kernel matrix. Thus we can save storage space by using a data structure that stores only the semiband of K, and R can be stored over K. Just as importantly, computer time is saved because

all operations involving entries outside of the band can be skipped. As we shall soon see, these savings are substantial.

Instead of analyzing banded systems, we will introduce a more general idea, that of the envelope of a matrix. This will increase the generality of the discussion while simplifying the analysis. The envelope of a symmetric or upper-triangular matrix K is a set of ordered pairs (i, j), $i < j$, representing element locations in the upper triangle of K, denned as follows: (i, j) is in the envelope of K if and only if $\mathsf{K}_{k,j} \neq 0$ for some $k \leq i$. Thus if the first nonzero entry of the jth column is $\mathsf{K}_{m,j}$ and $m < j$, then (m, j), $(m + 1, j)$, ..., $(j - 1, j)$ are the members of the envelope of K from the jth column.

The crucial theorem about envelopes (Theorem 2) states that if R is the Cholesky factor of K, then R has the same envelope as K. Thus K can be stored in a data structure that stores only its main diagonal and the entries in its envelope, and R can be stored over K. All operations involving the off-diagonal entries lying outside of the envelope can be skipped. If the envelope is small, substantial savings in computer time and storage space are realized. It should be noted that Banded matrices have small envelopes.

Theorem 2. *Let K be positive definite matrix, and let R be the Cholesky factor of K. Then R and K have the same envelope [40].*

Corollary 2. *Let K be a banded, positive definite matrix with semiband width s. Then its Cholesky factor R also has semiband width s.*

To get an idea of the savings that can be realized by exploiting the envelope structure of a matrix, consider the banded case. If K has semiband width s, then the portion of the jth row that lies in the envelope has at most s entries, so the flop count for the jth step is about s^2. Since there are N steps in the algorithm, the total flop count is about Ns^2.

Letting $Z = R^{-1}$, we have $RZ = I$ where I is the identity matrix. Rewriting this equation in partitioned form as

$$R\,[z_1\ z_2\ ...\ z_N] = [e_1\ e_2\ ...\ e_N]\,,$$

where z_1, ..., z_N and e_1, ..., e_N are the columns of Z and I, respectively, we find that the equation $RZ = I$ is equivalent to the N equations

$$Rz_i = e_i, \quad i = 1, ..., N.$$

Solving these N systems by back substitution, we obtain R^{-1}. Let R be an $N \times N$ upper-triangular matrix with semiband width s. Show that the system $Rz_i = e_i$ can be solved by back substitution in about $2Ns$ flops.

The algorithm for solving a upper -triangular system with semiband width s by back substitution looks like this:

$$\begin{array}{l} for\ j = 1, ..., N \\ \quad for\ k = j - s, ..., j - 1 \\ \quad\quad e_{j,i} \longleftarrow e_{j,i} - R_{j,k}e_{k,i} \\ \quad if\ R_{j,i} = 0,\ set\ error\ flag,\ exit \\ \quad e_{j,i} \longleftarrow e_{j,i}/R_{j,j} \end{array}$$

To get an idea of the execution time of algorithm, let us count the floating-point operations (flops). In the inner loop of algorithm, two flops are executed. These flops are performed s times on the jth time through the outer loop. The outer loop is performed N times, so the total number of flops performed in the k loop is $2Ns$. Each of the N systems has to be solved by back substitution at a cost of $2Ns$ flops. Thus the total flop count is $2N^2s$.

How much does it cost to calculate K^{-1}? The Cholesky decomposition has to be done once, at a cost of Ns^2 flops. the N systems in algorithm have to be solved by back substitution at a cost of $2N^2s$ flops. Thus the total flop count is $Ns(2N + s)$.

For example, the kernel matrix K is 100×100 ($N = 100$) with a semiband width of $s = 10$. If we perform a Cholesky decomposition using a program that does not exploit the band structure of the matrix, the cost of the arithmetic is about $\frac{1}{3}N^3 \approx 3.3 \times 10^5$ flops. In contrast, if we do exploit the band structure,

the cost is about $Ns^2 = 10^4$ flops, which is about 3% of the previous case. In the back substitution steps, substantial but less spectacular savings are achieved. The combined arithmetic cost of back substitution without exploiting the band structure is about $N^3 = 10^6$ flops. If the band structure is exploited, the flop count is about $2N^2 s = 2 \times 10^5$, which is 20% of the previous case. For more details, refer to [40].

6. Numerical results

In this section, we show that the GPR based on the observation from CSRKs can produce sparse kernel matrix, which is in contrast to the kernel matrix derived from GSRKs. The major purpose of this section is to show the capabilities and improved efficiency of the CSRK method for GPR problems with high dimensional and complicated geometries rather than GSRK. The performance of a GPR model can be modified efficiently in producing of a more sparsity and well-conditioned kernel matrix using the prior-CSRK method. We investigate the influence of the prior-CSRK method on the predictability of the GPR models. We will conclude how prior distributions for hyperparameters affect the performance of GPR-based CSRKs such that kernel matrix is more sparse and well-conditioned. To this end, we will investigate the influence of the modified version of MLE on the hyperparameters in sparsity structure of kernel matrix for GPR-based GSRKs problems with high dimensional and complicated geometries. Also, we can obtain statistical criteria such as mean, standard deviation and confidence interval for our prediction. All these results in some tables and figures have been carried out in MATLAB on a laptop with a 2.4 GHz Intel Core i5 processor. We consider M4 kernel as GSRK, and then we put different prior distributions for the hyperparameters in the following example such that our prior distributions cover enough cases. But in most cases Gamma and Normal distributions were appropriate prior distributions. Here, there are several examples with all circumstances, 2D and irregular domains. In addition, W2 kernel used as CSRK for different domains. For test problem, the parameter $\sigma_\zeta^2 = 10^{-2}$ as a variance for noise in the presented method is used to obtain the results and we perform 1000 realizations based on the simulation design. Note that the prior-CSRK can be used to estimate optimal value of the hyperparameter ϵ which implemented by Conjugate Gradient algorithm based on the prior-MLE estimation. In order to investigate the accuracy of the GPR based on the prior-CSRK method, we compute maximum absolute error (MAE) and root mean squared error (RMSE) given by

$$MAE = \max_{1 \le i \le M} |f_i^* - f_i|,$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (f_i^* - f_i)^2},$$

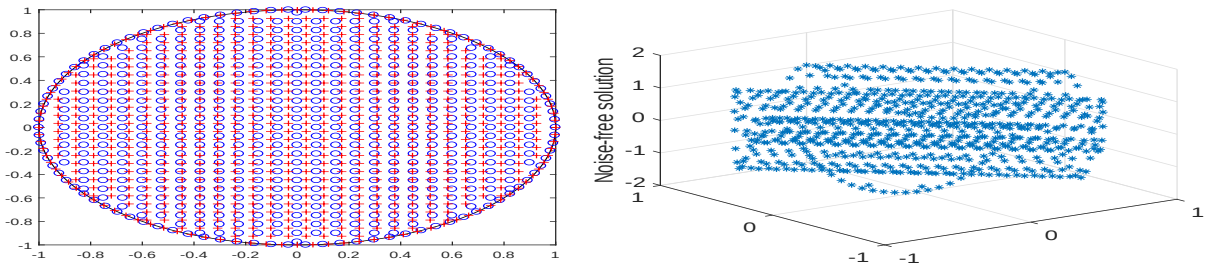
where M is the total number of test points. Also, f_i^* and f_i is used respectively for the predicted mean values and the actual test values. On the other hand, the criterion CN obtained by using the MATLAB command *condest*. We calculate the criteria of sparsity and memory as the number of non-zero entries in the kernel matrix divided by the number of entries in the kernel matrix and the memory requirement for the kernel matrix, respectively.

Example 1

Now for the 2D example on the convex domain, consider 1680 training data $\boldsymbol{x}_i$'s over a circular domain Ω as shown in Figure 2 (a). In this example, we consider a training data set $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, 1680\}$ by the following test function

$$f(\boldsymbol{x}) = \cos\left(2\pi\left(x_1 + x_2\right)\right).$$

Also, the noise distribution is chosen as $\mathcal{N}\left(0, \sigma_\zeta^2\right)$. In addition, to evaluate the accuracy of GPR model 720 test data are used in this example. Figure 2(a) shows training data distribution, marked with circles, and test data distribution, marked with plus signs. The profile of the surface representing the noise-free solution



(a) (b)

Figure 2: The data distribution (a) and the noise-free solution (b) on the circular domain in Example 1

is depicted in Figure 2(b). In Table 2 we present analysis on the CSRK and GSRK methods. More precisely, Table 2 gives the optimal ϵ computed with MLE, the CN of the kernel matrix, MAE, RMSE, sparsity and memory requirement for kernel matrix.

