Journal of Machine Learning Research 23 (2022) 1-30 Submitted 2/22; Revised 9/22; Published 9/22

Scalable Gaussian-process regression and variable selection using Vecchia approximations

Jian Cao JIAN.CAO@TAMU.EDU
Department of Statistics and Institute of Data Science
Texas A&M University
College Station, TX 77843, USA

Joseph Guinness GUINNESS@CORNELL.EDU
Department of Statistics
Cornell University
Ithaca, NY 14853, USA

Marc G. Genton MARC.GENTON@KAUST.EDU.SA
Statistics Program
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia

Matthias Katzfuss KATZFUSS@TAMU.EDU
Department of Statistics
Texas A&M University
College Station, TX 77843, USA

Editor: Xiaotong Shen

Abstract

Gaussian process (GP) regression is a flexible, nonparametric approach to regression that naturally quantifies uncertainty. In many applications, the number of responses and covariates are both large, and a goal is to select covariates that are related to the response. For this setting, we propose a novel, scalable algorithm, coined VGPR, which optimizes a penalized GP log-likelihood based on the Vecchia GP approximation, an ordered conditional approximation from spatial statistics that implies a sparse Cholesky factor of the precision matrix. We traverse the regularization path from strong to weak penalization, sequentially adding candidate covariates based on the gradient of the log-likelihood and deselecting irrelevant covariates via a new quadratic constrained coordinate descent algorithm. We propose Vecchia-based mini-batch subsampling, which provides unbiased gradient estimators. The resulting procedure is scalable to millions of responses and thousands of covariates. Theoretical analysis and numerical studies demonstrate the improved scalability and accuracy relative to existing methods.

Keywords: adaptive bridge penalty; gradient-based variable selection; mini-batch subsampling; ordered conditional approximation; penalized Gaussian regression

1. Introduction

Gaussian process regression Many tasks in statistics and machine learning can be viewed as regression problems, with the goal of inferring the functional relationship between

a response and a number of covariates. Gaussian processes (GPs) are an attractive choice for modeling the regression function (e.g., Rasmussen and Williams, 2006), as they naturally quantify uncertainty, they can flexibly capture nonlinear and nonparametric behavior, they are interpretable, and much of the resulting inference involves closed-form expressions. We focus on GP regression for datasets with a large number of responses, n, and a large number of covariates, d, under the assumption that only few covariates, $d_0 \ll d$, are useful for predicting the response. In this setting, our goals are variable selection, model estimation, and subsequent prediction based on the selected sparse model.

Existing approaches for large n Basic GP regression scales poorly to large n or d. Many approaches have been proposed that deal with one or both of these issues. The challenge with large n is that direct GP inference requires $\mathcal{O}(n^3)$ time. Heaton et al. (2019) and Liu et al. (2020) provide reviews of methods that tackle the large-n problem in spatial statistics and machine learning, respectively. These methods include fully (e.g., Quiñonero-Candela and Rasmussen, 2005; Banerjee et al., 2008; Finley et al., 2009) and partially (e.g., Snelson and Ghahramani, 2007; Sang et al., 2011) independent conditional (FIC/PIC) approximations, but these low-rank approaches can have limitations in many settings (e.g., Stein, 2014), even when optimizing over pseudo-inputs (Hensman et al., 2015). Other GP approximations, such as multi-level PIC (Katzfuss, 2017; Katzfuss and Gong, 2020), approximations based on stochastic partial differential equations (Lindgren et al., 2011), distributed GPs (Deisenroth and Ng, 2015) or KISS-GP (Wilson and Nickisch, 2015), can struggle with high input dimension d.

The Vecchia approximation A highly promising approach to scaling GP inference to large n may be the Vecchia approximation (Vecchia, 1988), which has become very popular in spatial statistics (e.g., Stein et al., 2004; Datta et al., 2016; Guinness, 2018; Katzfuss and Guinness, 2021; Katzfuss et al., 2020), but which has not received much attention in machine learning. This approach can be viewed as an ordered conditional approximation, in which the joint density of the GP response is approximated as a product of univariate conditional distributions. The resulting approximation can be highly accurate even with small conditioning sets. Katzfuss et al. (2022) proposed a scaled Vecchia approximation that further improves the accuracy of the Vecchia approximation and used it for GP emulation of expensive computer experiments in $d = \mathcal{O}(10)$ dimensions. A more detailed review of Vecchia approximations will be provided in Section 2.2.

Existing approaches for large d There has also been extensive work on scaling GPs to moderate or high input dimension d. Moderate d can be handled by variable selection using automatic relevance determination (ARD) kernel functions (Neal, 1996) and Bayesian model selection (Dearmon and Smith, 2016; Posch et al., 2021). However, for larger d (say $d \gg 100$), these methods are not sufficiently scalable due to computation and convergence issues caused by the high dimensionality of the parameter space. For such high dimensions, existing approaches include penalized GP regression (e.g., Yi et al., 2011), manifold GP regression (e.g., Calandra et al., 2016), and hierarchical diagonal sampling (HDS; e.g., Chen et al., 2012). However, both penalized GP and manifold GP regressions consider all covariates simultaneously, leading to $\mathcal{O}(d)$ optimization parameters, which may negatively impact model inference in three aspects, namely convergence to local optima, over-fitting,

and computational inefficiency. Furthermore, Yi et al. (2011) and Calandra et al. (2016) optimized the exact GP likelihood, not scalable with respect to n, while HDS assumes that responses are sampled where needed, mainly addressing Bayesian optimization instead of GP regression.

Large numbers of responses and covariates Several methods have been proposed to handle large n and d by approximating the GP using FIC and transforming and reducing the dimension of the input domain, such as randomly-projected additive GPs (Delbridge et al., 2020), deep kernel learning (Wilson et al., 2016), and dimension reduction with pseudo-inputs (Snelson and Ghahramani, 2006). These approaches mainly achieve dimension reduction rather than variable selection. To our knowledge, none of the existing approaches is suitable for our goal of simultaneous variable selection and GP regression for large n and large d.

The VGPR algorithm Here we propose the VGPR algorithm, for Vecchia GP Regreession, which is highly scalable in n and d. Specifically, to handle large n, we extend the scaled Vecchia GP approximation (Katzfuss et al., 2022) and propose Vecchia-based mini-batch subsampling, which provides unbiased gradient estimators. To achieve variable selection for large d, we consider a penalized Vecchia-GP loglikelihood, and we traverse the regularization path from strong to weak penalization, sequentially adding candidate covariates based on the gradient of the log-likelihood and deselecting irrelevant covariates through a new quadratic constrained coordinate descent algorithm (QCCD). QCCD builds a quadratic approximation of the objective function at each iteration and applies constrained coordinate descent to find the constrained quadratic optimum. Compared with existing GP regression methods such as Yi et al. (2011) and Katzfuss et al. (2022), traversing the regularization path with warm starts effectively avoids local optima while QCCD can reach boundary values, achieving covariate deselection without artificial thresholding. We provide theoretical and numerical evidence for our gradient-based variable selection. The dominant complexity of VGPR depends linearly on the batch size and quadratically on the number of selected covariates (as opposed to the total number of responses or covariates).

Outline In Section 2, we briefly review ARD kernels and the scaled Vecchia approximation. Section 3 introduces our new VGPR algorithm that involves the QCCD subroutine, the choice of the penalty function, the selection of covariates based on the gradient, and a mini-batch sampling technique specific to the Vecchia approximation. In Section 4, we compare VGPR with state-of-the-art GP regressions in terms of posterior inference and variable selection based on simulated GP datasets. Section 5 provides a comparison with methods commonly used in machine learning for variable selection and prediction based on real datasets, including an example with $n = 10^6$ and $d = 10^3$. Section 6 concludes the paper. The code for replicating the numerical results in this paper are published at `https://github.com/katzfuss-group/Vecchia_GPR_var_select`.

2. Review

2.1 GP regression and ARD kernels

We consider the standard GP regression model (e.g., Rasmussen and Williams, 2006):

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \qquad i = 1, \ldots, n,$$

where y_i is the i-th response observed at the d-dimensional covariate vector $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$, $f(\cdot) \sim \mathcal{GP}(0, K)$ is a GP with zero mean and a positive-definite covariance or kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and $\{\epsilon_i \sim \mathcal{N}(0, \tau^2)\}$ are independent noise terms. Then, the vector of responses, $\mathbf{y} = \big(y_1, \ldots, y_n\big)^\top$, at input values $\mathbf{x}_1, \ldots, \mathbf{x}_n$ follows an n-variate Gaussian distribution, $\mathcal{N}_n(\mathbf{0}, \boldsymbol{\Sigma})$, with covariance matrix $\boldsymbol{\Sigma} = \big(K(\mathbf{x}_i, \mathbf{x}_j)\big)_{i,j=1,\ldots,n} + \tau^2 \mathbf{I}_n$, whose (i, j)-th entry describes the covariance between responses y_i and y_j as a function of their corresponding covariate vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Throughout, we assume a centered response vector $\mathbf{y}$ and a zero mean structure; if desired, a (non-zero) linear mean structure can be profiled out during maximum likelihood estimation (Guinness, 2021).

An automatic relevance determination (ARD) kernel (Neal, 1996) is an anisotropic kernel that assigns each covariate a separate parameter, controlling its impact in the covariance structure. Specifically, we assign a separate relevance (i.e., inverse range) parameter $r_l \geq 0$ to each input dimension l:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tilde{K}(q^{\mathbf{r}}(\mathbf{x}_i, \mathbf{x}_j)), \qquad q^{\mathbf{r}}(\mathbf{x}_i, \mathbf{x}_j)^2 = \textstyle\sum_{l=1}^d r_l^2 (x_{i,l} - x_{j,l})^2, \tag{1}$$

where the superscript $\mathbf{r}$ emphasizes the dependence of the distance q on the relevances $\mathbf{r} = (r_1, \ldots, r_d)^\top$. Note that $r_l = 0$ is equivalent to deselecting the l-th covariate. In (1), $\tilde{K}$ can be any isotropic kernel that is valid in $\mathbb{R}^d$; for our numerical results, we used a Matérn covariance kernel with smoothness 2.5 as recommended in Chapter 4 of Rasmussen and Williams (2006):

$$\tilde{K}(q) = \sigma^2 (1 + q + q^2/3) \exp(-q), \tag{2}$$

where σ^2 is the variance parameter. Our model depends on unknown parameters $\boldsymbol{\theta} = (\sigma^2, \mathbf{r}^2, \tau^2)$, whose inference is usually achieved by maximum likelihood estimation (MLE). We denote by $\mathbf{r}^2$ the element-wise square of $\mathbf{r}$; we use the squared relevance (SR) as the optimization parameters for the purpose of variable selection, which is explained in Section 3. Computing the exact GP density, $p_{\boldsymbol{\theta}}(\mathbf{y}) = \mathcal{N}_n(\mathbf{y} | \mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$, requires $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory, often becoming infeasible for $n > 10{,}000$.

2.2 Review of scaled Vecchia

We use the (scaled) Vecchia approximation to tackle GP regressions with large n (e.g., $n > 10^4$), because it can achieve higher approximation accuracy while having the same linear complexity compared with other state-of-the-art GP approximations. The original Vecchia approximation (Vecchia, 1988) starts from the conditional representation of the density function, $p_{\boldsymbol{\theta}}(\mathbf{y}) = \prod_{i=1}^n p_{\boldsymbol{\theta}}(y_i | \mathbf{y}_{1:(i-1)})$, and truncates the conditioning sets to sets $c(i)$ with a maximum of $m \ll n$ elements:

$$\hat{p}_{\boldsymbol{\theta}}(\mathbf{y}) = \textstyle\prod_{i=1}^n p_{\boldsymbol{\theta}}(y_i | \mathbf{y}_{c(i)}) = \mathcal{N}_n(\mathbf{0}, \hat{\boldsymbol{\Sigma}}). \tag{3}$$

The Vecchia approximation has several attractive properties. It partitions the n-dimensional GP density into n computationally independent univariate conditional densities, and hence results in n parallel computations each requiring only $\mathcal{O}(m^3)$ time, where even small $m \ll n$ can achieve high accuracy due to the screening effect (Stein, 2011). As indicated by (3), the approximation also implies a joint Gaussian distribution, whose inverse Cholesky factor $\hat{\boldsymbol{\Sigma}}^{-1/2}$ is sparse with fewer than nm nonzero entries (e.g., Katzfuss and Guinness, 2021). Furthermore, Vecchia approximation produces the smallest KL divergence from $p_{\boldsymbol{\theta}}(\mathbf{y})$ subject to certain sparsity constraints on $\hat{\boldsymbol{\Sigma}}^{-1/2}$ (Schäfer et al., 2021a) and can achieve ϵ-accurate approximations with $m = \mathcal{O}(\log^d(n))$ for certain Matérn-type kernels up to edge effects (Schäfer et al., 2021a).

The accuracy of Vecchia approximations depends on the ordering of $\mathbf{y}$ and the choice of $\{c(i)\}$; the scaled Vecchia approximation in Katzfuss et al. (2022) takes varying relevances of the covariates into account. Specifically, the scaled Vecchia approximation uses the maximum-minimum distance ordering (MM) and the nearest-neighbor conditioning (NN) based on the scaled distances $q^{\mathbf{r}}(\mathbf{x}_i, \mathbf{x}_j)$ between y_i and y_j. MM is a sequential ordering that selects each response to maximize the minimum distance toward previous responses in the ordering, and NN chooses the $\min(i-1, m)$ nearest responses of y_i among $\{y_1, \ldots, y_{i-1}\}$ as $\mathbf{y}_{c(i)}$. MM and NN can be obtained in quasilinear time in n (Schäfer et al., 2021b,a). We use $\hat{p}^{\tilde{\mathbf{r}}}_{\boldsymbol{\theta}}(\mathbf{y})$ to represent the scaled Vecchia likelihood evaluated at $\boldsymbol{\theta}$ with MM and NN computed based on $q^{\tilde{\mathbf{r}}}$, where $\tilde{\mathbf{r}}$ does not necessarily have to take on the same values as the $\mathbf{r}$ indicated by $\boldsymbol{\theta}$.