Table 2: Comparison of ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) between CSRK and GSRK methods on the circular domain in Example 1

CSRK						**GSRK**					
ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
0.5915	1.3153×10^6	0.0343	8.7×10^{-3}	0.9460	3.0851×10^6	3.4597	1.3080×10^7	0.0229	7.2×10^{-3}	1	3.0851×10^6

Also, in Figure 3, we show the sparsity structure of the kernel matrix in GPR model based on CSRK and GSRK schemes. These results points out essentially that CSRK method is a little more sparser and a

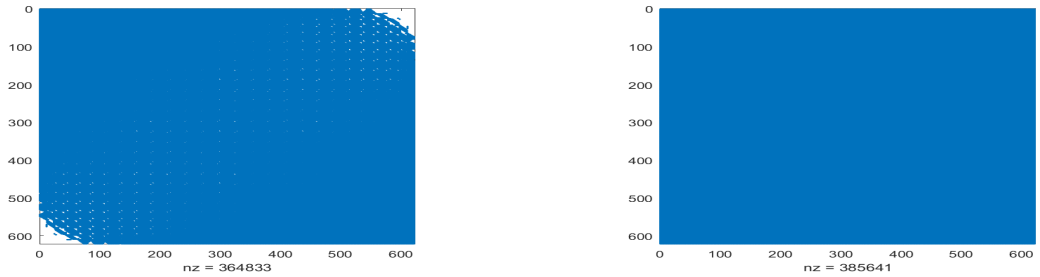


Figure 3: Comparison of sparsity structure between CSRK(left) and GSRK (right) methods on the circular domain in Example 1

little more well-conditioned than GSRK method. As a result, it is computationally very expensive to apply GSRK method to GPR problems. Also, when using kernels with compact support (without priors on the hyperparameters), during GPR training, the main focus will be on providing a high level of accuracy. In this case, the advantage of achieving a sparse covariance matrix for CSRKs will almost disappear, as we can see in Figure 3. To overcome this difficulty, we use prior-MLE method which will be proportional to the product of MLE and a given prior distribution for the hyperparameters as can be seen in (3). Table 3 gives us some useful information on the possible choice of normal priors.

Table 3: Influence of normal priors for ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) on the circular domain in Example 1

Normal prior	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
$\mathcal{N}\left(4.8, 1 \times 10^{-3}\right)$	4.6747	9.4584×10^{1}	0.0839	1.9×10^{-2}	0.0376	2.3667×10^{5}
$\mathcal{N}\left(3.4, 1 \times 10^{-3}\right)$	3.2229	4.6746×10^{2}	0.0552	1.3×10^{-2}	0.0799	4.9811×10^{5}
$\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$	2.2081	3.1850×10^{3}	0.0317	9.6×10^{-3}	0.1502	9.3190×10^{5}
$\mathcal{N}\left(2.2, 1.3 \times 10^{-3}\right)$	1.4664	2.3345×10^{4}	0.0383	9.4×10^{-3}	0.3070	1.8993×10^{6}
$\mathcal{N}\left(1.9, 1.5 \times 10^{-3}\right)$	0.9310	2.0246×10^{5}	0.0287	9.1×10^{-3}	0.6086	3.0851×10^{6}

It is therefore interesting to know how prior distributions for hyperparameters affect the performance of GPR-based CSRKs. In fact, we should investigate the empirical study of prior-MLE strategy on the hyperparameter estimation and the performance of GPR-based CSRKs, for some commonly used kernels in GPR modeling. Indeed, in table 3, we can see how the normal prior of hyperparameter affect the estimates of the hyperparameter. Simulation results show that by selecting a suitable prior on the kernel hyperparameters, and simply estimating the hyperparameters using a modified version of the MLE, during GPR training, in addition to providing the desired accuracy, an appropriate quantity of sparsity will also be obtained. Furthermore, based on the CN of kernel matrix, GPR models using prior-CSRK scheme is more well-conditioned than GPR models using GSRK and CSRK schemes. As a result, it is computationally very suitable to apply the prior-CSRK method to do GPR. In general, for the prior-CSRK scheme we found a good compromise between CN, MAE, RMSE, sparsity and memory requirement.
By empirical analysis, the optimal balanced between accuracy, sparsity and conditioning obtained when prior distribution is $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$. In Table 4 we present analysis on the $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior-CSRK method, which is also compared with the CSRK and GSRK methods studied in Table 2. More precisely, Table 4 gives the optimal ϵ, the CN of the kernel matrix, MAE, RMSE, sparsity and memory for each methods, separately.

Table 4: Comparison of ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) between $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior-CSRK, CSRK and GSRK methods on the circular domain in Example 1

Method	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
Prior-CSRK	2.2081	3.1850×10^{3}	0.0317	9.6×10^{-3}	0.1502	9.3190×10^{5}
CSRK	0.5915	1.3153×10^{6}	0.0343	8.7×10^{-3}	0.9460	3.0851×10^{6}
GSRK	3.4597	1.3080×10^{7}	0.0229	7.2×10^{-3}	1	3.0851×10^{6}

From this study we can note a quite uniform behavior: accuracy of the $(\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right))$ prior-CSRK versus CSRK and GSRK is almost similar. But, on the other hand, we observe a significant reduction of CN, the criteria of sparsity and memory requirement in kernel matrix for the $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior-CSRK scheme compared to the CSRK and GSRK schemes.

To illustrate the applicability of our method for other prior distributions, Table 5 gives us some useful information on the possible choice of *Gamma* priors. A similar empirical analysis has been performed for the *Gamma* prior distribution, and we show that the best prior distribution is $Gamma\left(3030, 1 \times 10^{-3}\right)$. In Table 6 we compare $Gamma\left(3030, 1 \times 10^{-3}\right)$ prior-CSRK method with the CSRK and GSRK methods studied in Table 2. Analyzing these experiments, we can thus observe a behavior - in terms of ϵ, CN, MAE, RMSE, memory and sparsity - similar to that exhibited for $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior-CSRK method and already beforehand remarked.

Table 5: Influence of Gamma priors for ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) on the circular domain in Example 1

Gamma prior	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
$Gamma\left(5280, 1\times10^{-3}\right)$	4.6928	9.3002×10^{1}	0.0925	1.9×10^{-2}	0.0375	2.3654×10^{5}
$Gamma\left(3960, 1\times10^{-3}\right)$	3.2148	5.1371×10^{2}	0.0549	1.2×10^{-2}	0.0800	4.9836×10^{5}
$Gamma\left(3030, 1\times10^{-3}\right)$	2.2122	3.1564×10^{3}	0.0358	9.6×10^{-3}	0.1502	9.3177×10^{5}
$Gamma\left(2310, 1\times10^{-3}\right)$	1.4747	2.2698×10^{4}	0.0336	9.9×10^{-3}	0.3003	1.8577×10^{6}
$Gamma\left(1550, 1\times10^{-3}\right)$	0.9354	1.9803×10^{5}	0.0338	9.0×10^{-3}	0.6004	3.0851×10^{6}

Table 6: Comparison of ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) between $Gamma\left(3030, 1\times10^{-3}\right)$ prior-CSRK, CSRK and GSRK methods on the circular domain in Example 1

Method	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
Prior-CSRK	2.2122	3.1564×10^{3}	0.0358	9.6×10^{-3}	0.1502	9.3177×10^{5}
CSRK	0.5915	1.3153×10^{6}	0.0343	8.7×10^{-3}	0.9460	3.0851×10^{6}
GSRK	3.4597	1.3080×10^{7}	0.0229	7.2×10^{-3}	1	3.0851×10^{6}

Also, in Figure 4, we show the sparsity structure of the kernel matrix in GPR model based on $\mathcal{N}\left(2.8, 1.6\times10^{-3}\right)$ and $Gamma\left(3030, 1\times10^{-3}\right)$ prior-CSRK schemes. It is evident from graphs that kernel matrix is very sparse.