Another attractive property of the Vecchia approximation is that many existing GP approximations, including FIC and PIC, can be viewed as its special cases corresponding to particular choices of the ordering and conditioning (Katzfuss and Guinness, 2021); however, the scaled MM and NN choices in scaled Vecchia can be much more accurate. To demonstrate this, we used a numerical experiment to compare FIC, FITC (with optimized pseudo-inputs), PIC, Vecchia (with MM and NN based on $q^{\mathbf{1}}$), and scaled Vecchia approximations in terms of their KL divergence from an exact multivariate Gaussian distribution (see details in Appendix A). Figure 1 shows the results for the comparison with $n = 5{,}000$, $d = 10$, $\sigma^2 = 1$, $\mathbf{r} = (10, 5, 2, 1, 0.5, 0, \ldots, 0)^\top$, $\tau^2 = 0$, averaged over ten repetitions. While Vecchia without scaling outperformed FIC, FITC, and PIC, the scaled Vecchia approach, which will be used in our proposed methods below, resulted in additional improvements of several orders of magnitude.

The construction of the conditioning sets $c(i)$ in the scaled Vecchia approximation can be also applied to posterior prediction to achieve an $\mathcal{O}(m^3)$ complexity at each unknown location. Specifically, the m nearest in-sample neighbors of an unknown location based on $q^{\tilde{\mathbf{r}}}$ is defined as its conditioning set, based on which the conditional mean and variance is computed. Fast computation of the joint posterior predictive distribution at a large set of test inputs is also possible (Katzfuss et al., 2020).

2.3 Gradient and Fisher information

The (penalized) negative log-likelihood, $h^{\tilde{\mathbf{r}}}_{\lambda}(\boldsymbol{\theta}) = -\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}) + w_\lambda(\boldsymbol{\theta})$ is typically used as the objective function for parameter inference in GP regression, where here $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}) = \log \hat{p}^{\tilde{\mathbf{r}}}_{\boldsymbol{\theta}}(\mathbf{y})$ is the log-likelihood under the scaled Vecchia approximation and $w_\lambda(\boldsymbol{\theta})$ is a penalty func-

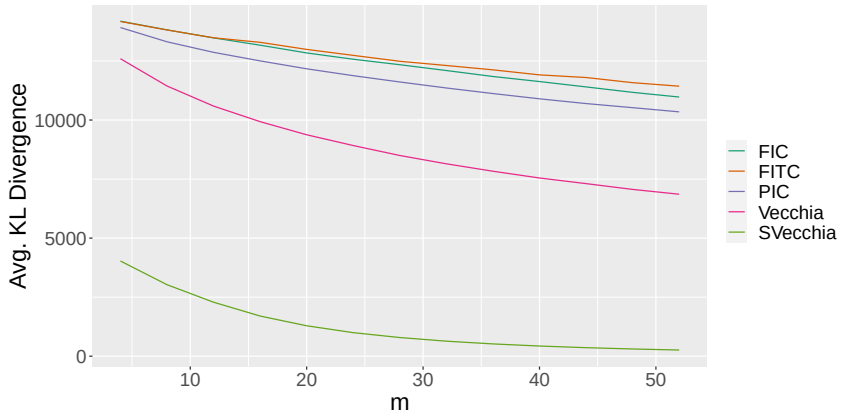


Figure 1: Approximation accuracy (in terms of KL divergence from the true GP density) versus conditioning-set size, for five GP approximations, namely FIC, FITC, PIC, Vecchia and Scaled Vecchia (SVecchia)

tion whose magnitude increases with λ. Under the Vecchia approximation, not only the log-likelihood but also its first- and second-order information can be computed in parallel and at linear complexity in n. Specifically, $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ can be decomposed into the sum of n computationally independent terms:

$$h_\lambda^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}) = -\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}) + w_\lambda(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \left(\log p_{\boldsymbol{\theta}}(\mathbf{y}_{\{i\}\cup c(i)}) - \log p_{\boldsymbol{\theta}}(\mathbf{y}_{c(i)})\right) + w_\lambda(\boldsymbol{\theta}). \tag{4}$$

Based on this expression involving a sum of (log) Gaussian densities, it is straightforward to compute the gradient $\hat{\mathbf{g}}_{\boldsymbol{\theta}}^{\tilde{\mathbf{r}}}$ and the Fisher information matrix (FIM) $-\hat{\mathbf{H}}_{\boldsymbol{\theta}}^{\tilde{\mathbf{r}}}$ of $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$. Notice that $\hat{\mathbf{H}}_{\boldsymbol{\theta}}^{\tilde{\mathbf{r}}}$ can be used as a surrogate of the Hessian matrix. The computations of $\hat{\mathbf{g}}_{\boldsymbol{\theta}}^{\tilde{\mathbf{r}}}$ and $\hat{\mathbf{H}}_{\boldsymbol{\theta}}^{\tilde{\mathbf{r}}}$ are $\mathcal{O}(nm^3d)$ and $\mathcal{O}(nm^2d^2)$, respectively, based on the closed-form formula for multivariate normal gradient and FIM; refer to Guinness (2021) and the R package 'GpGp' (Guinness, 2018) for the computation details.

The availability of the second-order information under the Vecchia approximation benefits the convergence rate of parameter inference. Along this direction, a state-of-the-art method is the Fisher scoring algorithm proposed in Guinness (2021) that substitutes the Hessian matrix in the natural gradient descent with FIM to achieve a quadratic convergence rate:

$$\boldsymbol{\theta}^{(\iota+1)} = \boldsymbol{\theta}^{(\iota)} - \left(\hat{\mathbf{H}}_{\boldsymbol{\theta}^{(\iota)}}^{\tilde{\mathbf{r}}^{(\iota)}}\right)^{-1} \hat{\mathbf{g}}_{\boldsymbol{\theta}^{(\iota)}}^{\tilde{\mathbf{r}}^{(\iota)}}, \tag{5}$$

where the superscript denotes the parameter estimates at the ι-th iteration. However, it is not ideal for constrained optimization. Specifically, Fisher scoring uses variable transformation (e.g., logarithm) to enforce positivity constraints, and so it is typically impossible for optimization parameters to reach boundary values (i.e., zero), which is crucial for variable deselection. We introduce a new second-order optimization algorithm that addresses this limitation in Section 3.5.

3. Scalable GP Regression and Variable Selection

3.1 Overview of VGPR

Algorithm 1 contains a high-level overview of our VGPR algorithm for scalable variable selection and model estimation in GP regression, with subsequent sections providing details and theoretical and numerical support. VGPR traverses the regularization path of the penalized log-likelihood from strong to weak penalization until a stopping criterion based on an out-of-sample (OOS) score is reached (Section 3.2). For a given penalization level, VGPR conducts a forward-backward-selection procedure (Section 3.3), which iteratively adds covariates to a candidate set based on the gradient with respect to the squared relevances (Section 3.4) and deselects covariates through QCCD optimization (Section 3.5). We introduce an iterative adaptive bridge penalty (Section 3.6) and provide further speed-ups via an unbiased mini-batch subsampling method (Section 3.7), resulting in a computational complexity that is essentially independent from n and d (Section 3.8).

Algorithm 1: VGPR

Input: $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}), w_\lambda(\boldsymbol{\theta}), \lambda_0, k$

1: Initialize $\boldsymbol{\theta}$ with $\mathbf{r}$ set to $\mathbf{0}^+$ and $\tilde{\mathbf{r}} \leftarrow \mathbf{r}, \lambda \leftarrow \lambda_0, \zeta \leftarrow \phi$
2: **while** OOS score improves **do**
3: $\quad(\boldsymbol{\theta}, \zeta) \leftarrow$ forward-backward$(\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}), w_\lambda(\boldsymbol{\theta}), \boldsymbol{\theta}, \zeta, k)$ — see Alg. 2
4: $\quad$Reduce λ
5: **end while**

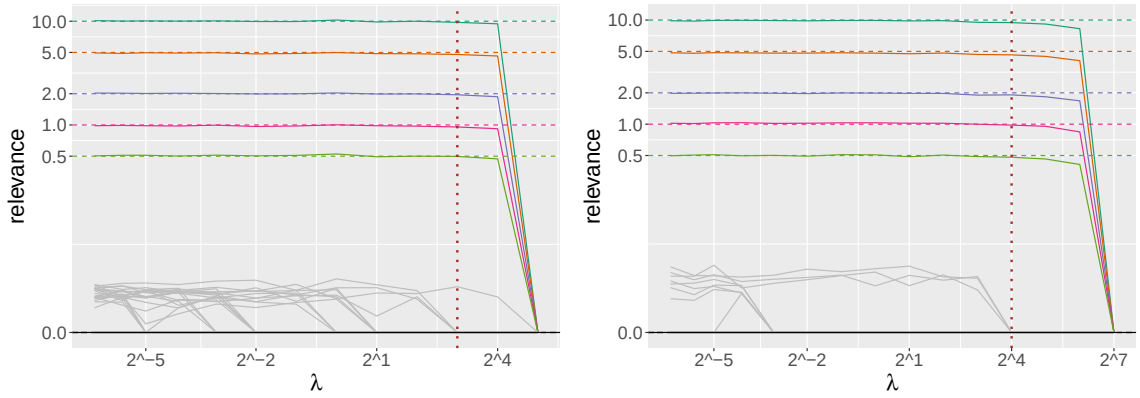
3.2 Traversing the regularization path

VGPR traverses the regularization path of the penalized log-likelihood, $h_\lambda^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$, from strong to weak penalization (i.e., large to small λ) until a stopping criterion based on an out-of-sample (OOS) score is reached. We recommend starting with a penalty strength of $\lambda_0 = n$, which is typically sufficient to imply a completely sparse model without any covariates selected. (Otherwise, we simply increase λ exponentially until a fully sparse model is obtained.) The regularization path is constructed over a decreasing series of λ, for example, a geometric series with a common ratio of $1/2$. VGPR stops when an out-of-sample (OOS) score such as mean-squared error fails to show obvious improvement.

Figure 2 illustrates the regularization path computed by VGPR using $n = 10^4$ responses and $d = 10^3$ covariates under the bridge penalty (see Section 3.6). The covariance kernels used for dataset simulation in Sections 3 and 4 are the Matérn covariance kernel defined by (2) and parameterized by:

$$\sigma^2 = 1, \quad \tau^2 = 0.05^2, \quad [r_1^2, r_2^2, r_3^2, r_4^2, r_5^2] = [10^2, 5^2, 2^2, 1^2, 0.5^2], \quad r_l^2 = 0 \text{ if } l > 5, \tag{6}$$

unless specified otherwise. The covariates are generated either independently from the Latin hypercube or dependently from a multivariate normal distribution with a constant correlation of 0.9 and normalized to have a standard deviation of one. Our Vecchia approximation uses a maximum conditioning set size of $m = 100$. A quarter of the responses were set aside to compute the OOS RMSE based on which, the stopping condition was defined as that the OOS RMSE improves less than 1% after any new covariate is selected. The OOS



(a) Independent covariates

(b) Dependent covariates

Figure 2: Regularization path computed by VGPR using simulated independent or dependent covariates. The relevance parameters of the true covariates are color-coded, and their true values are marked by horizontal colored dashed lines. The fake covariates, whose true relevance parameters are zero, are colored in grey. The vertical red dotted lines mark the optimal model indicated by the stopping condition.

sample size of $n/4$ and the 1% OOS score threshold are used throughout this paper and are generally recommended as default values.

In Figure 2, the true covariates and their relevance parameters were correctly selected and well estimated, respectively. All fake covariates, except for one when using independent covariates, were filtered out, highlighting the efficacy of VGPR in variable selection even given a large pool of highly correlated covariates. Moreover, the number of optimization parameters was always kept at $\mathcal{O}(d_0)$ until the stopping condition was reached. Also due to the small number of optimization parameters, VGPR completed the model estimation within minutes.

3.3 Forward-backward selection

To keep the "active" set of covariates small when running optimization, VGPR keeps a candidate set of covariates $\zeta \subset \{1, 2, \ldots, d\}$ representing the covariates currently selected. Assuming model sparsity, the size of ζ can be kept much smaller than d. Given a current ζ, standard forward selection would fit $\mathcal{O}(d)$ models with covariates $\zeta \cup l$ for each $l \notin \zeta$, but this procedure is prohibitively expensive for large d.

Instead, we propose a forward-backward-selection algorithm, provided in Algorithm 2, to find the optimal model under each λ. The algorithm iteratively performs a forward step and a backward step. The forward step adds to ζ a small number k of "promising" covariates corresponding to the k largest entries in the squared-relevance gradient (SR-gradient), given by the derivatives of $\ell(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathbf{y})$ with respect to each r_l^2 with $l \notin \zeta$, evaluated at the current estimates of the relevances (i.e., $r_l = 0$ for $l \notin \zeta$). For example, we set $k = 3$ in Figure 2. We provide numerical and theoretical support for the forward step in Section 3.4. After the forward step, we run a backward step on the new ζ via our QCCD algorithm (see Section 3.5), which finds the new parameter estimates using a warm start based on the previous estimates and potentially deselects covariates by returning estimates of zero for some SRs. The forward-backward procedure for a given λ value stops (and VGPR moves

Algorithm 2: Forward-backward selection

Input: $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}), w_\lambda(\boldsymbol{\theta}), \boldsymbol{\theta}, \zeta, k$

1: **while** OOS score improves **do**
2: $\mathcal{S} \leftarrow$ mini-batch subsampling, $\hat{\mathbf{g}}_{\mathbf{r}^2} \leftarrow \frac{\partial \hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta} \mid \mathcal{S})}{\partial \mathbf{r}^2}$
3: Define $\Delta\zeta$ as the indices of the k largest coefficients in $\hat{\mathbf{g}}_{\mathbf{r}^2}[-\zeta]$
4: $\zeta \leftarrow \zeta \bigcup \Delta\zeta$, initialize $\mathbf{r}[\Delta\zeta]$, $\tilde{\mathbf{r}} \leftarrow \mathbf{r}$
5: $h^{\tilde{\mathbf{r}}}_{\lambda,\zeta}(\boldsymbol{\theta}_\zeta) \leftarrow \hat{\ell}^{\tilde{\mathbf{r}}}_{\zeta}(\boldsymbol{\theta}_\zeta) + \lambda w_\lambda(\boldsymbol{\theta}_\zeta)$, $\boldsymbol{\theta}_\zeta \leftarrow \text{QCCD}(h^{\tilde{\mathbf{r}}}_{\lambda,\zeta}, \boldsymbol{\theta}_\zeta, \mathbf{0})$ — see Alg. 3
6: Remove covariates with zero relevance from ζ
7: **end while**
8: **return** $\boldsymbol{\theta}$ and ζ

on to a smaller λ) based on the same stopping criterion as in Algorithm 1, using the OOS score.

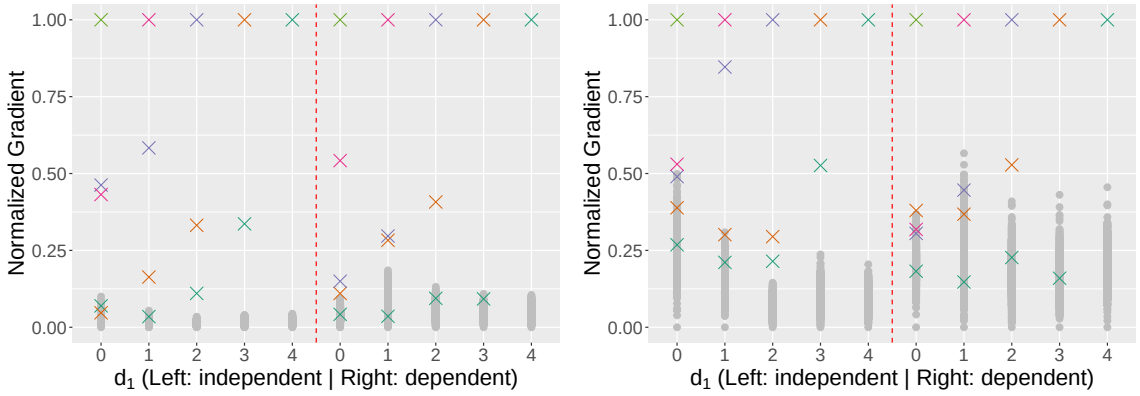
We now provide more notational details on Algorithm 2. We use square brackets for indexing, with negative indices corresponding to dropped elements. The parts in blue font (in all algorithms) provide the mini-batching modifications to be discussed in Section 3.7. MM and NN are implicitly updated at each occurrence of $\tilde{\mathbf{r}} \leftarrow \mathbf{r}$, which improves the accuracy of the scaled Vecchia approximation $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$. In Line 5, we use the ζ subscript to indicate the parameter vector, the log-likelihood function, and the objective function defined over the subset of covariates in ζ, as opposed to all covariates, which reduces the number of parameters involved in QCCD. Notice that $\boldsymbol{\theta}_\zeta$ is viewed as a subvector of $\boldsymbol{\theta}$ and the assignment to the former indicates changes to the latter as well, which implies warm starts and avoids local optima.

3.4 Numerical and theoretical support for gradient-based covariate selection

The SR-gradient can be used to order the covariates' relevance levels in the ARD model. Specifically, assuming that the SRs of the covariates in ζ are fixed at their correct values, the derivatives of $\ell(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathbf{y})$ with respect to the remaining SRs evaluated at zero can be used to rank the unselected covariates in ζ^{C}.

We illustrate this idea using an example of selecting $d_0 = 5$ true covariates from $d = 10^3$ total covariates, shown in Figure 3, with σ^2 and τ^2 are fixed at their true values. It is evident that true covariates (with $r_{l0}^2 > 0$) have bigger coefficients in the SR-gradient. In fact, the magnitudes of the coefficients reflect the magnitudes of $\{r_{l0}^2\}$. This conclusion is valid even assuming strong dependence among covariates or using the gradient under mini-batch subsampling (see Section 3.7). The full dataset has $n = 5{,}000$ responses and the mini-batch size is $\check{n} = 128$. We used the derivatives under the scaled Vecchia approximation (i.e., $\hat{\ell}^{\tilde{r}}(\boldsymbol{\theta})$) to substitute those of $\ell(\boldsymbol{\theta})$, indicating sufficient accuracy from the scaled Vecchia approximation.

In the remainder of this section, we provide theoretical support for why the SR-gradient can be used for variable selection. The following notations are used in the theoretical results



(a) Normalized gradient without mini-batching

(b) Normalized gradient with mini-batching

Figure 3: Relative magnitudes of the coefficients in the SR-gradient. The number of covariates $d = 10^3$, among which the first five are true (i.e., with positive true SRs). For each $d_1 = 0, 1, 2, 3, 4$, we assume that only the first d_1 true covariates are selected and their SRs are correctly estimated. The SRs of unselected covariates are zero. The coefficients in the gradient are normalized to $[0, 1]$. The first five coefficients in the SR-gradient are marked by colored crosses and the rest by grey dots. Only coeffcients corresponding to unselected covariates are plotted to align with goal of variable selection. The red dashed line separates scenarios with independent and dependent covariates. Notice that some colored crosses are covered by grey dots and that the coefficients for unselected true covariates were typically bigger than those for fake covariate.

and their derivations:

$\mathbf{r}_0$	the true relevance vector $[r_{10}, r_{20}, \ldots, r_{d0}]$
d_0	the number of true covariates (i.e., $r_{l0} > 0$ if $l \leq d_0$ $r_{l0} = 0$ otherwise)
d_1	an integer between 0 and d_0, $0 < d_1 < d_0 < d$
$\mathbf{r}_1$	$[r_{10}, \ldots, r_{d_1 0}, 0, \ldots, 0]^\top$
$(\sigma_0, \tau_0), (\sigma_1, \tau_1)$	the true and an arbitrary values for (σ, τ)
$\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1$	covariance matrix and its values evaluated at $(\sigma_0, \mathbf{r}_0, \tau_0)$ and $(\sigma_1, \mathbf{r}_1, \tau_1)$
$\tilde{\boldsymbol{\Sigma}}, \tilde{\boldsymbol{\Sigma}}_0, \tilde{\boldsymbol{\Sigma}}_1$	correlation matrix and its values evaluated at $(1, \mathbf{r}_0, 0)$ and $(1, \mathbf{r}_1, 0)$

In this section, the expectations are taken with respect to both $\{\mathbf{x}_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$.

Proposition 1 *Assume that* $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-q^{\mathbf{r}}(\mathbf{x}_i, \mathbf{x}_j)^2)$ *and that* $\{x_{il}\}_{i=1,\ldots,n, l=1,\ldots,d}$ *have i.i.d. normal or uniform distributions. When evaluated at* $(\sigma_1, \mathbf{r}_1, \tau_1)$, $E[\frac{\partial \ell}{\partial r_{l_1}^2}] > E[\frac{\partial \ell}{\partial r_{l_2}^2}]$.

Proposition 1 suggests that under the squared exponential kernel, when an arbitrary number of SRs are at their true values while the others at zero, the order of the SR-gradient coefficients indicates the relevance order of the covariates. While the condition on $\mathbf{r}_1$ in Proposition 1 is somewhat restrictive, we conjecture that when the gradient is evaluated at $\mathbf{r}$ no greater than $\mathbf{r}_0$ coefficient-wise, the above conclusion still holds, which can be readily shown if we assume $E[\frac{\partial \ell}{\partial r_{l_1}^2}] - E[\frac{\partial \ell}{\partial r_{l_2}^2}]$, evaluated at $(\sigma_1, \mathbf{r}, \tau_1)$, changes monotonically with each coefficient in $\mathbf{r}$. In general, numerical examples in Sections 3.1, 3.7, and 4 suggest that the conditions in Proposition 1 can be relaxed and the result still holds.

Based on Proposition 1, two corollaries addressing the initialization of $\mathbf{r}$ and correlated fake covariates, respectively, can be derived.

Corollary 2 *Assume that* $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-q^{\mathbf{r}}(\mathbf{x}_i, \mathbf{x}_j)^2)$ *and that* $\{x_{il}\}_{i=1,\ldots,n, l=1,\ldots,d}$ *have i.i.d. normal or uniform distributions. When evaluated at* $\mathbf{r} \to \mathbf{0}$*:*

$$E[\tfrac{\partial \ell}{\partial r_{l_1}^2}] \geq E[\tfrac{\partial \ell}{\partial r_{l_2}^2}] \text{ for } 0 < l_1 \leq d_0 \text{ and } d_0 < l_2 \leq d.$$

Corollary 3 *Assume that* $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-q^{\mathbf{r}}(\mathbf{x}_i, \mathbf{x}_j)^2)$ *and that* $\{x_{il}\}_{i=1,\ldots,n, l=1,\ldots,d}$ *have i.i.d. normal distributions. Let* $\mathbf{x}_{d+1}$ *be a new covariate constructed as* $\rho_1 \mathbf{x}_{l_1} + \rho_2 \mathbf{x}_{l_2}$ *with* $d_1 < l_1 \leq d_0 < l_2 \leq d$*,* $\rho_2 > 0$*, and* $\rho_1^2 + \rho_2^2 = 1$*. When evaluated at* $(\sigma_1, \mathbf{r}_1, \tau_1)$*:*

$$E[\tfrac{\partial \ell}{\partial r_{l_1}^2}] > E[\tfrac{\partial \ell}{\partial r_{d+1}^2}].$$

Noticing that $\zeta = \phi$ can be closely approximated by $\mathbf{r} \to \mathbf{0}$, Corollary 2 indicates that SRs should be initialized to small magnitudes but big enough to avoid numerical singularity (e.g., 10^{-8}), which is denoted by $\mathbf{0}^+$ in Algorithm 1. Corollary 3 suggests that the order of the SR-gradient coefficients can distinguish fake covariates that are correlated with true covariates. Theoretical support for the previous proposition becomes more challenging under general covariance kernels, due to the lack of the separability property and the straight-forward derivative formula. Proposition 4 aims to reach the same conclusion for general ARD kernels but uses a first-order approximation of $\mathbf{\Sigma}_0$.

Proposition 4 *Assume that* $\{x_{il}\}_{i=1,\ldots,n, l=1,\ldots,d}$ *have i.i.d. distributions and that* $\{r_{l0}\}_{l=d_1+1}^{d_0}$ *are small enough s.t.* $\tilde{\mathbf{\Sigma}}_0$ *can be closely approximated by the first-order Taylor expansion of* $\tilde{\mathbf{\Sigma}}$ *at* $\mathbf{r}_1$*:*

$$\tilde{\mathbf{\Sigma}}_0 \approx \tilde{\mathbf{\Sigma}}_1 + \textstyle\sum_{l=d_1+1}^{d_0} \frac{\partial \tilde{\mathbf{\Sigma}}}{\partial r_l^2}\Big|_{\mathbf{r}=\mathbf{r}_1} r_{l0}^2.$$

When evaluated at $(\sigma_1, \mathbf{r}_1, \tau_1)$*:*

$$E[\tfrac{\partial \ell}{\partial r_{l_1}^2}] > E[\tfrac{\partial \ell}{\partial r_{l_2}^2}] \text{ for } d_1 < l_1 \leq d_0 \text{ and } d_0 < l_2 \leq d.$$

The first-order approximation typically holds when $r_{l0} \to 0, l = d_1 + 1, \ldots, d_0$ while on the other hand, the expectation of the derivative of r_l^2 evaluated at $r_l = 0$ is intuitively positively correlated with r_{l0} (i.e., $E[\frac{\partial \ell}{\partial r_l^2}]\Big|_{r_l=0} \nearrow r_{l0}$). The two aspects collectively support that the order of the gradient coefficients is indicative for the order of relevance levels of the covariates under general ARD covariance kernels.

3.5 Quadratic constrained coordinate descent

We introduce our quadratic constrained coordinate descent (QCCD) algorithm in the context of minimizing a general objective function $h(\boldsymbol{\theta})$, whose gradient and (positive-definite) negative FIM, denoted by $\mathbf{g}$ and $\mathbf{H}$, respectively, can be computed. QCCD is described in Algorithm 3 with the assumption that parameter constraints are given by their lower bounds $\mathbf{b}$, but broader constraints on $\boldsymbol{\theta}$ can be similarly accommodated. $\boldsymbol{\theta}_0$ denotes the initial pa-

Algorithm 3: Quadratic constrained coordinate descent (QCCD)

Input: $h(\cdot), \boldsymbol{\theta}_0, \mathbf{b}$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$, $\alpha \leftarrow 1$
while Not Converged **do**
  $\mathcal{S} \leftarrow$ mini-batch subsampling, $\mathbf{g} \leftarrow \nabla h(\boldsymbol{\theta} \mid \mathcal{S})$, $\mathbf{H} \leftarrow E[\nabla^2 h(\boldsymbol{\theta} \mid \mathcal{S})]$
  $\boldsymbol{\theta}_{\text{CCD}} \leftarrow \text{CCD}(\boldsymbol{\theta}, \alpha\mathbf{g}, \mathbf{H}, \mathbf{b})$ — see Alg. 4
  $\beta \leftarrow \text{argmax}_{\beta \in (0,1]} \text{Armijo}(\beta) > c$, $\boldsymbol{\theta}_{\text{NEW}} \leftarrow \boldsymbol{\theta} + \beta(\boldsymbol{\theta}_{\text{CCD}} - \boldsymbol{\theta})$
  if stationarity is detected **then**
    $\alpha \leftarrow \alpha/2$
  end if
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_{\text{NEW}}$
end while
return $\boldsymbol{\theta}$

rameter values. Intuitively, QCCD iterates between building a quadratic approximation at the current $\boldsymbol{\theta}$,

$$\hat{h}(\boldsymbol{\theta}_{\text{NEW}}) = h(\boldsymbol{\theta}) + \mathbf{g}^\top(\boldsymbol{\theta}_{\text{NEW}} - \boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta}_{\text{NEW}} - \boldsymbol{\theta})^\top \mathbf{H}(\boldsymbol{\theta}_{\text{NEW}} - \boldsymbol{\theta}), \tag{7}$$

and finding the minimum of $\hat{h}(\boldsymbol{\theta}_{\text{NEW}})$ subject to the constraints on $\boldsymbol{\theta}$ using constrained coordinate descent (CCD), described in Algorithm 4.