We present predictive mean, predictive standard deviation, confidence interval, noise-free value and ab-

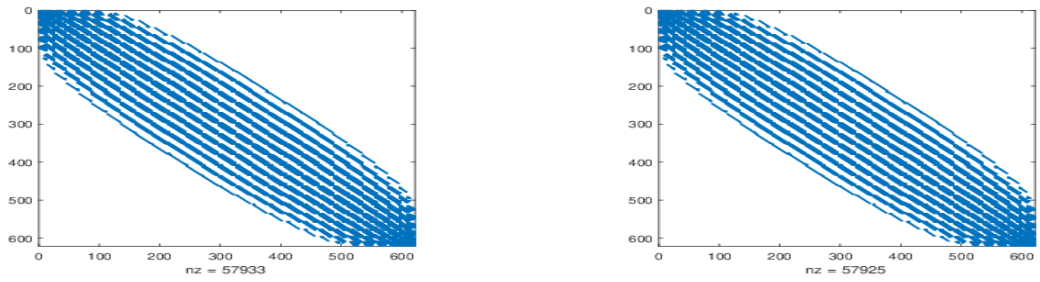


Figure 4: Comparison of sparsity structure between $\mathcal{N}\left(2.8, 1.6\times10^{-3}\right)$ (left) and $Gamma\left(3030, 1\times10^{-3}\right)$ (right) prior-CSRK methods on the circular domain in Example 1

solute error of $\mathcal{N}\left(2.8, 1.6\times10^{-3}\right)$ prior-CSRK for some points in Table 7. Also, we present the results on $Gamma\left(3030, 1\times10^{-3}\right)$ prior-CSRK in Table 8.

The lower bounds and upper bounds are presented for prediction of 720 test data, by using $\mathcal{N}\left(2.8, 1.6\times10^{-3}\right)$ prior in Figures 5 (a) and 5 (b). In addition, the lower bounds and upper bounds are presented for prediction of 720 test data, by using $Gamma\left(3030, 1\times10^{-3}\right)$ prior in Figures 6 (a) and 6 (b).

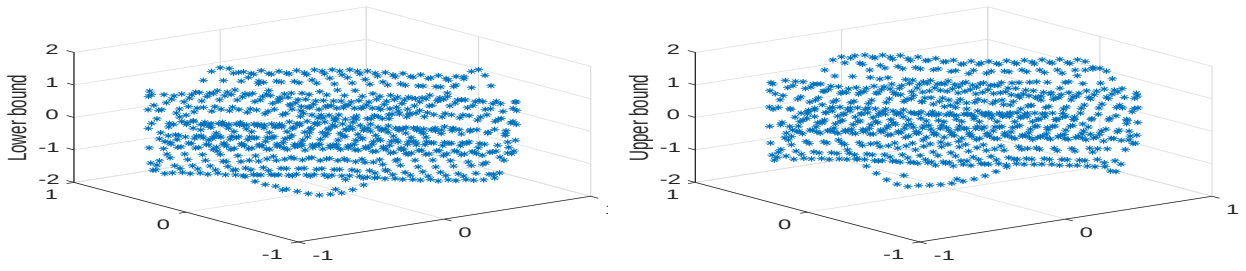
Table 7: The detailed analysis of $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior-CSRK on the circular domain in Example 1

x^*	Mean	Standard deviation	Confidence interval	Noise-free value	Absolute error
$(-0.9257, -0.2400)$	0.5035	0.1076	$(0.2927, 0.7143)$	0.5052	1.7×10^{-3}
$(-0.7200, -0.5143)$	0.0987	0.1071	$(-0.1111, 0.3086)$	0.0986	2×10^{-4}
$(-0.6514, 0.4457)$	0.2758	0.0959	$(0.0879, 0.4637)$	0.2747	1.1×10^{-3}
$(-0.4457, -0.3771)$	0.4411	0.0256	$(0.3910, 0.4911)$	0.4420	9×10^{-4}
$(0, -1)$	1.0012	0.0836	$(0.8373, 1.1651)$	1	1.2×10^{-3}
$(0.3086, 0.4457)$	0.0272	0.0463	$(-0.0636, 0.1180)$	0.0269	3×10^{-4}
$(0.4457 - 0.7886)$	-0.5503	0.1129	$(-0.7717, -0.3290)$	-0.5509	6×10^{-4}
$(0.58290.0343)$	-0.7392	0.1231	$(-0.9804, -0.4980)$	-0.7411	2×10^{-3}
$(0.7886, 0.1029)$	0.7743	0.1309	$(0.5177, 1.0308)$	0.7762	1.9×10^{-3}
$(0.8558, -0.5173)$	-0.5270	0.0390	$(-0.6033, -0.4506)$	-0.5275	6×10^{-4}

Table 8: The detailed analysis of $Gamma\left(3030, 1 \times 10^{-3}\right)$ prior-CSRK on the circular domain in Example 1

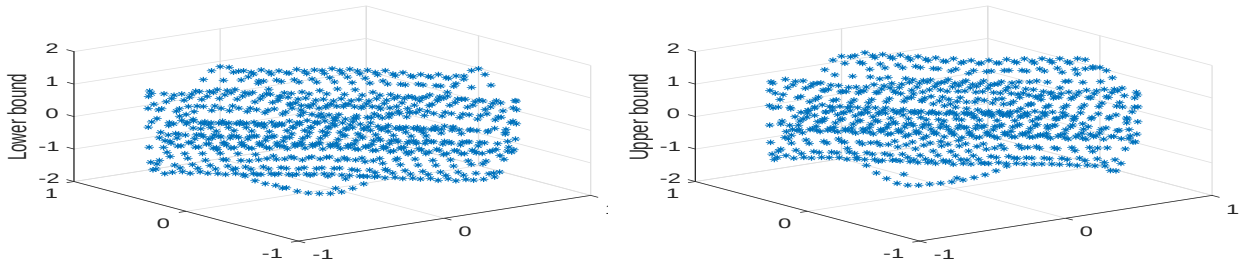
x^*	Mean	Standard deviation	Confidence interval	Noise-free value	Absolute error
$(-0.9257, -0.2400)$	0.5042	0.1007	$(0.3069, 0.7015)$	0.5052	9×10^{-4}
$(-0.7200, -0.5143)$	0.0980	0.1119	$(-0.1212, 0.3173)$	0.0986	5×10^{-4}
$(-0.6514, 0.4457)$	0.2732	0.0931	$(0.0907, 0.4558)$	0.2747	1.4×10^{-3}
$(-0.4457, -0.3771)$	0.4405	0.0238	$(0.3939, 0.4871)$	0.4420	1.4×10^{-3}
$(0, -1)$	0.9981	0.0801	$(0.8412, 1.1550)$	1	1.9×10^{-3}
$(0.3086, 0.4457)$	0.0283	0.0477	$(-0.0651, 0.1218)$	0.0269	1.4×10^{-3}
$(0.4457, -0.7886)$	-0.5490	0.1043	$(-0.7534, -0.3446)$	-0.5509	1.9×10^{-3}
$(0.5829, 0.0343)$	-0.7429	0.1163	$(-0.9708, -0.5151)$	-0.7411	1.8×10^{-3}
$(0.7886, 0.1029)$	0.7751	0.1279	$(0.5244, 1.0259)$	0.7762	1.1×10^{-3}
$(0.8558, -0.5173)$	-0.5281	0.0364	$(-0.5994, -0.4569)$	-0.5275	6×10^{-4}

The predictive means and standard deviation are shown for 720 test data by using $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior in Figure 7. The predictive means and standard deviation are shown for 720 test data by using $Gamma\left(3030, 1 \times 10^{-3}\right)$ prior in Figure 8. The absolute errors are demonstrated in Figure 9 for prediction of 720 test data, by using these priors.
Based on these results, a low standard deviation indicates that the predicted values for each test point in simulations tend to be close to the predicted mean. Also, the predicted mean of each test point will fall within interval of lower and upper bounds.



(a) (b)

Figure 5: The lower bounds (a) and upper bounds (b) by using $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior on the circular domain for prediction of test data in Example 1



(a) (b)

Figure 6: The lower bounds (a) and upper bounds (b) by using $Gamma\left(3030, 1 \times 10^{-3}\right)$ prior on the circular domain for prediction of test data in Example 1

Example 2.

Now, to show the efficiency of proposed method on the more complexed and irregularly shaped domain with holes (non-convex domain), we consider 1820 data $\boldsymbol{x}_i$'s over a non-convex domain Ω as shown in Figure 10 (a). Specifically, we hereby define the boundary of this non-convex domain Ω as follows

$$\partial\Omega = \{(\rho, \theta) \,|\rho\,(\theta) = 1 + 0.1\,(\sin\,(6\theta) + \sin\,(3\theta))\}.$$

Let $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, 1820\}$ be a training data set. Now, we evaluate the noisy scattered test data set based on the $\zeta_i \overset{i.i.d}{\sim} \mathcal{N}\left(0, \sigma_{\zeta}^2\right)$, $i = 1, \ldots, 1820$. It should be mentioned that 780 test data are used to evaluate the accuracy of GPR model in in this example. Figure 10 (a) shows training data distribution, marked with circles, and test data distribution, marked with plus signs. The profile of the surface representing the noise-free solution is depicted in Figure 10 (b). In Table 9, we present the analysis of the GPR based on the CSRK

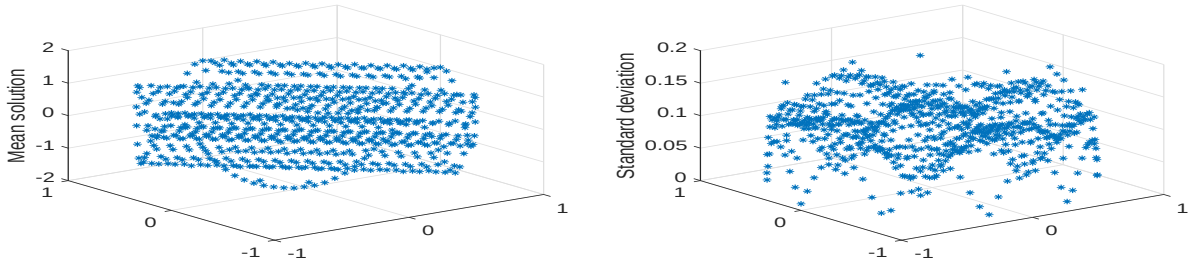


Figure 7: The predictive means (left) and standard deviation (right) for test data on the circular domain in Example 1 by using $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior

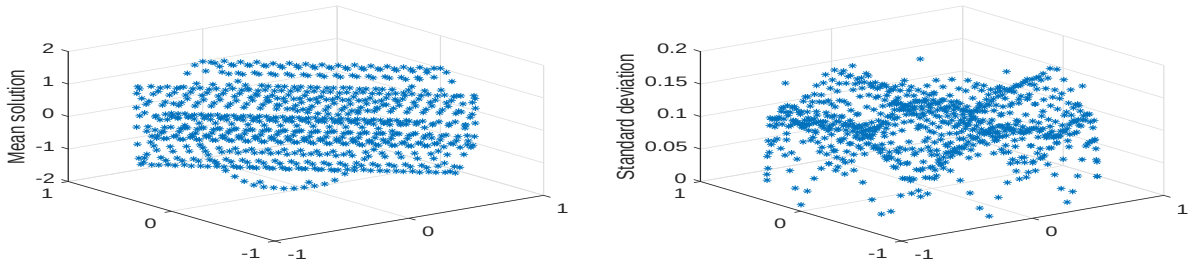


Figure 8: The predictive means (left) and standard deviation (right) for test data on the circular domain in Example 1 by using $Gamma\left(3030, 1 \times 10^{-3}\right)$ prior

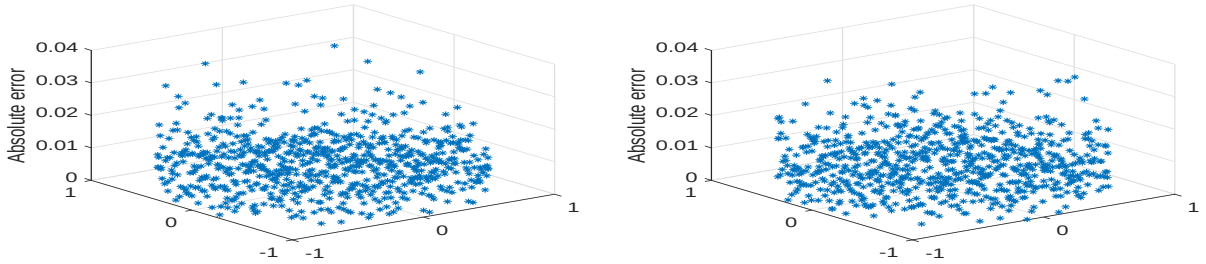
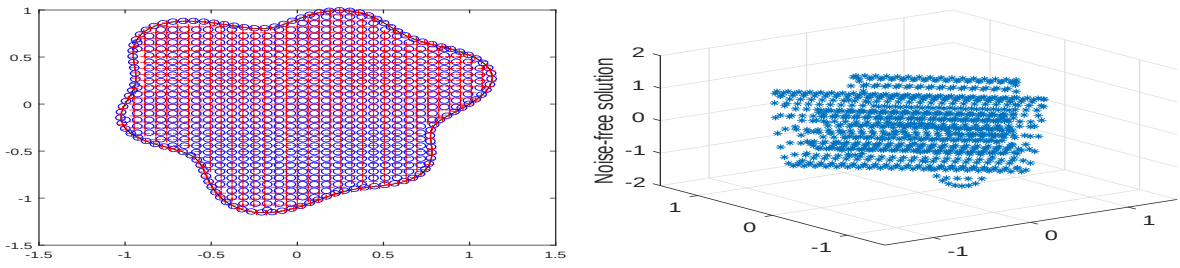


Figure 9: The absolute errors of $\mathcal{N}\left(2.8, 1.6 \times 10^{-3}\right)$ prior (left) and $Gamma\left(3030, 1 \times 10^{-3}\right)$ prior (right) for prediction of test data on the circular domain in Example 1



(a) (b)

Figure 10: The data distribution (a) and the noise-free solution (b) on the non-convex domain in Example 2

and GSRK methods. More precisely, Table 9 gives the optimal hyperparameter ϵ which is computed with MLE. Also, the CN of kernel matrix for both the CSRK and GSRK methods is calculated. In addition, we report the MAE, RMSE, sparsity and memory requirement for kernel matrix computed with optimal ϵ.

Table 9: Comparison of ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) between CSRK and GSRK methods on the non-convex domain in Example 2

CSRK						**GSRK**					
ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
0.5758	2.2045×10^6	0.0288	8.3×10^{-3}	0.9455	4.5120×10^6	3.4323	1.8617×10^7	0.0269	6.9×10^{-3}	1	4.5120×10^6

Also, in Figure 11, we show the sparsity structure of the kernel matrix in GPR model based on CSRK versus GSRK schemes. These results points out essentially that CSRK method is a little more sparse and a

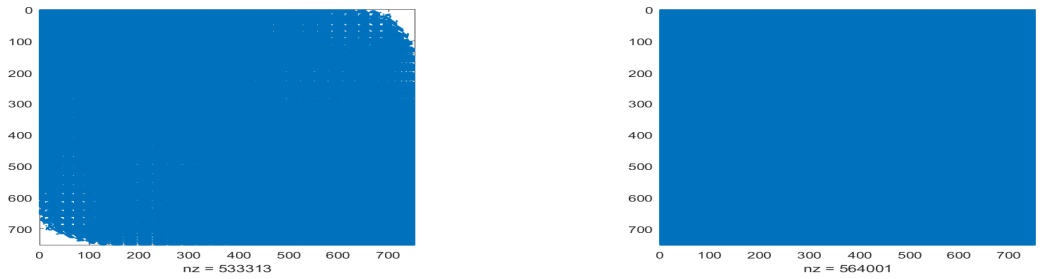


Figure 11: Comparison of sparsity structure between CSRK (left) and GSRK (right) methods on the non-convex domain in Example 2

little more well-conditioned than GSRK method. According to the results, the use of kernels with compact support did not significantly increase the sparseness of the kernel matrix. Consequently, we will not benefit from the computational advantages of sparse matrices. As mentioned before, to meet this challenge, we use the prior-MLE method to achieve the best possible balance between sparsity, accuracy and conditioning.
Table 10 gives us the prior-MLE estimation for hyperparameter based on the possible choice of normal priors.
Simulation results show that by selecting a suitable prior on the kernel hyperparameters, and simply estimating the hyperparameters using a modified version of the MLE, during GPR training, in addition to providing the desired accuracy, an appropriate quantity of sparsity will also be obtained. Furthermore, based on the CN of kernel matrix, GPR models using prior-CSRK scheme is more well- conditioned than GPR models using GSRK and CSRK schemes. As a result, it is computationally very suitable to apply the prior-CSRK method to do GPR even on more complexed domain. In general, for the suitable prior-CSRK scheme we found a good compromise between CN, MAE, RMSE, sparsity and memory requirement.