Algorithm 4: Constrained coordinate descent (CCD)

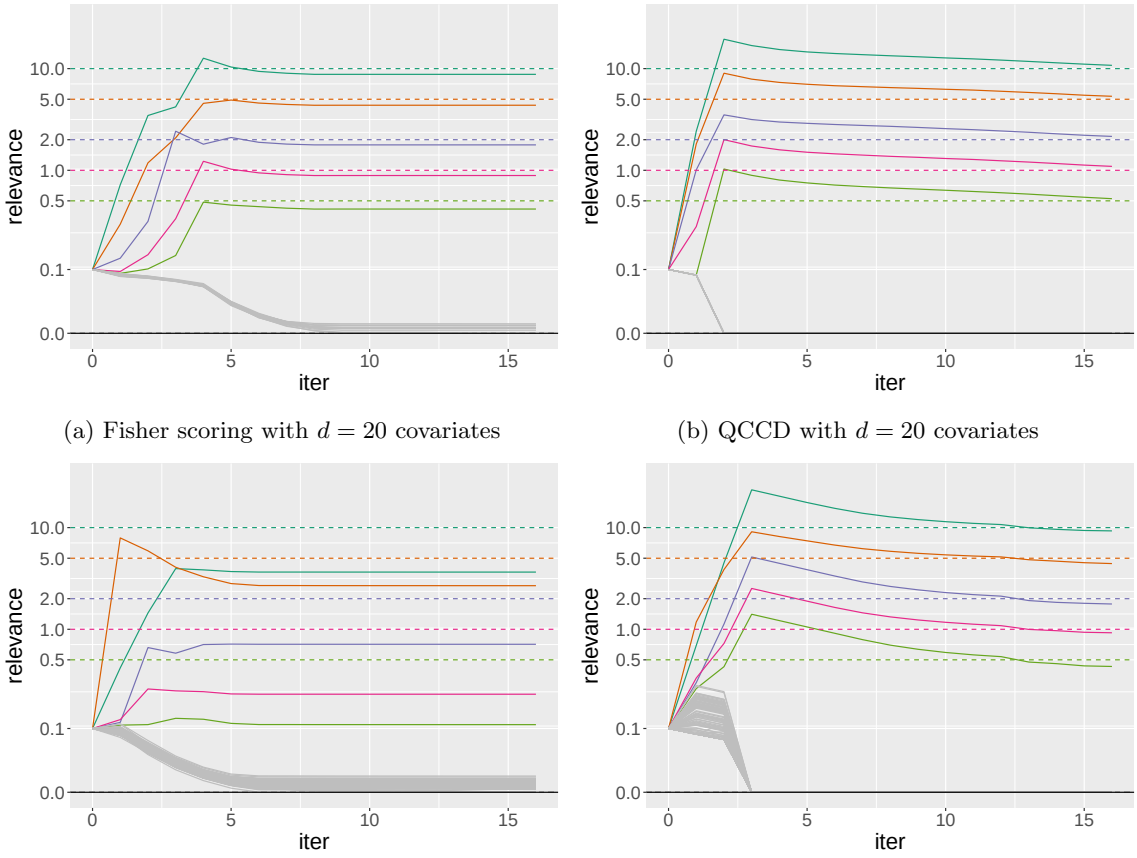
Input: $\boldsymbol{\theta}, \mathbf{g}, \mathbf{H}, \mathbf{b}$

$\mathbf{d} \leftarrow \mathbf{g} - \mathbf{H}\boldsymbol{\theta}$
while Not Converged **do**
  for i in $1 : \text{length}(\boldsymbol{\theta})$ **do**
    $\boldsymbol{\theta}[i] \leftarrow \max\left((-\mathbf{d}[i] - \mathbf{H}[i, -i] \cdot \boldsymbol{\theta}[-i])/\mathbf{H}[i, i], \mathbf{b}[i]\right)$
  end for
end while
return $\boldsymbol{\theta}$

The CCD algorithm cyclically considers each parameter of $\boldsymbol{\theta}$ in a constrained univariate quadratic optimization, where the minimum is analytically available and can be equal to the boundary value. The minimum returned by CCD is subsequently used in a line search subject to the Armijo condition that compares the ratio:

$$\frac{\beta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{CCD}}) \cdot \mathbf{g}}{h(\boldsymbol{\theta}) - h((1-\beta)\boldsymbol{\theta} + \beta\boldsymbol{\theta}_{\text{CCD}})},$$

with a threshold c to achieve 'sufficient decrease' of the objective function (i.e., to avoid unreasonably large steps) and $\boldsymbol{\theta}_{\text{NEW}}$ is guaranteed to exist subject to mild regularity conditions (Kressner, 2015). QCCD is similar to the cyclical coordinate descent algorithm (Friedman et al., 2010) in terms of building a quadratic approximation and using coordinate descent



(a) Fisher scoring with $d = 20$ covariates

(b) QCCD with $d = 20$ covariates

(c) Fisher scoring with $d = 100$ covariates

(d) QCCD with $d = 100$ covariates

Figure 4: Convergence of Fisher scoring and QCCD algorithms. y-axis is the relevance on the pseudo-log scale. The true covariates, with relevance $r_l > 0$, are color-coded with their true values marked by the colored dashed lines. The fake covariates, with relevance $r_l = 0$, are colored in grey.

but has two improvements, namely, the Armijo line search condition and the incorporation of parameter constraints.

QCCD has the same theoretical convergence rate as Fisher scoring because both find the minimum of the same quadratic approximation, but the former's ability to reach boundary values makes covariate deselection (i.e., r_l being optimized to zero) more straight-forward. Figure 4 compares the performance of Fisher scoring and QCCD when d is relatively small (i.e., $d \leq 100$). Covariates were independently generated at $n = 10^4$ locations and a bridge penalty with $\lambda = 32$ was used in the objective function, which will be further discussed in Section 3.6. The relevance vector $\tilde{\mathbf{r}}$ used for MM and NN was updated together with the updates of $\boldsymbol{\theta}$. The parameter estimates from QCCD were closer to the truth than those from Fisher scoring. Further, QCCD was able to deselect all fake covariates, achieving $r_l = 0$ for all $l > 5$, while Fisher scoring was unable to deselect any covariate without setting a truncation level. The ability to automatically deselect covariates becomes increasingly important when addressing GP regressions with larger numbers of covariates.

3.6 Bridge penalty and its extension

The desired properties of the penalty function for GP regression can be different from those for linear regression. Yi et al. (2011) compared several penalties in GP regression that include Lasso, SCAD, and bridge penalties, concluding that the bridge penalty has overall the best performance. This agrees with our analysis that unlike linear regression, GP regression automatically avoids improperly large magnitudes of $\mathbf{r}$. Therefore, a penalty function that becomes flat more quickly as the parameter magnitude increases is more suitable for GP regression, leading to higher model sparsity and smaller estimation bias.

However, one issue with the bridge penalty is that its derivative is infinite at zero, and so it is impossible to escape this local optimum for any parameter that reaches zero during optimization. This is especially problematic for the mini-batching procedure to be introduced later, where zero can be reached erroneously due to a "bad" batch. Hence, we adopt an iterative adaptive bridge penalty that amounts to a combination of the classic bridge penalty and the iterative adaptive technique in Ziel (2016) and Sun et al. (2010):

$$w_\lambda(\boldsymbol{\theta}) = \lambda \sum_{l=1}^{d} (c_{\iota,l}^{\kappa} + r_l^2)^\gamma, \tag{8}$$

where ι is the iteration number during optimization and $c_{\iota,l}^{\kappa}$ is the sum of the parameter r_l^2 over the previous κ iterations. In addition to allowing parameters to escape zero values, this adaptive bridge penalty also has the advantage that bigger r_l tends to have bigger $c_{\iota,l}^{\kappa}$, hence weaker penalty and smaller bias. Notice that $\kappa = 0$ corresponds to the classic bridge penalty used in Sections 3.1 and 3.5 and that when computing $\hat{\mathbf{H}}_{\boldsymbol{\theta}}^{\tilde{\mathbf{r}}}$, we ignore the second-order information of the penalty function to guarantee the non-negative definiteness of the FIM, which is equivalent to applying a linear approximation to $w_\lambda(x)$.

In this paper, we fix γ at 0.25 and select κ based on how likely the relevance parameters of the true covariates are to reach zero during optimization; see Section 3.9 for a more detailed analysis.

3.7 Mini-batching for Vecchia approximation

Although the Vecchia approximation has reduced the complexity of model estimation to be linear in n, we aim to further improve the computation efficiency of VGPR through mini-batch subsampling that has created considerable success in stochastic gradient descent. In this section, we propose a subsampling method specific to the Vecchia approximation that reduces the complexity to be linear in the batch size $\check{n}$ and leads to unbiased estimating equations. Specifically, we propose to sample the summands of the scaled Vecchia log-likelihood $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$:

$$\check{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}) = \sum_{i \in \mathcal{S} \subset \{1,\ldots,n\}} \log p_{\boldsymbol{\theta}}(y_i | \mathbf{y}_{c(i)}), \tag{9}$$

with equal probability and without replacement. Here, $\mathcal{S}$ is the mini-batch index set of size $\check{n}$ and we use $\check{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ and $\check{h}_\lambda^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ to denote the counterparts of $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ and $h_\lambda^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ under mini-batch subsampling.

This mini-batch subsampling can be applied to covariate selection and parameter estimation through slight modifications to Algorithms 2 and 3, respectively, as indicated by their blue underscored components. To avoid oscillation around the optimum, which is a common issue for mini-batch subsampling, we apply the technique introduced in Chee and Toulis (2018) to our QCCD algorithm for the detection of stationarity as indicated in Lines 6 to 8 of Algorithm 3. Specifically, the detection depends on the running sum of the inner product of successive stochastic gradients, and the learning rate α is halved upon detection of convergence; refer to Algorithm 1 of Chee and Toulis (2018) for more details.

One advantage of this mini-batch subsampling based on the Vecchia approximation is having unbiased gradient estimators:

$$E[\nabla\check{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})] = \nabla\left(E[\sum_{i=1}^{n}\log p_{\boldsymbol{\theta}}(y_i|\mathbf{y}_{c(i)})\delta_{i\in\mathcal{S}}]\right) = \nabla\left(\frac{\check{n}}{n}\sum_{i=1}^{n}\log p_{\boldsymbol{\theta}}(y_i|\mathbf{y}_{c(i)})\right) = \frac{\check{n}}{n}\nabla\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta}), \tag{10}$$

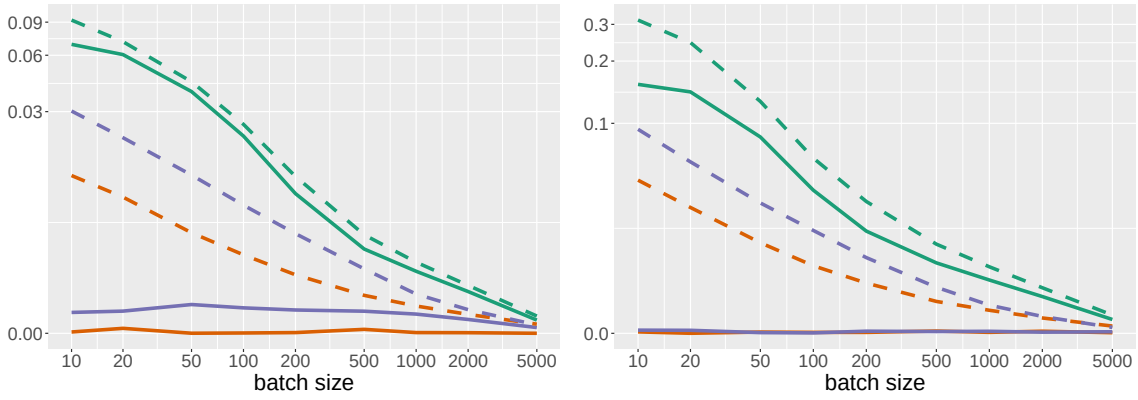
which is generally not the case for other mini-batch subsampling methods used in GP regression such as Chen et al. (2020). In (10), the expectation is taken with respect to $\mathcal{S}\subset\{1,\ldots,n\}$. The unbiased property of the mini-batch subsampling is relative to $\nabla\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ as opposed to $\nabla\ell(\boldsymbol{\theta})$; however, optimizing the Vecchia log-likelihood generally leads to the correct values for $\boldsymbol{\theta}$:

Proposition 5 *Assuming that* $\mathbf{y}$ *is a realization of a Gaussian process with zero mean and a covariance structure parameterized by* $\boldsymbol{\theta}=\boldsymbol{\theta}_0$ *and that* $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ *is its Vecchia-type log-likelihood, the true parameter value* $\boldsymbol{\theta}=\boldsymbol{\theta}_0$ *maximizes the expectation of* $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ *with respect to* $\mathbf{y}$*:* $\boldsymbol{\theta}_0\in argmax_{\boldsymbol{\theta}}E[\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})]$.

Corollary 6 $\nabla\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})=\mathbf{0}$ *are unbiased estimating equations assuming that* $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ *is first-order differentiable.*

The proof of Proposition 5 is in the Appendix, based on which the proof of Corollary 6 is straight-forward. Stein et al. (2004) showed that the Vecchia approximation of the restricted log-likelihood leads to unbiased estimating equations; here, we provide a stronger result for the Vecchia approximation of the log-likelihood.

We numerically compared our subsampling strategy to two other strategies in terms of the bias and the variance of their gradient estimators. Comparison method I selects $\check{n}$ responses from $\{(y_i)\}_{i=1}^{n}$ with equal probability and without replacement, and then the scaled Vecchia approximation for the GP defined over $\{(y_i)\}_{i\in\mathcal{S}}$ is used for computing the SR-gradient. Comparison method II is similar to what we proposed in (9), sampling the summands of $\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})$ but with probabilities proportional to $i^{-1/d}$ and without replacement, which compensates the $\mathcal{O}(i^{-1/d})$ decrease of $\min_{j\in c(i)}\|\mathbf{x}_i-\mathbf{x}_j\|$ (e.g., Katzfuss and Schäfer, 2021) into consideration and balances the presences of short-range and long-range distances. Figure 5 compares the three mini-batch subsampling methods using a GP defined over $n=10^4$ locations in $\mathbb{R}^2$ whose true parameters $\boldsymbol{\theta}$ are $(\sigma^2,r_1,r_2,\tau^2)=(1,1,0.5,0.05^2)$ and assumed known. The numbers of mini-batches averaged over are 5,000 if the batch size is smaller than 500 and 500 otherwise. Our proposed sampling method had the smallest empirical absolute bias and RMSE, highlighting its advantage as the SR-gradient estimator.