By empirical analysis, the optimal balanced between accuracy, sparsity and conditioning obtained when prior distribution is $\mathcal{N}\left(2.7, 1 \times 10^{-3}\right)$. In Table 11 we present analysis on the $\mathcal{N}\left(2.7, 1 \times 10^{-3}\right)$ prior-CSRK method, which is also compared with the CSRK and GSRK methods studied in Table 9 . More precisely, Table 11 gives the optimal ϵ, the CN of the kernel matrix, MAE, RMSE, sparsity and memory for each methods, separately.

From Table 11, we observe that the accuracy of the $(\mathcal{N}\left(2.7, 1 \times 10^{-3}\right))$ prior-CSRK method is comparable to the CSRK and GSRK methods studied in Table 9. Additionally, the $(\mathcal{N}\left(2.7, 1 \times 10^{-3}\right))$ prior-CSRK scheme

Table 10: Influence of normal priors for ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) on the non-convex domain in Example 2

Normal prior	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
$\mathcal{N}\left(4.7, 1\times10^{-3}\right)$	4.5355	1.9802×10^{2}	0.0736	1.6×10^{-2}	0.0438	4.0110×10^{5}
$\mathcal{N}\left(3.5, 1\times10^{-3}\right)$	3.2447	9.3335×10^{2}	0.0598	1.2×10^{-2}	0.0790	7.1892×10^{5}
$\mathcal{N}\left(2.7, 1\times10^{-3}\right)$	2.2562	5.3997×10^{3}	0.0316	9.4×10^{-3}	0.14727	1.3350×10^{6}
$\mathcal{N}\left(2.5, 1.5\times10^{-3}\right)$	1.4562	4.4881×10^{4}	0.0411	9.1×10^{-3}	0.3025	2.7356×10^{6}
$\mathcal{N}\left(1.8, 1.1\times10^{-3}\right)$	0.9110	3.8470×10^{5}	0.0366	8.8×10^{-3}	0.6034	4.5120×10^{6}

Table 11: Comparison of ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) between $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior-CSRK, CSRK and GSRK methods on the non-convex domain in Example 2

Method	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
Prior-CSRK	2.2562	5.3997×10^{3}	0.0316	9.4×10^{-3}	0.14727	1.3350×10^{6}
CSRK	0.5758	2.2045×10^{6}	0.0288	8.3×10^{-3}	0.9455	4.5120×10^{6}
GSRK	3.4323	1.8617×10^{7}	0.0269	6.9×10^{-3}	1	4.5120×10^{6}

offers a remarkable decrease of of CN, the criterions of sparsity and memory in kernel matrix compared to the CSRK and GSRK schemes.

To illustrate the applicability of our method for other prior distributions on the more complexed domain with holes, Table 12 gives us the prior-MLE estimation for hyperparameter based on the possible choice of Gamma priors. A similar empirical analysis has been performed for the Gamma prior distribution, and we show that the best prior distribution is $Gamma\left(3260, 1\times10^{-3}\right)$. In Table 13 we compare $Gamma\left(3260, 1\times10^{-3}\right)$ prior-CSRK method with the CSRK and GSRK methods studied in Table 9.

Table 12: Influence of Gamma priors for ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) on the non-convex domain in Example 2

Gamma prior	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
$Gamma\left(5350, 1\times10^{-3}\right)$	4.5527	1.9464×10^{2}	0.0772	1.6×10^{-2}	0.0437	4.0081×10^{5}
$Gamma\left(4140, 1\times10^{-3}\right)$	3.2191	9.6888×10^{2}	0.0513	1.2×10^{-2}	0.0792	7.2062×10^{5}
$Gamma\left(3260, 1\times10^{-3}\right)$	2.2572	5.2306×10^{3}	0.0323	9.6×10^{-3}	0.1472	1.3348×10^{6}
$Gamma\left(2470, 1\times10^{-3}\right)$	1.4567	4.4792×10^{4}	0.0327	9.0×10^{-3}	0.3024	2.7352×10^{6}
$Gamma\left(1660, 1\times10^{-3}\right)$	0.9153	3.8425×10^{5}	0.0322	8.8×10^{-3}	0.6010	4.1520×10^{6}

Analyzing these experiments, we can thus observe a behavior – in terms of ϵ, CN, MAE, RMSE, memory and sparsity – similar to that exhibited for $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior-CSRK method and already beforehand remarked. In Figure 12 we show the sparsity structure of the kernel matrix in GPR model based on $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior-CSRK in comparison with $Gamma\left(3260, 1\times10^{-3}\right)$ prior-CSRK. It is evident from graphs that the kernel matrix is very sparse.

We present an extensive and detailed analysis on $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior-CSRK method is used to obtain predictive mean, predictive standard deviation, confidence interval, noise-free value and absolute error for

Table 13: Comparison of ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) between $Gamma\left(3260, 1\times10^{-3}\right)$ prior-CSRK, CSRK and GSRK methods on the non-convex domain in Example 2

Method	ϵ	**CN**	**MAE**	**RMSE**	**Sparsity**	**Memory**
Prior-CSRK	2.2572	5.2306×10^{3}	0.0323	9.6×10^{-3}	0.1472	1.3348×10^{6}
CSRK	0.5758	2.2045×10^{6}	0.0288	8.3×10^{-3}	0.9455	4.5120×10^{6}
GSRK	3.4323	1.8617×10^{7}	0.0269	6.9×10^{-3}	1	4.5120×10^{6}

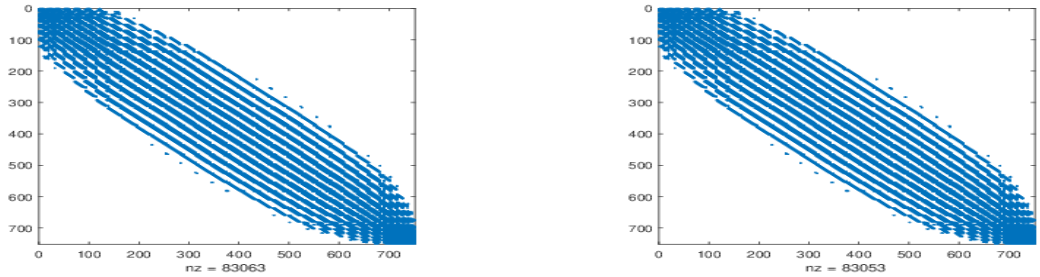


Figure 12: Comparison of sparsity structure between $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ (left) and $Gamma\left(3260, 1\times10^{-3}\right)$ (right) prior-CSRK methods on the non-convex domain in Example 2

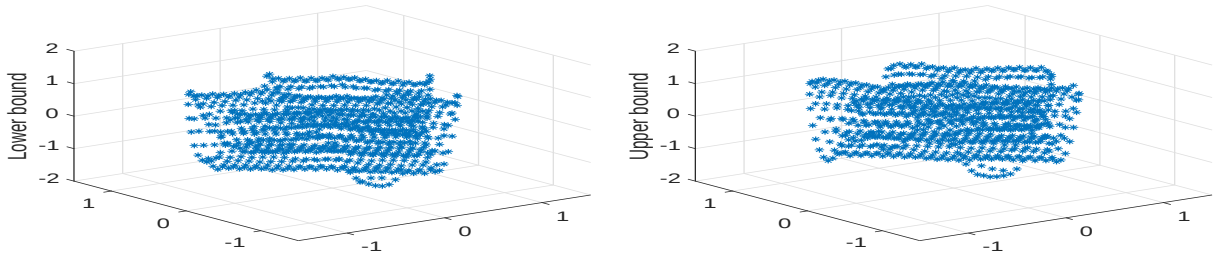
some points in Table 14. Also, we present an extensive and detailed analysis on $Gamma\left(3260, 1\times10^{-3}\right)$ prior-CSRK in Table 15.
Figures 13 (a) and 13 (b) show lower bounds and upper bounds for prediction of 780 test data, by using $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior. In addition, Figures 14 (a) and 14 (b) show lower bounds and upper bounds for prediction of 780 test data, by using $Gamma\left(3260, 1\times10^{-3}\right)$ prior. Figure 15 show the predictive means and standard deviation for 780 test data by using $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior. Also, Figure 16 show the predictive means and standard deviation for 780 test data by using $Gamma\left(3260, 1\times10^{-3}\right)$ prior. Figure 17 show absolute errors for prediction of 780 test data, by using these priors.