(a) Absolute bias and RMSE of $\frac{\partial \tilde{\ell}^{\mathrm{r}}(\boldsymbol{\theta})}{\partial r_1^2}$

(b) Absolute bias and RMSE of $\frac{\partial \tilde{\ell}^{\mathrm{r}}(\boldsymbol{\theta})}{\partial r_2^2}$

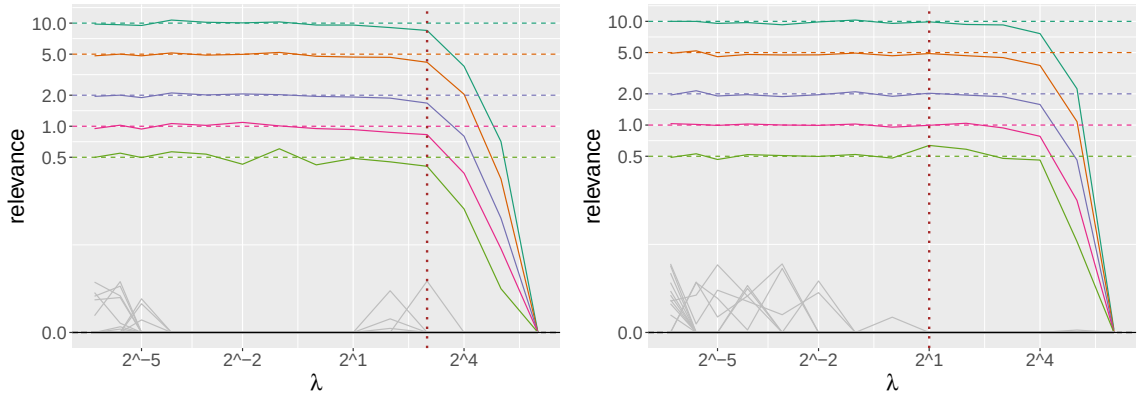
Figure 5: Absolute bias (solid) and RMSE (dashed) of the SR-gradient estimators of the three mini-batch subsampling methods. Red, green, and blue represent our proposed mini-batch subsampling of (9), comparison method I and comparison method II, respectively.

Comparison method I, as a most intuitive mini-batch subsampling method, leads to a poor gradient estimator because the responses not selected in $\mathcal{S}$ are ignored, losing significant amount of information compared with the other two methods. While it is desirable to reduce the dependence within each mini-batch, the smaller bias and variance of our proposed method over Comparison method II suggests finding sampling probabilities that lead to smaller variance is non-trivial and may lead to nonzero bias.

Figure 6 shows regularization paths in the same setting as in Figure 2, except using mini-batch subsampling with a batch size of 128 and increasing κ in the penalty function from zero to two. The choice of batch size poses a trade-off between computation efficiency and variability of the gradient estimator, which may depend on the training dataset and computation capacity; in general, larger batches improve the convergence stability but increase the computational cost. A discussion on the choice of κ is provided in Section 3.9. The estimated models indicated by the red dashed lines were almost the same as those in Figure 2 while the computation time was reduced by more than 90%. When considering the overall sparsity patterns in Figures 2 and 6, the combination of mini-batch subsampling and the iterative adaptive bridge penalty leads to a stronger capacity of deselecting fake covariates, because different mini-batches tend to select the same set of true covariates but different sets of fake covariates, inducing bigger variance on the gradient estimators of the fake covariates while $c_{\iota,l}^{\kappa}$ in (8) is smaller for the fake covariates, indicating stronger penalization.

3.8 Complexity analysis

In this section, we analyze the computation gains from using the Vecchia approximation, the VGPR algorithm introduced in Algorithm 1, and the mini-batching technique from Section 3.7. The Vecchia approximation reduces the complexity of computing the log-likelihood and its gradient from $\mathcal{O}(n^3)$ and $\mathcal{O}(n^3 d)$ to $\mathcal{O}(nm^3)$ and $\mathcal{O}(ndm^3)$, respectively; refer to Guinness (2021) for the gradient computation under the Vecchia approximation. Based



(a) Independent covariates

(b) Dependent covariates

Figure 6: Regularization path computed by VGPR with mini-batch subsampling, using independent or dependent covariates. The relevance parameters of the true covariates are color-coded, and their true values are marked by horizontal colored dashed lines. The fake covariates, whose true relevance parameters are zero, are colored in grey. The vertical red dotted lines mark the optimal model indicated by the stopping condition.

on the intermediate results from the gradient computation, the FIM of the Vecchia log-likelihood needs only $\mathcal{O}(nd^2m^2)$ additional operations. The VGPR algorithm reduces the number of covariates involved in optimization, reducing the d in aforementioned complexities to $|\zeta|$, with $|\zeta| \approx d_0 \ll d$. Finally, the mini-batching technique further reduces the $\mathcal{O}(n)$ complexities to $\mathcal{O}(\check{n})$, leading to $\mathcal{O}(\check{n}m^3)$, $\mathcal{O}(\check{n}d_0m^3)$, and $\mathcal{O}(\check{n}d_0^2m^2)$ complexities for computing the objective function, its gradient, and FIM, respectively. The SR-gradient of all d covariates is needed in Algorithm 2 to select k new covariates at the cost of $\mathcal{O}(\check{n}dm^3)$, but its computation frequency is negligible compared with the number of gradient computations needed by QCCD and it is typically a minor component in the overall computation cost.

For very large n, it is also possible to reduce the cost of MM and NN by replacing them by random ordering and the index-based-on-inverted-file (IVF) method (implemented in the Faiss library of Johnson et al., 2017), respectively. The cost of NN could be further reduced by computing the m nearest neighbors on-the-fly only for the responses in the mini-batch $\mathcal{S}$.

GP prediction also benefits significantly from the techniques introduced in Section 3. The nearest neighbors of each test point can be computed much faster in d_0 dimensions than in d dimensions, based on which posterior inference at each test point can be achieved in $\mathcal{O}(m^3 + d_0m^2)$ time using the scaled Vecchia approximation, assuming that the number of selected covariates is $\mathcal{O}(d_0)$.

3.9 Sensitivity to tuning parameters

The VGPR algorithm includes several tuning parameters that are considered fixed when running the algorithm. We provide some guidance here. Larger values of the conditioning set size m lead to more accurate approximation of the exact GP and we choose $m = 100$ based on Katzfuss et al. (2022) and for computational feasibility. The Armijo constant c in

Algorithm 3 heuristically prevents 'overly large' steps; we choose $c = 10^{-4}$ as recommended in Chapter 3 of Wright et al. (1999) and used in the `GpGp` R package Guinness (2021). The learning rate parameter α in Algorithm 4 reduces oscillation around an optimum, hence promoting convergence; we use the same initialization (i.e., $\alpha = 1$) and scaling (i.e., by 1/2) for α as in Chee and Toulis (2018), where this oscillation-reduction technique was proposed.

The number of new covariates selected each iteration (k) and the penalty parameters κ and γ in (8) are unique to our proposed VGPR algorithm and iterative adaptive bridge penalty, and hence they have not been discussed in the existing literature. Here, we provide some recommendations and a sensitivity analysis on them; see Appendix B for more details. Larger k leads to higher optimization efficiency but also the risk of local optima; we recommend a value between 3 and 5. Bigger κ corresponds to weaker numerical singularity at $r_l = 0$. We recommend $\kappa > 0$ when mini-batch subsampling is applied and a large κ (e.g., 10 or 15) when the GP with ARD kernels is likely a misspecified model. Smaller γ causes higher difference in the penalty derivatives at small and large r_l. Yi et al. (2011) used $\gamma = 0.01$, whereas we recommend a choice between 0.1 and 0.25 for a smoother objective function. Based on Appendix B, we conclude that the VGPR algorithm is overall not sensitive to the choice of k, κ, and γ.

4. Simulation Study

4.1 Simulation setup

We compared the VGPR algorithm proposed in Algorithm 1 with methods commonly used in machine learning for variable selection or GP model estimation, namely Lasso regression (Tibshirani, 1996), the sparse additive model (SAM; Ravikumar et al., 2009), regression trees (Tree; Loh, 2011), penalized GP regression (PGPR Yi et al., 2011), kernel interpolation for scalable structured Gaussian processes (KISS; Wilson and Nickisch, 2015), Vecchia Fisher scoring (Fisher; Guinness, 2021), and GPs with forward selection (FWD). We used the scaled Vecchia approximation in Fisher and FWD but the exact GP log-likelihood in PGPR to respect the original algorithm of Yi et al. (2011). For 'Tree' and 'Lasso', the default setups from the 'glmnet' R package (Friedman et al., 2010) and the 'sklearn' Python module (Pedregosa et al., 2011) were used, respectively. KISS generally has high scalability in n but low scalability in d. Based on the GPyTorch Gardner et al. (2018) implementation, when $d > 5$, the kernel function needed to assume an additive structure to be computationally feasible, for which ARD kernels are yet available, hence we only consider KISS as a state-of-the-art competitor for prediction at unknown locations. We generated d independent or dependent covariates at $(n + 5000)$ locations and simulated $(n + 5000)$ GP responses. 5,000 responses were set aside as the testing dataset used to evaluate the four methods' performances. We considered $n \in \{500, 5{,}000, 25{,}000\}$, $d \in \{100, 1{,}000\}$, and independent versus dependent covariates, for a total of 12 simulation scenarios. Methods were compared from three aspects, namely posterior prediction as measured by the RMSE based on the test dataset, misclassification ratios as measured by false positive and false negative ratios, and computation times.

'PGPR' and 'VGPR' use penalty functions, for which we chose the classic bridge penalty and the iterative adaptive bridge penalty as in Section 3.7, respectively, to compute their regularization paths. Methods involving solution paths, including 'VGPR', needed an OOS

score in their stopping conditions, for which a quarter or 5,000, whichever is smaller, of the training dataset was set aside and only used in computing the OOS RMSE. Similar to Sections 3.2 and 3.7, all stopping conditions were defined as producing less than 1% improvement of OOS RMSE after the selection of any new covariate. The OOS RMSE was also used to choose the best model in each iteration of forward selection. Fisher scoring does not require a stopping condition based on an OOS score and hence used the whole training dataset for parameter estimation.

Because Fisher scoring and the conjugate gradient used in Yi et al. (2011) are unconstrained optimization algorithms that rely on variable transformations, their parameters, including $\mathbf{r}$, cannot reach exact zeros. We set a cut-off threshold of 10^{-7}, the same as in Yi et al. (2011), below which the corresponding covariate was viewed as deselected. The initial values for σ^2, $\{r_l\}_{l=0}^d$, and τ^2, when needed, were 0.25, 0.1, and 10^{-4}, respectively, while for PGPR, ten random initial values, as recommended in Yi et al. (2011), were used for the optimization at each λ. The maximum numbers of iterations were 100 for PGPR, Fisher, and FWD, while 200 for VGPR, as the latter used mini-batch subsampling with $\check{n} = 128 \ll n$. The computation times were measured on an Intel Xeon E5-2680 v4 CPU using 56 cores and capped at a 10-hour limit for each GP replicate.

4.2 Simulation results

The comparison results are shown in Figure 7. The RMSEs of 'Fisher', 'FWD', 'PGPR', and 'VGPR', when computationally feasible, were similar for $d = 100$ but diverged for $d = 1{,}000$, indicating convergence to local optima when the number of optimization parameters was high. Specifically, both 'PGPR' and 'Fisher' involve $\mathcal{O}(d)$ parameters in optimization, while 'FWD' and 'VGPR' sequentially increase the number of parameters based on warm starts, which achieved significantly better result for reaching the global optimum. While 'FWD' provided slightly more accurate predictions than 'VGPR' for $n = 500$, it quickly became computationally infeasible as n or d increased. In contrast, 'VGPR' had a better trade-off between data efficiency and computation scalability. 'Lasso', 'SAM', and 'Tree' were less suitable for the simulated multivariate normal datasets due to model misspecification. While 'KISS' is a GP-based model, its idea of finding a (large) common set of pseudo-inputs for all locations became impractical when d is moderately large. In terms of 'FPos', which measures the proportion of fake covariates among the selected, 'VGPR' outperformed all other methods, achieving zero 'FPos' ratios when $n \geq 5{,}000$. This highlights the capability of 'VGPR' for deselecting fake covariates, hence the advantages of using QCCD over conjugate gradient and Fisher scoring for simultaneous variable selection and parameter estimation. The false negative ratios were almost constantly zero for all methods, and are hence not shown. Although slightly slower than the compared machine-learning models, 'VGPR' tremendously outperformed the other GP-based methods, becoming the only feasible GP-based method when $n = 25{,}000$ and $d = 1{,}000$ under the 10-hour limit.