Table 14: The detailed analysis of $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior-CSRK on the non-convex domain in Example 2

x^*	**Mean**	**Standard deviation**	**Confidence interval**	**Noise-free value**	**Absolute error**
$(-0.6384, -0.5274)$	0.5057	0.0747	$(0.3592, 0.6522)$	0.5048	9×10^{-4}
$(-0.5253, -0.7763)$	-0.3170	0.0310	$(-0.3779, -0.2562)$	-0.3186	1.5×10^{-3}
$(-0.4737, -0.9357)$	-0.8423	0.0272	$(-0.8956, -0.7890)$	-0.8424	1×10^{-4}
$(-0.3789, 0.6947)$	-0.4026	0.0821	$(-0.5634, -0.2417)$	-0.4017	9×10^{-4}
$(-0.2526, 0.3158)$	0.9221	0.0623	$(0.8001, 1.0442)$	0.9223	2×10^{-4}
$(-0.1895, -0.4421)$	-0.6767	0.0712	$(-0.8162, -0.5372)$	-0.6773	6×10^{-4}
$(0.0632, -0.3158)$	-0.0154	0.0931	$(-0.1979, 0.1670)$	-0.0165	1.1×10^{-3}
$(0.1895, -0.7579)$	-0.9071	0.1077	$(-1.1183, -0.6960)$	-0.9090	1.9×10^{-3}
$(0.2526, 0.3789)$	-0.6757	0.0549	$(-0.7832, -0.5681)$	-0.6773	1.6×10^{-3}
$(0.5053, 0.2526)$	0.0510	0.0651	$(-0.0766, 0.1785)$	0.0496	1.4×10^{-3}
$(0.6947, 0.3789)$	0.8967	0.0872	$(0.7258, 1.0675)$	0.8947	1.9×10^{-3}
$(0.7842, -0.2529)$	-0.9815	0.0701	$(-1.1190, -0.8440)$	-0.9808	7×10^{-4}

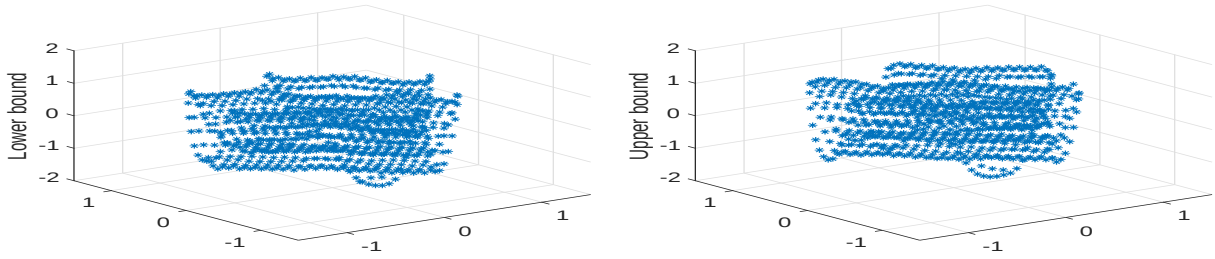
Table 15: The detailed analysis of $Gamma\left(3260, 1\times10^{-3}\right)$ prior-CSRK on the non-convex domain in Example 2

x^*	Mean	Standard deviation	Confidence interval	Noise-free value	Absolute error
$(-0.6384, -0.5274)$	0.5048	0.0775	$(0.3530, 0.6566)$	0.5048	0
$(-0.5253, -0.7763)$	-0.3198	0.0327	$(-0.3838, -0.2557)$	-0.3186	1.2×10^{-3}
$(-0.4737, -0.9357)$	-0.8439	0.0248	$(-0.8925, -0.7952)$	-0.8424	1.4×10^{-3}
$(-0.3789, 0.6947)$	-0.4035	0.0858	$(-0.5716, -0.2354)$	-0.4017	1.8×10^{-3}
$(-0.2526, 0.3158)$	0.9231	0.0559	$(0.8136, 1.0326)$	0.9223	8×10^{-4}
$(-0.1895, -0.4421)$	-0.6792	0.0714	$(-0.8191, -0.5393)$	-0.6773	1.9×10^{-3}
$(0.0632, -0.3158)$	-0.0177	0.0932	$(-0.2003, 0.1649)$	-0.0165	1.1×10^{-3}
$(0.1895, -0.7579)$	-0.9109	0.1007	$(-1.1083, -0.7136)$	-0.9090	1.9×10^{-3}
$(0.2526, 0.3789)$	-0.6759	0.0547	$(-0.7830, -0.5688)$	-0.6773	1.4×10^{-3}
$(0.5053, 0.2526)$	0.0481	0.0643	$(-0.0778, 0.1741)$	0.0496	1.5×10^{-3}
$(0.6947, 0.3789)$	0.8936	0.0799	$(0.7370, 1.0501)$	0.8947	1.2×10^{-3}
$(0.7842, -0.2529)$	-0.9801	0.0697	$(-1.1166, -0.8435)$	-0.9808	7×10^{-4}



(a) (b)

Figure 13: The lower bounds (a) and upper bounds (b) by using $\mathcal{N}\left(2.7, 1\times10^{-3}\right)$ prior on the non-convex domain for prediction of test data in Example 2



(a) (b)

Figure 14: The lower bounds (a) and upper bounds (b) by using $Gamma\left(3260, 1\times10^{-3}\right)$ prior on the non-convex domain for prediction of test data in Example 2

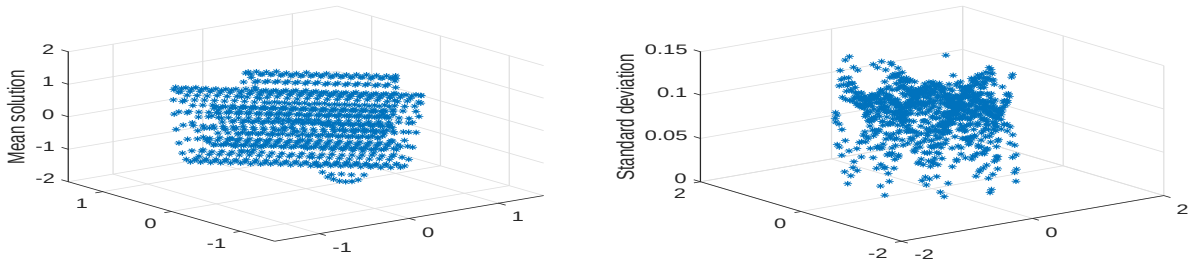


Figure 15: The predictive means (left) and standard deviation (right) for test data on the non-convex domain in Example 2 by using $\mathcal{N}\left(2.7, 1 \times 10^{-3}\right)$ prior

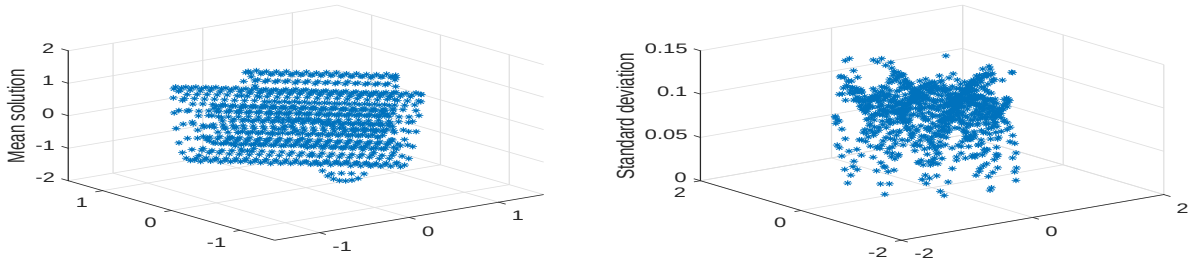


Figure 16: The predictive means (left) and standard deviation (right) for test data on the non-convex domain in Example 2 by using $Gamma\left(3260, 1 \times 10^{-3}\right)$ prior