5. Application Study

We performed a comparison on several real datasets and data produced by a physical model. Specifically, we compared the methods from Section 4 that are computationally feasible at $n = 25{,}000$ and $d = 1{,}000$, namely 'Lasso', 'Tree', 'VGPR', and 'SAM'. For these examples,

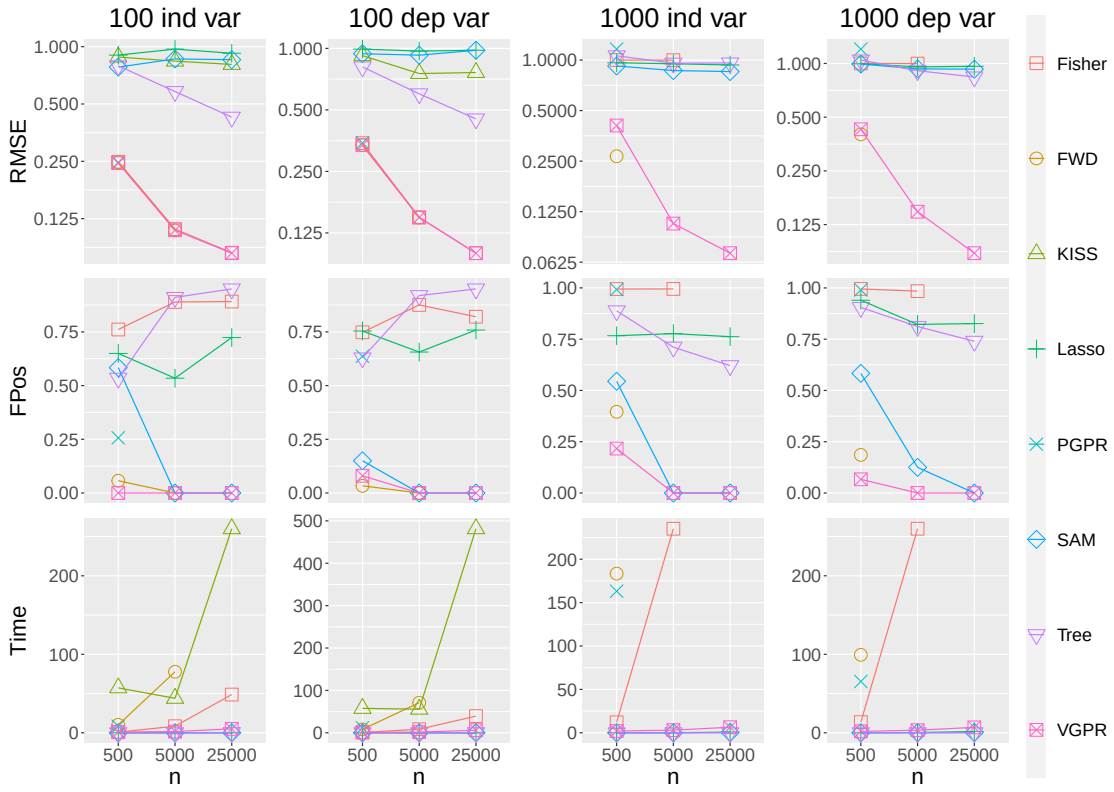


Figure 7: Comparison of eight methods for variable selection and/or GP regression, in terms of RMSE, false positive rates (FPos), and computation time in minutes. The results were averaged over five replicates. When $d = 100$, 'Fisher', 'FWD', 'PGPR', and 'VGPR' had close RMSE scores when available. The computation times of 'Lasso', 'SAM', and 'Tree' are similar, all faster than 'VGPR'.

our assumed model, GP with ARD covariance kernels, is likely to be misspecified and furthermore, true covariates may not exist in the given covariate pool, and so we used the iterative adaptive bridge penalty with κ bigger than those in Sections 3 and 4 to select the most predictive covariates.

The first dataset was generated from the Piston function (e.g., Surjanovic and Bingham, 2013), which is a (deterministic) physical model with $d_0 = 7$ true covariates; a total of $d = 10^3$ covariates were simulated at $n = 10^6$ locations. While the underlying model is not a GP, the true covariates were included in the covariate pool, and we chose $\kappa = 5$. The second dataset was the "relative location of CT slices on axial axis" (Slice) from the UCI Machine Learning Repository (Dua and Graff, 2017) that has $n = 53{,}500$ images and $d = 386$ features. The images belong to 74 individuals, among which a quarter were selected as the testing dataset. The third dataset was the "physicochemical properties of protein tertiary structure" (CASP) dataset also from the UCI Repository with $n = 45{,}730$ responses and $d = 9$ features. A fourth dataset was the temperature (Temp) data used in Garnett et al. (2013) that contains 7,117 training samples and 3,558 testing samples, each with $d = 106$ features. For the last three datasets, we set $\kappa = 15$ to compensate for model misspecification and the potential lack of true covariates, and supplemented with

$(10^3 - d)$ artificial covariates such that a total of 10^3 covariates were used to compare the three methods' capability of variable selection. Similar to previous experiments, we generated either uncorrelated or correlated covariates but here, the latter was constructed as random linear combinations of original covariates plus independent Gaussian noise. Both the responses $\mathbf{y}$ and the covariates $\mathbf{x}_i$ were standardized to have zero mean and unit variance.

Although the set of true covariates was unknown, misclassification ratios, specifically the false positive ratio, could still be estimated based on the number of included artificial covariates. On the other hand, the number of selected covariates is also an important indicator for the quality of variable selection that directly relates to over-fitting and computation efficiency. Table 1 summarizes the three metrics of the four methods under the previously mentioned datasets. The 'Piston' and the 'CASP' datasets had too few original covariates

Dataset	Method	RMSE	nSel	FPos	Dataset	Method	RMSE	nSel	FPos
Piston-I	VGPR	0.00	7	0%	CASP-I	VGPR	0.78	3	0%
	Lasso	0.17	7	0%		Lasso	0.85	177	96%
	SAM	NA	NA	NA		SAM	0.84	6	0%
	Tree	0.92	6	17%		Tree	0.92	6	17%
Piston-D	VGPR	0.00	7	0%	CASP-D	VGPR	0.75	10	60%
	Lasso	0.17	7	0%		Lasso	0.85	221	96%
	SAM	NA	NA	NA		SAM	0.84	6	0%
	Tree	0.71	147	95%		Tree	0.92	6	33%
Slice	VGPR	0.38	64		Temp	VGPR	0.29	6	
	Lasso	0.44	359			Lasso	0.28	84	
	SAM	0.50	118			SAM	0.29	40	
	Tree	0.44	334			Tree	0.32	26	
Slice-I	VGPR	0.32	49	18%	Temp-I	VGPR	0.29	8	0%
	Lasso	0.44	792	59%		Lasso	0.29	122	79%
	SAM	0.50	118	0%		SAM	0.29	36	8%
	Tree	0.51	294	31%		Tree	0.40	47	49%
Slice-D	VGPR	0.31	50	12%	Temp-D	VGPR	0.29	9	0%
	Lasso	0.43	696	54%		Lasso	0.29	92	70%
	SAM	0.49	142	20%		SAM	0.29	36	22%
	Tree	0.41	908	64%		Tree	0.39	46	46%

Table 1: Performance comparison of Lasso linear regression (Lasso), sparse additive model (SAM), regression tree (Tree), and VGPR. 'Slice', 'Piston', 'CASP', and 'Temp' are dataset names. 'I' and 'D' indicates being supplemented by uncorrelated and correlated artificial covariates, respectively. 'RMSE' measures the RMSE based on the testing dataset. 'nSelect' is the number of selected covariates. 'FPos' is short for false positive ratio.

to be used for comparing variable selection, and so corresponding results are not listed. The 'SAM' method exceeded our memory capacity (128 GB) when $n = 10^6$ using the 'Piston' dataset, and so the results are not available. The optimization setups for 'VGPR' were the same as in Section 4, except for the change of κ and that k was increased from 3 to 5 to further improve computation efficiency.

'VGPR' outperformed the other three methods in almost all three aspects (same as in Section 4), especially in terms of the number of selected covariates and the false positive ratios, highlighting the strength of using the iterative adaptive bridge penalty and QCCD for covariate deselection. For the 'Piston' dataset, our GP properly captured its non-linear and continuous features, hence predicting with significantly higher accuracy. 'VGPR' had a relatively high false positive ratio when the 'CASP-D' dataset was used but considering that there were only nine original covariates, the 'FPos' was already high with few fake covariates selected. Besides, the fake covariates in this case were correlated with the original covariates, potentially improving posterior inference as reflected by the lower RMSE of 'VGPR'. 'Lasso' and 'SAM' had comparable RMSE to 'VGPR' in modeling the 'Temp' dataset but its number of selected covariates and 'FPos' were significantly higher. The complexity of 'VGPR' is tremendously reduced by the Vecchia approximation, gradient-based covariate selection, and mini-batch subsampling, to achieve a computation time of less than forty minutes for a dataset with $n = 10^6$ and $d = 10^3$, for which 'Lasso' and 'Tree' used sixteen and eight minutes, respectively. Despite being slower, 'VGPR' is arguably as scalable as the other two methods (and much more so than existing GP regression methods) based on the complexity analysis in Section 3.8.

6. Conclusions

We provide a highly scalable method, coined VGPR, for variable selection and model estimation in GP regression, suitable for datasets with large numbers of responses n and covariates d. ARD covariance kernels naturally combine variable selection and model estimation, while a (scaled) Vecchia approximation provides fast and highly parallel computation of the loglikelihood, its gradient and its FIM. We introduced a forward-backward-selection algorithm that iteratively adds predictive covariates to a candidate set ζ based on the gradient and removes irrelevant covariates from the candidate set using an efficient QCCD algorithm. We provided theoretical support for the gradient-based covariate-candidate selection. To further speed up our method for even larger n, we introduced a mini-batch subsampling method specific to Vecchia-type approximations that has unbiased gradient estimators whose expectations are shown to be zero at the true parameter values. The resulting procedure requires only $\mathcal{O}(\check{n}|\zeta|^2 + \check{n}d)$ time, where $\check{n}$ is the mini-batch size, and hence the computational complexity is essentially independent of n. To compensate for the sampling variance of the stochastic gradient estimators under mini-batch subsampling, we also introduced an iterative adaptive bridge penalty.

In our simulation study, VGPR was substantially faster and selected fewer (almost zero) false covariates than other state-of-the-art GP regression methods that can be adapted for variable selection. When using real datasets, VGPR was robust enough to select only a small number of the most predictive covariates, maintaining the lowest misclassification ratios and the best predictive power among standard methods for regression with variable selection. VGPR is able to handle $n = 10^6$ responses with $d = 10^3$ features within 40 minutes on a standard scientific workstation. Due to its flexibility and accurate results, we consider VGPR to be a suitable candidate for a default benchmark method for nonlinear regression and variable selection on large datasets.

One possible extension of the results in this paper is variable selection and model estimation for generalized GP models, such as logistic or probit GPs for classification problems. For example, Cao et al. (2022) derived the marginal and posterior predictive probabilities of the probit GP. A second idea is to examine if the gradient of the objective function or similarly simple criteria can be used to select new covariates for other regression models, hence achieving a forward selection procedure that tremendously benefits the optimization.

Acknowledgments

Jian Cao was partially supported by the Texas A&M Institute of Data Science (TAMIDS) Postdoctoral Project program, Jian Cao and Matthias Katzfuss by National Science Foundation (NSF) Grant DMS–1654083, Matthias Katzfuss and Joe Guinness by NSF Grant DMS–1953005, Matthias Katzfuss by NSF Grant CCF–1934904, and Jian Cao and Marc Genton were partially supported by the King Abdullah University of Science and Technology (KAUST). We would like to thank Felix Jimenez for helpful comments and discussions.

A. Implementation of FIC, FITC, and PIC

FIC selects the first m locations in MM as the inducing inputs. FITC selects the same locations as the initial values of the m inducing inputs, which is then optimized using the 'GPflow' Python package, whose result is used as the final inducing inputs of FITC. PIC selects the first $m/2$ locations in MM as inducing inputs and divides the responses into disjoint subsets of size $m/2$. PIC considers the subsets of responses conditionally independent given the inducing inputs as opposed to that responses are conditionally independent, which is assumed by FIC and FITC. In other words, PIC considers also local correlation.

Among the five GP approximations, namely, FIC, FITC, PIC, Vecchia and scaled Vecchia, FIC typically has the lowest cost per likelihood estimation, requiring only $\mathcal{O}(nm^2)$ operations because the conditioning sets remain the same for all responses. PIC has higher computation cost than FIC but its complexity stays at the same level. Given the inducing inputs, FITC is as efficient as FIC but the inducing inputs of FITC require an optimization with $\mathcal{O}(md)$ parameters, which could become the dominant complexity. Vecchia and scaled Vecchia approximations have a complexity of $\mathcal{O}(nm^3)$ for likelihood estimation, which although higher than FIC and PIC, is still linear with n and has a highly parallel implementation. Furthermore, the grouping technique introduced in Guinness (2018) can reduce the previous complexity to between $\mathcal{O}(nm^2)$ and $\mathcal{O}(nm^3)$ and is already implemented in the 'GpGp' R package.

B. Sensitivity Analysis

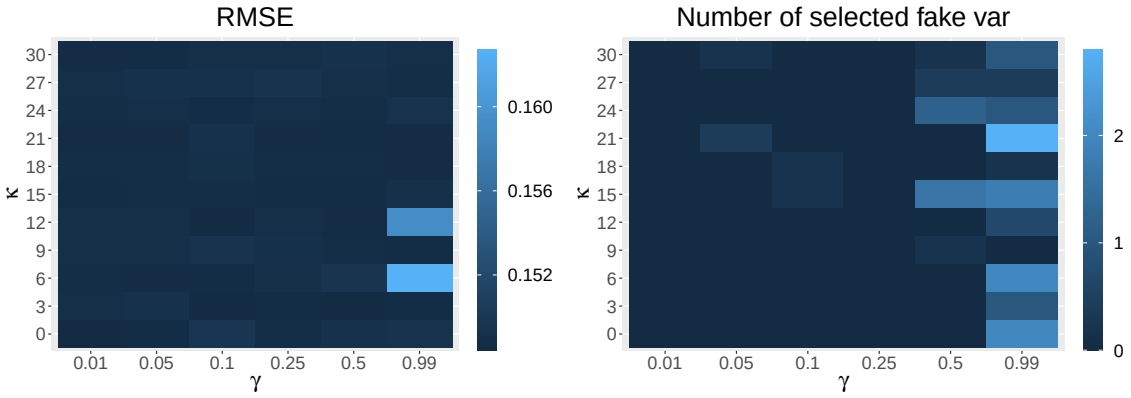
We generated $d = 100$ dependent covariates at 10^4 locations, half of which were used for training and the other half were used to compute the RMSE score. The sensitivity was assessed in terms of RMSE score and number of fake covariates selected. From Figure 8, we conclude that our proposed VGPR algorithm is largely robust across different values of k, γ, and κ within the recommended intervals.

C. Proofs

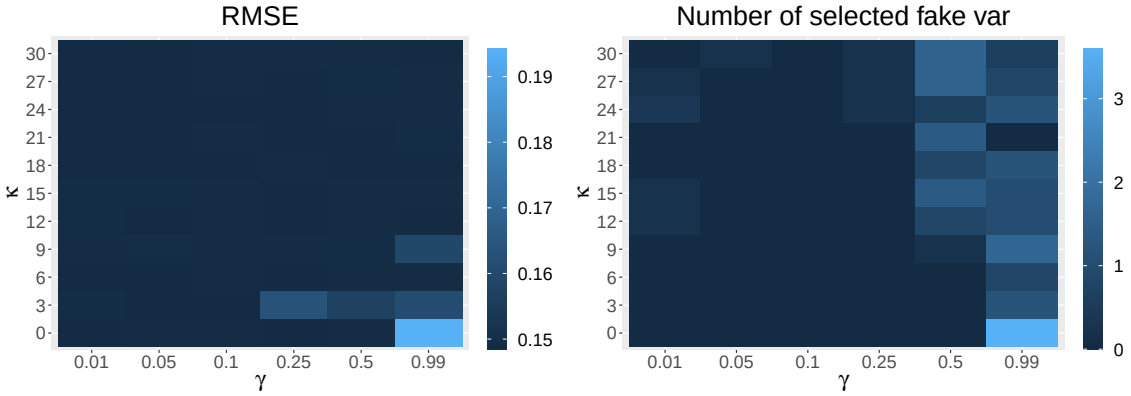
Proof [Partial proof of Proposition 1] Ignoring the constant term in ℓ:

$$\begin{aligned} \ell &= -\frac{1}{2}\mathbf{y}^\top \boldsymbol{\Sigma}^{-1}\mathbf{y} - \frac{1}{2}\log|\boldsymbol{\Sigma}| \\ \left.\frac{\partial \ell}{\partial r_l^2}\right|_{(\sigma,\mathbf{r},\tau)=(\sigma_1,\mathbf{r}_1,\tau_1)} &= \mathbf{y}^\top \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{l,1}\boldsymbol{\Sigma}_1^{-1}\mathbf{y} - \operatorname{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{l,1}), \end{aligned}$$

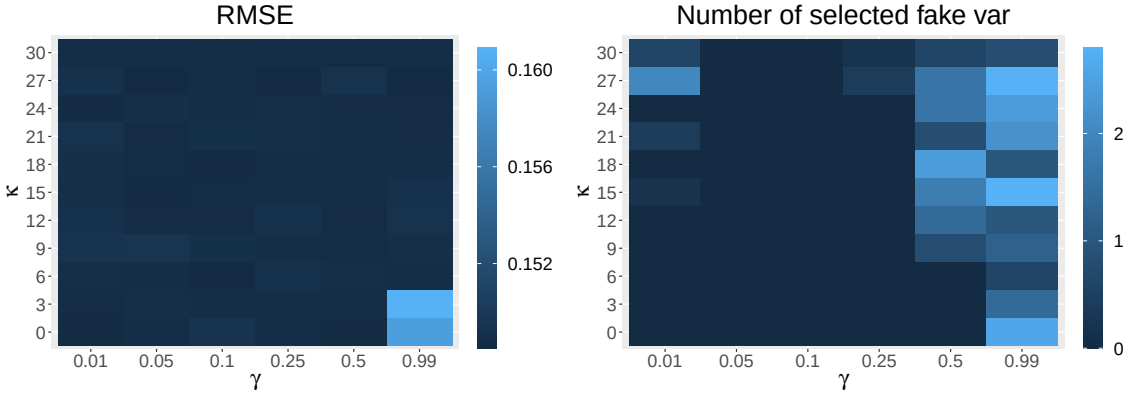
where $\boldsymbol{\Sigma}_{l,1} = \left.\frac{\partial \boldsymbol{\Sigma}}{\partial r_l^2}\right|_{(\sigma,\mathbf{r},\tau)=(\sigma_1,\mathbf{r}_1,\tau_1)} = -\sigma_1^2\tilde{\boldsymbol{\Sigma}}_1 \odot \mathbf{D}_l$, $\mathbf{D}_l$ is an $n \times n$ matrix whose (i,j)-th coefficient is $(x_{il} - x_{jl})^2$, and $\odot$ is the Hadamard product. Because $\{x_{il}\}_{i=1,\ldots,n,l=1,\ldots,d}$ have i.i.d. distributions and



(a) $k = 3$



(b) $k = 4$



(c) $k = 5$

Figure 8: Sensitivity of the VGPR algorithm with respect to the number of new covariates selected at each iteration (k), and the penalty parameters (γ, κ) defined in (8).

$r_{l_1 1} = r_{l_2 1}$, we have:

$$E[\mathrm{tr}(\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l_1,1})] = E[\mathrm{tr}(\mathbf{\Sigma}_1^{-1}(-\sigma_1^2\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_{l_1}))] = E[\mathrm{tr}(\mathbf{\Sigma}_1^{-1}(-\sigma_1^2\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_{l_2}))] = E[\mathrm{tr}(\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l_2,1})].$$

Therefore, we only need to compare $E[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l_1,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}]$ and $E[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l_2,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}]$. First consider $\{\mathbf{x}_i\}_{i=1}^n$ as fixed and take expectation with respect to $\mathbf{y}$:

$$E_{\mathbf{y}}[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}] = \mathrm{tr}(\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l,1}\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_0).$$

$\mathbf{\Sigma}_0$ can be also written as:

$$\mathbf{\Sigma}_0 = \sigma_0^2\tilde{\mathbf{\Sigma}}_1 \odot \exp\left(-\sum_{l=d_1+1}^{d}(r_{l0}^2 - r_{l1}^2)\mathbf{D}_l\right) + \tau_0^2\mathbf{I}_n = \sigma_0^2\tilde{\mathbf{\Sigma}}_1 \odot \exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C}) + \tau_0^2\mathbf{I}_n,$$

where $\mathbf{C} = -\sum_{l=\{d_1+1,\ldots,d_0\}\setminus l_1} r_{l0}^2\mathbf{D}_l$. Hence, we can re-write $E_{\mathbf{y}}[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}]$:

$$\begin{aligned}
E_{\mathbf{y}}[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}] &= \mathrm{tr}\left(\mathbf{\Sigma}_1^{-1}(-\sigma_1^2\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_l)\mathbf{\Sigma}_1^{-1}(\sigma_0^2\tilde{\mathbf{\Sigma}}_1 \odot \exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C}) + \tau_0^2\mathbf{I}_n)\right) \\
&= \mathrm{tr}\left(\mathbf{\Sigma}_1^{-1}(-\sigma_1^2\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_l)\mathbf{\Sigma}_1^{-1}(\sigma_0^2\tilde{\mathbf{\Sigma}}_1 \odot \exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C}))\right) + c_\tau,
\end{aligned}$$

where $c_\tau = \tau_0^2\mathrm{tr}(\mathbf{\Sigma}_1^{-1}(-\sigma_1^2\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_l)\mathbf{\Sigma}_1^{-1})$ and $E_{\mathbf{X}}[c_\tau]$ remains the same for $l = l_1$ and $l = l_2$ because $\{x_{il}\}_{i=1,\ldots,n,l=1,\ldots,d}$ have i.i.d. distributions. We can also remove the σ_0^2 and σ_1^2 from the equation above with $c_\sigma = \sigma_0^2\sigma_1^2$:

$$E_{\mathbf{y}}[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}] = c_\sigma\mathrm{tr}\left(\mathbf{\Sigma}_1^{-1}(-\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_l)\mathbf{\Sigma}_1^{-1}(\tilde{\mathbf{\Sigma}}_1 \odot \exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C}))\right) + c_\tau. \tag{11}$$

Since $\mathbf{\Sigma}_1^{-1}(-\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_l)\mathbf{\Sigma}_1^{-1}$ and $\tilde{\mathbf{\Sigma}}_1 \odot \exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C})$ are symmetric, we have:

$$\begin{aligned}
E_{\mathbf{y}}[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}] &= c_\sigma\langle\mathrm{vec}\left(\mathbf{\Sigma}_1^{-1}(-\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_l)\mathbf{\Sigma}_1^{-1}\right), \mathrm{vec}\left(\tilde{\mathbf{\Sigma}}_1 \odot \exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C})\right)\rangle + c_\tau \\
&= c_\sigma\langle(\mathbf{\Sigma}_1^{-1} \otimes \mathbf{\Sigma}_1^{-1})\mathrm{vec}(-\tilde{\mathbf{\Sigma}}_1 \odot \mathbf{D}_l), \mathrm{vec}\left(\tilde{\mathbf{\Sigma}}_1 \odot \exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C})\right)\rangle + c_\tau \\
&= c_\sigma\langle(\mathbf{\Sigma}_1^{-1} \otimes \mathbf{\Sigma}_1^{-1})\mathrm{diag}(\mathrm{vec}(\tilde{\mathbf{\Sigma}}_1))\mathrm{vec}(-\mathbf{D}_l), \mathrm{diag}(\mathrm{vec}(\tilde{\mathbf{\Sigma}}_1))\mathrm{vec}(\exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C}))\rangle + c_\tau,
\end{aligned}$$

where $\langle,\rangle$ is the Euclidean inner product, $\otimes$ is the Kronecker product, and $\mathrm{vec}(\cdot)$ is the vectorization of a matrix that stacks the columns of a matrix on top of one another. Use $\mathbf{M}$ to denote $\mathrm{diag}(\mathrm{vec}(\tilde{\mathbf{\Sigma}}_1))(\mathbf{\Sigma}_1^{-1} \otimes \mathbf{\Sigma}_1^{-1})\mathrm{diag}(\mathrm{vec}(\tilde{\mathbf{\Sigma}}_1))$, which is a positive definite matrix:

$$E_{\mathbf{y}}[\mathbf{y}^\top \mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{l,1}\mathbf{\Sigma}_1^{-1}\mathbf{y}] = c_\sigma\langle\mathbf{M}\mathrm{vec}(-\mathbf{D}_l), \mathrm{vec}(\exp(-r_{l_10}^2\mathbf{D}_{l_1} + \mathbf{C}))\rangle + c_\tau.$$

Now consider the expectation with respect to $\mathbf{X}$ and notice that $\mathbf{M}$, $\mathbf{D}_{l_1}$, and $\mathbf{C}$ are mutually independent. Assuming $l = l_1$ or $l = l_2$:

$$E_{\mathbf{X}}[\langle\mathbf{M}\mathrm{vec}(-\mathbf{D}_l), \mathrm{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))\rangle] = \mathrm{tr}\left(E_{\mathbf{X}}[\mathrm{vec}(-\mathbf{D}_l)\mathrm{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))^\top]E_{\mathbf{X}}[\mathbf{M}]\right) \tag{12}$$

To show (12) is bigger when $l = l_1$ than when $l = l_2$, it remains to show that:

$$\mathrm{tr}\left(\mathbf{Cov}_{\mathbf{X}}\left[\mathrm{vec}(-\mathbf{D}_{l_1}), \mathrm{vec}\left(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C})\right)\right]E_{\mathbf{X}}[\mathbf{M}]\right) > 0.$$

It suffices to show that $\mathbf{Cov}_{\mathbf{X}}[\mathrm{vec}(-\mathbf{D}_{l_1}), \mathrm{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))]$ is positive semi-definite and has a rank greater than zero because the trace of the multiplication between one positive definite matrix and one non-zero positive semi-definite matrix is positive. It is obvious that $\mathbf{Cov}_{\mathbf{X}}[\mathrm{vec}(-\mathbf{D}_{l_1}), \mathrm{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))]$ has a rank greater than zero. Its coefficients have three types of values:

$$\begin{aligned}
\mathbf{Cov}_{\mathbf{X}}[-d_{i_1,j_1}^{l_1}, \exp(-r_{l_1}^2 d_{i_2,j_2}^{l_1} + c_{i_2,j_2})] &= \mathbf{Cov}_{\mathbf{X}}[-d_{i_1,j_1}^{l_1}, \exp(-r_{l_1}^2 d_{i_2,j_2}^{l_1})]E_{\mathbf{X}}[\exp(c_{i_2,j_2})] \\
&= \begin{cases} 0 & i_1 = j_1 \text{ or } i_2 = j_2 \text{ or } |\{i_1,i_2,j_1,j_2\}| = 4 \\ a & i_1 \neq j_1 \text{ and } \{i_1,j_1\} = \{i_2,j_2\} \\ b & i_1 \neq j_1 \text{ and } i_2 \neq j_2 \text{ and } |\{i_1,i_2,j_1,j_2\}| = 3 \end{cases},
\end{aligned}$$

where $d_{i,j}^{l_1}$ and $c_{i,j}$ denote the (i,j)-th coefficients of $\mathbf{D}_{l_1}$ and $\mathbf{C}$, respectively, and $|\cdot|$ denotes the cardinality of a set. First, we can take out $E_{\mathbf{X}}[\exp(c_{i_2,j_2})]$ and have the following:

$$\mathbf{Cov}_{\mathbf{X}}[\mathrm{vec}(-\mathbf{D}_{l_1}), \mathrm{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))] = \mathbf{Cov}_{\mathbf{X}}[\mathrm{vec}(-\mathbf{D}_{l_1}), \mathrm{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1}))] \times c,$$

where $c = E_{\mathbf{X}}[\exp(c_{i_2,j_2})]$ for $i_2 \neq j_2$. Second, notice that the structures of

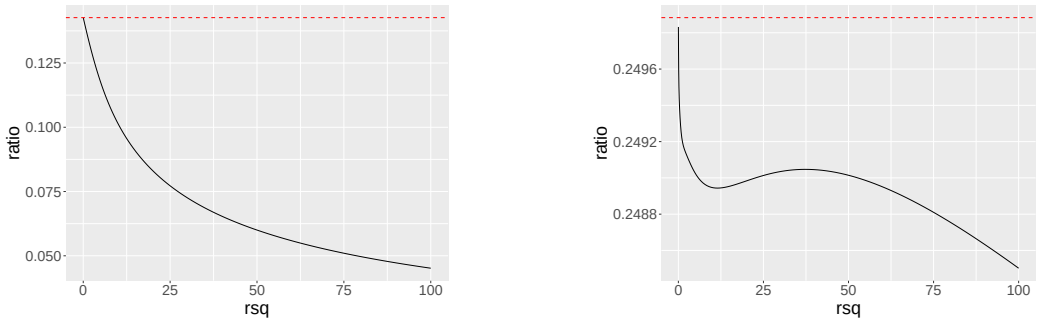
$$\mathbf{Cov}_{\mathbf{X}}[\text{vec}(-\mathbf{D}_{l_1}), \text{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1}))] \text{ and } \mathbf{Cov}_{\mathbf{X}}[\text{vec}(\mathbf{D}_{l_1}), \text{vec}(\mathbf{D}_{l_1})]$$

are the same, except for that the latter has different values for a and b, denoted by $\tilde{a}$ and $\tilde{b}$, respectively. Since $\mathbf{Cov}_{\mathbf{X}}[\text{vec}(\mathbf{D}_{l_1}), \text{vec}(\mathbf{D}_{l_1})]$ is positive semi-definite and a and $\tilde{a}$ are positive, to show that $\mathbf{Cov}_{\mathbf{X}}[\text{vec}(-\mathbf{D}_{l_1}), \text{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1}))]$ is also positive semi-definite, it is sufficient to show that $0 \leq \frac{b}{a} \leq \frac{\tilde{b}}{\tilde{a}}$.

$$\frac{\tilde{b}}{\tilde{a}} = \frac{\text{cov}[(X_1 - X_2)^2, (X_1 - X_3)^2]}{\text{var}[(X_1 - X_2)^2]}, \tag{13}$$

$$\frac{b}{a} = \frac{\text{cov}[-(X_1 - X_2)^2, \exp(-r^2(X_1 - X_3)^2)]}{\text{cov}[-(X_1 - X_2)^2, \exp(-r^2(X_1 - X_2)^2)]}. \tag{14}$$

Without loss of generality, we can assume $X_1, X_2, X_3 \overset{i.i.d.}{\sim} U(-\frac{1}{2}, \frac{1}{2})$ or $X_1, X_2, X_3 \overset{i.i.d.}{\sim} N(0, 1)$, under which $\frac{b}{a} \leq \frac{\tilde{b}}{\tilde{a}}$ can be shown numerically as in Figure 9.