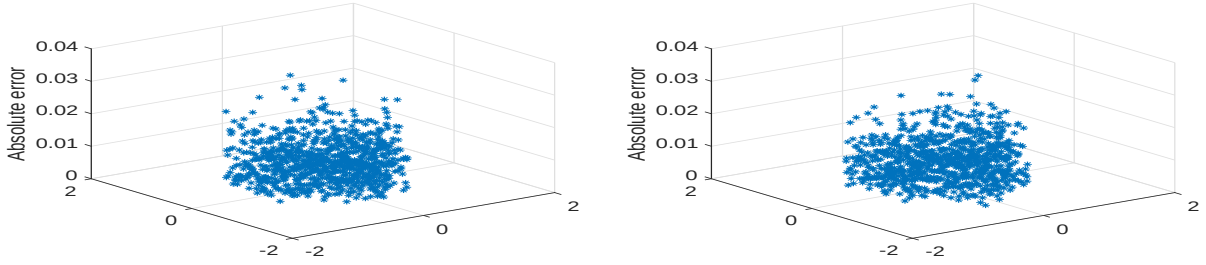


Figure 17: The absolute errors of $\mathcal{N}\left(2.7, 1 \times 10^{-3}\right)$ prior (left) and $Gamma\left(3260, 1 \times 10^{-3}\right)$ prior (right) for prediction of test data on the non-convex domain in Example 2

Based on numerical results in Examples 1 and 2, by selecting a suitable prior on the kernel hyperparameters, and simply estimating the hyperparameters using a modified version of the MLE, during GPR training, in addition to providing the desired accuracy, an appropriate quantity of sparsity will also be obtained. This modified version of the MLE will be proportional to the product of MLE and a given prior distribution for the hyperparameters. Here, the MLE ensures that the obtained model has a maximum agreement with the training data, and the suitable prior distribution guarantees the achievement of a sparse covariance matrix.

7. Application

To emphasize the importance of this work, we provide real dataset to illustrate how our method could be applied in practice. Therefore, we will fit a GP and compare our presented method (Prior-CSRK) with GPR-based GSRK and CSRK methods.

CO_2 data

We will use a modelling problem concerning the concentration of CO_2 in the atmosphere to illustrate how prior distribution affect the estimates of the hyperparameters and the accuracy of GPR models such that kernel matrix is more sparse and well-conditioned. The data consists of weekly average atmospheric CO_2 concentrations (in parts per million by volume (ppmv)) derived from in situ air samples collected at the Mauna Loa Observatory, Hawaii, from 2010 until 2020 ($https://gml.noaa.gov/ccgg/trends/data.html$). Our goal is the model the CO_2 concentration as a function of time (calendar year). The data set is split into the training data set (2/3 samples) and the testing data set (1/3 samples). In Table 16 we present analysis on the CSRK and GSRK methods. More precisely, Table 16 gives the optimal computed ϵ with MLE, the CN of the kernel matrix, sparsity and memory requirement for kernel matrix.

Table 16: Comparison of ϵ, CN, sparsity and memory (in bytes) between CSRK and GSRK methods for CO_2 data

CSRK				**GSRK**			
ϵ	**CN**	**Sparsity**	**Memory**	ϵ	**CN**	**Sparsity**	**Memory**
0.1195	9.5397×10^4	0.9316	1.2418×10^6	1.4777	1.9488×10^5	1	1.2418×10^6

Also, in Figure 18, we show the sparsity structure of the kernel matrix in GPR model based on CSRK and GSRK schemes. These results point out essentially that CSRK method is a little more sparser and a

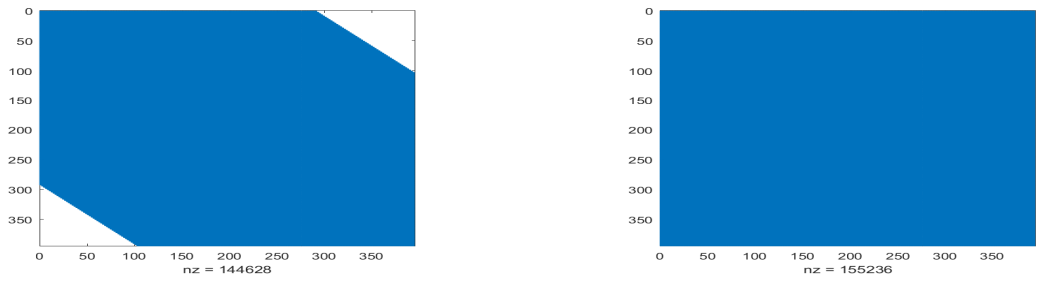


Figure 18: Comparison of sparsity structure between CSRK(left) and GSRK (right) methods for CO_2 data

little more well-conditioned than GSRK method. As a result, it is computationally very expensive to apply

GSRK method to GPR problems. Also, when using kernels with compact support (without priors on the hyperparameters), during GPR training, the main focus will be on providing a high level of accuracy. In this case, the advantage of achieving a sparse covariance matrix for CSRKs will almost disappear, as we can see in Figure 18. To overcome this difficulty, we use prior-MLE method which will be proportional to the product of MLE and a given prior distribution for the hyperparameters as can be seen in (3). It is therefore interesting to know how prior distributions affect the performance of GPR. A similar empirical analysis has been performed for prior distribution of previous example, the optimal balanced between accuracy, sparsity and conditioning obtained when prior distributions are $\mathcal{N}\left(2.6, 1.1 \times 10^{-2}\right)$ and $Gamma\left(1.85 \times 10^{2}, 3 \times 10^{-2}\right)$. Also, the accuracy of the prior-CSRK method compared with the CSRK and GSRK methods is almost same, we will thus report the numerical results regardless of them. In Table 17 we present analysis on the $\mathcal{N}\left(2.6, 1.1 \times 10^{-2}\right)$ prior-CSRK method, which is also compared with the CSRK and GSRK methods studied in Table 16. More precisely, Table 17 gives the optimal ϵ, the CN of the kernel matrix, sparsity and memory for each methods, separately.

Table 17: Comparison of ϵ, CN, MAE, RMSE, sparsity and memory (in bytes) between $\mathcal{N}\left(2.6, 1.1 \times 10^{-2}\right)$ prior-CSRK, CSRK and GSRK methods for CO_2 data

Method	$\boldsymbol{\epsilon}$	**CN**	**Sparsity**	**Memory**
Prior-CSRK	1.0976	7.1093×10^{3}	0.1535	3.8444×10^{5}
CSRK	0.1195	19.5397×10^{4}	0.9316	1.2418×10^{6}
GSRK	1.4777	1.9488×10^{5}	1	1.2418×10^{6}

In Table 18 we compare $Gamma\left(1.85 \times 10^{2}, 3 \times 10^{-2}\right)$ prior-CSRK method with the CSRK and GSRK methods studied in Table 16. From these studies we can note a quite uniform behavior: accuracy of the prior-CSRK versus CSRK and GSRK is almost similar. But, on the other hand, we observe a significant reduction of CN, the criteria of sparsity and memory requirement in kernel matrix for the prior-CSRK scheme compared to the CSRK and GSRK schemes.

Also, in Figure 19, we show the sparsity structure of the kernel matrix in GPR model based on $\mathcal{N}\left(2.6, 1.1 \times 10^{-2}\right)$ and $Gamma\left(1.85 \times 10^{2}, 3 \times 10^{-2}\right)$ prior-CSRK schemes. It is evident from graphs that kernel matrix is very sparse.

Table 18: Comparison of ϵ, CN, sparsity and memory (in bytes) between $Gamma\left(1.85 \times 10^{2}, 3 \times 10^{-2}\right)$ prior-CSRK, CSRK and GSRK methods for CO_2 data

Method	$\boldsymbol{\epsilon}$	**CN**	**Sparsity**	**Memory**
Prior-CSRK	1.0888	7.2024×10^{3}	0.1539	3.8549×10^{5}
CSRK	0.1195	19.5397×10^{4}	0.9316	1.2418×10^{6}
GSRK	1.4777	1.9488×10^{5}	1	1.2418×10^{6}

In conclusion, we have seen an example of how prior distribution may affect the performance of GPR models by using CSRK schemes, and that the ability to estimate hyperparameters by the real data is useful in practice. Accordingly, the final sparse kernel matrix by CSRKs can be efficiently derived from the modified version of MLE, and available data.

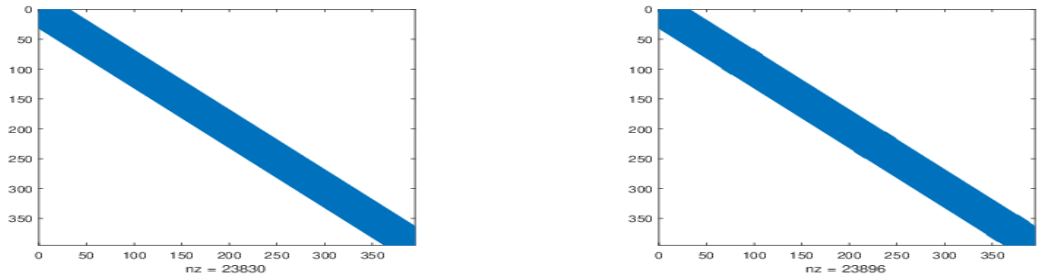


Figure 19: Comparison of sparsity structure between $\mathcal{N}\left(2.6, 1.1 \times 10^{-2}\right)$ (left) and $Gamma\left(1.85 \times 10^{2}, 3 \times 10^{-2}\right)$ (right) prior-CSRK methods for CO_2 data

8. Conclusion

In this paper, we illustrated that the GPR learned from the data and selected CSRKs can produce sparse kernel matrix which is considerably more efficient than the GP model constructed using GSRKs. The proposed method in this paper deals directly with the kernel matrix itself which is more convenient to implement in practice and turns out to be well-conditioned. If we use the CSRKs in GPR then the main difference to the global support structure is that now the kernel matrix can be made sparse by scaling the support of the CSRK appropriately. In practice, the interest in CSRKs waned slightly as it became evident that, to obtain good accuracy, the size of the support is increased such that the overlap distance should cover most nodes in the point set. Hence, when using CSRKs, during GPR training, the main focus will be on providing a high level of accuracy. In this case, the advantage of achieving a sparse covariance matrix for CSRKs will almost disappear, as we will see in the numerical results. Accordingly, by selecting a suitable prior on the kernel hyperparameters, and simply estimating the hyperparameters using a modified version of the MLE, during GPR training, in addition to providing the desired accuracy, an appropriate quantity of sparsity will also be obtained. Therefore, we use prior-MLE method which will be proportional to the product of MLE and a given prior distribution for the hyperparameters as can be seen in (3). Here, the MLE ensures that the obtained model has a maximum agreement with the training data, and the suitable prior distribution guarantees the achievement of a sparse covariance matrix. We show that the modified version of MLE would have a great impact on the sparse representation of the GP models and its performance. Then, we provided the first empirical study of the impact of the prior distributions on the hyperparameter estimation and the performance of GPR-based CSRKs, for some commonly used kernels in GPR modeling through several examples on the irregular domains. We finally investigate the effect of prior on the predictability of GPR models based on the real dataset. The derived results suggest the proposed method leads to more sparsity and well-conditioned kernel matrices in all cases.

References

[1] Bouhlel MA, Martins JRRA. Gradient-enhanced kriging for high-dimensional problems. Engineering with Computers. 2019;35:157–173.
[2] Zhang X, Pandey MD. HALK: A hybrid active-learning Kriging approach and its applications for structural reliability analysis. Engineering with Computers. 2021; https://doi.org/10.1007/s00366-021-01308-8.
[3] Gao W, Karbasi M, Hasanipanah M, Zhang X, Guo J. Developing GPR model for forecasting the rock fragmentation in surface mines. Engineering with Computers. 2018;34:339–345.
[4] Arthur CK, Temeng VA, Ziggah YY. Novel approach to predicting blast-induced ground vibration using Gaussian process regression. Engineering with Computers. 2020;36:29–42.
[5] Isaaks E, Srivastava R. Applied geostatistics. Oxford University, London; 2011.
[6] Rasmussen CE. Evaluation of Gaussian Processes and Other Methods for Non-Linear Regression. University of Toronto; 1999.
[7] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. MIT Press, Boston; 2006.
[8] Chatrabgoun O, Esmaeilbeigi M, Cheraghi M, Daneshkhah A. Stable Likelihood Computation for Machine Learning of Linear Differential Operators with Gaussian Processes. International Journal for Uncertainty Quantification. 2022;12:75–99.

[9] Shamshirband S, Goci'c M, Petkovi'c D, Saboohi H, Herawan T, Kiah MLM, Akib S. Soft-computing methodologies for precipitation estimation: a case study. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2015;8:1353–1358.
[10] Shamshirband S, Mohammadi K, Yee L, Petkovi'c D, Mostafaeipour A. A comparative evaluation for identifying the suitability of extreme learning machine to predict horizontal global solar radiation. Renewable and Sustainable Energy Reviews. 2015;52:1031–1042.
[11] MacKay DJ. Gaussian processes a replacement for supervised neural networks?. Tutorial lecture notes for NIPS 1997; 1997.
[12] Wilson AG, Adams RP. Gaussian process kernels for pattern discovery and extrapolation. in: Proceedings of the 30th International Conference on Machine Learning (ICML-13). 2013;1067–1075.
[13] Fasshauer GE, McCourt MJ. Kernel-based Approximation Methods using Matlab. World Scientific, Singapore; 2015.
[14] Csato L, Opper M. Sparse online Gaussian processes. Neural Computation. 2002;14:641–668.
[15] Quinonero-Candela J, Rasmussen CE. A unifying view of sparse approximate Gaussian process regression. Journal Of Machine Learning Research. 2005;6:1939–1959.
[16] Snelson E, Ghahramani Z. Local and global sparse Gaussian process approximations. Artificial Intelligence and Statistics. 2007;11.
[17] Williams CKI, Seeger M. Using the Nystrom method to speed up kernel machines. In Leen TK, Dietterich TG, Tresp V, editors, Advances in Neural Information Processing Systems, The MIT Press. 2001;13.
[18] Quiñonero-Candela J, Rasmussen CE, Williams CKI. "Approximation methods for Gaussian process regression," in Large-Scale Kernel Machines. MIT Press, Cambridge. 2007;203–224.
[19] Schreiter J, Nguyen-Tuong D, Toussaint M. Efficient sparsification for Gaussian process regression. Neurocomputing. 2016;192:29–37.
[20] Tresp V. A Bayesian committee machine. Neural Computation. 2000;12:2719–2741.
[21] Ranganathan A, Yang MH, Ho J. Online Sparse Gaussian Process Regression and Its Applications. IEEE Transactions on Image Processing. 2011;20:391–404.
[22] Brahim-Belhouari S, Bermak A. Gaussian process for nonstationary time series prediction. Computational Statistics & Data Analysis. 2004;47:705–712.
[23] MacKay DJ. Introduction to Gaussian processes. NATO ASI Series F Computer and Systems Sciences. 1998;168:133–166.
[24] Neal RM. Monte carlo implementation of Gaussian process models for bayesian regression and classification. arXiv preprint physics/9701026.
[25] Williams CKI, Rasmussen CE. Gaussian processes for regression. in: Advances in Neural Information Processing Systems. 1996;514–520.
[26] Bachoc F. Cross Validation and Maximum Likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. Computational Statistics & Data Analysis. 2013;66:55–69.
[27] Butler A, Haynes RD, Humphries TD, Ranjan P. Efficient optimization of the likelihood function in Gaussian process modelling. Computational Statistics & Data Analysis. 2014;73:40–52.
[28] Chen Z, Wang B. How priors of initial hyperparameters affect Gaussian process regression models. Neurocomputing. 2018;275:1702–1710.
[29] Heyde CC. Quasi-Likelihood and its Application. Springer, New York; 1997.
[30] Kitanidis PK. Introduction to Geostatistics: Applications in Hydrology. Cambridge University Press, New York; 1997.
[31] Majdisova Z, Skala V. Radial basis function approximations: comparison and applications. Applied Mathematical Modelling. 2017;51:728–743.
[32] Flaxman S, Gelman A, Neill D, Smola A, Vehtari A, Wilson AG. Fast hierarchical gaussian processes. 2015.
[33] Fasshauer GE. Meshfree Approximation Methods with Matlab. World Scientific, Singapore; 2007.
[34] Schoenberg I. Metric spaces and completely monotone functions. Annals of Mathematics. 1938;39:811–841.
[35] Wendland H. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. Advances in Computational Mathematics. 1995;4:389–396.
[36] Wu Z. Multivariate Compactly Supported Positive Definite Radial Functions. Advances in Computational Mathematics. 1995;4:283–292.
[37] Schaback R. Creating surfaces from scattered data using radial basis functions. Mathematical methods for curves and surfaces, Vanderbilt University Press, Nashville. 1995;477–496.
[38] Wendland H. Scattered data approximation. Cambridge University Press, Cambridge; 2005.
[39] Wilson AG. Covariance Kernels for Fast Automatic Pattern Discovery and Extrapolation with Gaussian Processes. Ph.D. Thesis, University of Cambridge; 2014.
[40] Watkins DS. Fundamentals of Matrix Computations. Wiley Series in Pure and Applied Mathematics; 2010.