(a) (b)

Figure 9: The ratio (14) (solid black) and the threshold (13) (dashed red) under (a) uniform distribution and (b) normal distribution for X_i.

With the accurately computed relationship between (13) and (14), we conclude the proof under the assumption that $\{x_{il}\}_{i=1,\ldots,n,l=1,\ldots,d}$ have i.i.d. uniform or normal distributions. For other distributions, similar numerical procedure can be used to draw the conclusion. ■

Proof [Proof of Corollary 2] When $\mathbf{r} \to \mathbf{0}$, $\mathbf{\Sigma} \to \mathbf{1}_{n\times n}$. We can substitute $\tilde{\mathbf{\Sigma}}_1$ in the proof of Proposition 1 by a non-singular covariance matrix arbitrarily close to $\mathbf{1}_{n\times n}$, denoted by $\tilde{\mathbf{\Sigma}}_1$, substitute d_1 by 0, and hence

$$\mathbf{\Sigma}_0 \approx \sigma_0^2\tilde{\mathbf{\Sigma}}_1 \odot \exp\left(-\sum_{l=d_1+1}^{d}(r_{l0}^2 - r_{l1}^2)\mathbf{D}_l\right) + \tau_0^2\mathbf{I}_n.$$

The rest of the proof should remain the same. ■

Proof [Proof of Corollary 3] Based on (12),

$$\begin{aligned}
&E\left[\left.\frac{\partial\ell}{\partial r_{l_1}^2}\right|_{(\sigma,\mathbf{r},\tau)=(\sigma_1,\mathbf{r}_1,\tau_1)}\right] - E\left[\left.\frac{\partial\ell}{\partial r_{d+1}^2}\right|_{(\sigma,\mathbf{r},\tau)=(\sigma_1,\mathbf{r}_1,\tau_1)}\right] \\
&= c_\sigma E_{\mathbf{X}}[\langle \mathbf{M}\text{vec}(-\mathbf{D}_{l_1}), \text{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))\rangle] - c_\sigma E_{\mathbf{X}}[\langle \mathbf{M}\text{vec}(-\mathbf{D}_{d+1}), \text{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))\rangle] \\
&\quad \text{Because } \{x_{il}\}_{i=1,\ldots,n,l=1,\ldots,d} \text{ have i.i.d. normal distributions, } E_{\mathbf{X}}[\text{vec}(-\mathbf{D}_{d+1})] = E_{\mathbf{X}}[\text{vec}(-\mathbf{D}_{l_1})], \\
&= c_\sigma\rho_2^2\text{tr}\left(\mathbf{Cov}_{\mathbf{X}}[\text{vec}(-\mathbf{D}_{l_1}), \text{vec}(\exp(-r_{l_1}^2\mathbf{D}_{l_1} + \mathbf{C}))]E_{\mathbf{X}}[\mathbf{M}]\right) > 0.
\end{aligned}$$

■

Proof [Proof of Proposition 4] From the proof of Proposition 1, we know that

$$\begin{aligned}&E\left[\left.\frac{\partial \ell}{\partial r_{l_1}^2}\right|_{(\sigma,\mathbf{r},\tau)=(\sigma_1,\mathbf{r}_1,\tau_1)}\right]-E\left[\left.\frac{\partial \ell}{\partial r_{l_2}^2}\right|_{(\sigma,\mathbf{r},\tau)=(\sigma_1,\mathbf{r}_1,\tau_1)}\right]\\&=E_{\mathbf{X}}[E_{\mathbf{y}}[\mathbf{y}^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{l_1,1}\boldsymbol{\Sigma}_1^{-1}\mathbf{y}]]-E_{\mathbf{X}}[E_{\mathbf{y}}[\mathbf{y}^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{l_2,1}\boldsymbol{\Sigma}_1^{-1}\mathbf{y}]].\end{aligned}$$

Based on (11),

$$\begin{aligned}&E_{\mathbf{y}}[\mathbf{y}^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{l_1,1}\boldsymbol{\Sigma}_1^{-1}\mathbf{y}]-E_{\mathbf{y}}[\mathbf{y}^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{l_2,1}\boldsymbol{\Sigma}_1^{-1}\mathbf{y}]\\&=c_\sigma\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}(-\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1})\boldsymbol{\Sigma}_1^{-1}\tilde{\boldsymbol{\Sigma}}_0\right)-c_\sigma\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}(-\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_2})\boldsymbol{\Sigma}_1^{-1}\tilde{\boldsymbol{\Sigma}}_0\right)\\&=c_\sigma\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}(-\tilde{\boldsymbol{\Sigma}}_1\odot(\mathbf{D}_{l_1}-\mathbf{D}_{l_2}))\boldsymbol{\Sigma}_1^{-1}\tilde{\boldsymbol{\Sigma}}_0\right)\\&\approx c_\sigma\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}(-\tilde{\boldsymbol{\Sigma}}_1\odot(\mathbf{D}_{l_1}-\mathbf{D}_{l_2}))\boldsymbol{\Sigma}_1^{-1}\left(\tilde{\boldsymbol{\Sigma}}_1+\sum_{\tilde{l}=d_1+1}^{d_0}\tilde{\boldsymbol{\Sigma}}_{\tilde{l},1}r_{\tilde{l}0}^2\right)\right)\\&=c_\sigma\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}(-\tilde{\boldsymbol{\Sigma}}_1\odot(\mathbf{D}_{l_1}-\mathbf{D}_{l_2}))\boldsymbol{\Sigma}_1^{-1}\left(\tilde{\boldsymbol{\Sigma}}_1-\sum_{\tilde{l}=d_1+1}^{d_0}r_{\tilde{l}0}^2\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{\tilde{l}}\right)\right),\end{aligned}$$

where $\tilde{\boldsymbol{\Sigma}}_{l,1}=\left.\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial r_l^2}\right|_{(\sigma,\mathbf{r},\tau)=(1,\mathbf{r}_1,0)}=-\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_l$. Noticing that $E_{\mathbf{X}}[\mathbf{D}_{l_1}-\mathbf{D}_{l_2}]$ is a zero matrix, that $\{\mathbf{D}_l\}_{l=1}^d$ are mutually independent, and that $\mathbf{D}_{l_1}$ and $\mathbf{D}_{l_2}$ are independent from $\tilde{\boldsymbol{\Sigma}}_1$, the expectation of the above equation with respect to $\mathbf{X}$ is equal to:

$$\begin{aligned}&c_\sigma E_{\mathbf{X}}\left[\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}(-\tilde{\boldsymbol{\Sigma}}_1\odot(\mathbf{D}_{l_1}-\mathbf{D}_{l_2}))\boldsymbol{\Sigma}_1^{-1}\left(-r_{l_10}^2\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1}\right)\right)\right]\\&=c_\sigma r_{l_10}^2E_{\mathbf{X}}\left[\mathrm{tr}\left((\boldsymbol{\Sigma}_1^{-1}\otimes\boldsymbol{\Sigma}_1^{-1})\mathrm{vec}(\tilde{\boldsymbol{\Sigma}}_1\odot(\mathbf{D}_{l_1}-\mathbf{D}_{l_2}))\mathrm{vec}\left(\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1}\right)^\top\right)\right]\\&=c_\sigma r_{l_10}^2\mathrm{tr}\left(E_{\mathbf{X}}[\boldsymbol{\Sigma}_1^{-1}\otimes\boldsymbol{\Sigma}_1^{-1}]\mathbf{Cov}_{\mathbf{X}}\left[\mathrm{vec}(\tilde{\boldsymbol{\Sigma}}_1\odot(\mathbf{D}_{l_1}-\mathbf{D}_{l_2})),\mathrm{vec}\left(\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1}\right)\right]\right)\\&=c_\sigma r_{l_10}^2\mathrm{tr}\left(E_{\mathbf{X}}[\boldsymbol{\Sigma}_1^{-1}\otimes\boldsymbol{\Sigma}_1^{-1}]\mathbf{Cov}_{\mathbf{X}}\left[\mathrm{vec}(\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1}),\mathrm{vec}(\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1})\right]\right)>0,\end{aligned}$$

because $E_{\mathbf{X}}[\boldsymbol{\Sigma}_1^{-1}\otimes\boldsymbol{\Sigma}_1^{-1}]$ is positive-definite and $\mathbf{Cov}_{\mathbf{X}}\left[\mathrm{vec}(\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1}),\mathrm{vec}(\tilde{\boldsymbol{\Sigma}}_1\odot\mathbf{D}_{l_1})\right]$ is positive semi-definite with a rank greater than zero. ■

Proof [Proof of Proposition 5] Here, we take expectations only with respect to $\mathbf{y}$ and consider $\{\mathbf{x}_i\}_{i=1}^n$ as fixed. Using the non-negativeness of the KL divergence, we can show that for a generic random vector $\mathbf{w}$, whose distribution is parameterized by $\boldsymbol{\theta}_0$:

$$\begin{aligned}E[\log p(\mathbf{w};\boldsymbol{\theta}_0)]-E[\log p(\mathbf{w};\boldsymbol{\theta})]&=\int\log\frac{p(\mathbf{w};\boldsymbol{\theta}_0)}{p(\mathbf{w};\boldsymbol{\theta})}p(\mathbf{w};\boldsymbol{\theta}_0)\mathrm{d}\mathbf{w}\geq 0\\&\Rightarrow E[\log p(\mathbf{w};\boldsymbol{\theta}_0)]\geq E[\log p(\mathbf{w};\boldsymbol{\theta})],\end{aligned}$$

with which Proposition 5 can be thus proved:

$$\begin{aligned}E_{\mathbf{y}}[\hat{\ell}^{\tilde{\mathbf{r}}}(\boldsymbol{\theta})]=E_{\mathbf{y}}[\log\hat{p}_{\boldsymbol{\theta}}^{\tilde{\mathbf{r}}}(\mathbf{y})]&=E_{\mathbf{y}}[\log\prod_{i=1}^n p_{\boldsymbol{\theta}}(y_i|\mathbf{y}_{c(i)})]=E_{\mathbf{y}}[\sum_{i=1}^n\log p_{\boldsymbol{\theta}}(y_i|\mathbf{y}_{c(i)})]\\&=\sum_{i=1}^n E_{\mathbf{y}_{c(i)}}\left[E_{y_i|\mathbf{y}_{c(i)}}[\log p_{\boldsymbol{\theta}}(y_i|\mathbf{y}_{c(i)})]\right],\end{aligned}$$

where $E_{y_i|\mathbf{y}_{c(i)}}[\log p_{\boldsymbol{\theta}}(y_i|\mathbf{y}_{c(i)})]$ achieves maximum at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. Therefore, $E_{\mathbf{y}}[\hat{\ell}^{\tilde{\mathbf{r}}}_{\boldsymbol{\theta}}(\mathbf{y})]$ achieves maximum at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. ■

References

Sudipto Banerjee, Alan E. Gelfand, Andrew O. Finley, and Huiyan Sang. Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society, Series B*, 70(4):825–848, 2008. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2008.00663.x. URL `http://www.ncbi.nlm.nih.gov/pubmed/19750209`.

Roberto Calandra, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth. Manifold Gaussian processes for regression. *Proceedings of the International Joint Conference on Neural Networks*, 2016-October:3338–3345, 2016. doi: 10.1109/IJCNN.2016.7727626.

Jian Cao, Daniele Durante, and Marc G Genton. Scalable computation of predictive probabilities in probit models with gaussian process priors. *Journal of Computational and Graphical Statistics*, page to appear, 2022.

Jerry Chee and Panos Toulis. Convergence diagnostics for stochastic gradient descent with constant learning rate. In *International Conference on Artificial Intelligence and Statistics*, pages 1476–1485. PMLR, 2018.

Bo Chen, Rui Castro, and Andreas Krause. Joint optimization and variable selection of high-dimensional gaussian processes. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1423–1430. International Machine Learning Society, 2012.

Hao Chen, Lili Zheng, Raed Al Kontar, and Garvesh Raskutti. Stochastic gradient descent in correlated settings: A study on gaussian processes. *Advances in Neural Information Processing Systems*, 2020.

Abhirup Datta, Sudipto Banerjee, Andrew O. Finley, and Alan E. Gelfand. Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016. ISSN 0162-1459. doi: 10.1080/01621459.2015.1044091. URL `http://arxiv.org/abs/1406.7343`.

Jacob Dearmon and Tony E Smith. Gaussian process regression and bayesian model averaging: an alternative approach to modeling spatial phenomena. *Geographical Analysis*, 48(1):82–111, 2016.

Marc Deisenroth and Jun Wei Ng. Distributed gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490. PMLR, 2015.

Ian Delbridge, David Bindel, and Andrew Gordon Wilson. Randomly projected additive gaussian processes for regression. In *International Conference on Machine Learning*, pages 2453–2463. PMLR, 2020.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Andrew O. Finley, Huiyan Sang, Sudipto Banerjee, and Alan E. Gelfand. Improving the performance of predictive process modeling for large datasets. *Computational Statistics & Data Analysis*, 53(8):2873–2884, 6 2009. ISSN 0167-9473. doi: 10.1016/j.csda.2008.09.008. URL `http://www.ncbi.nlm.nih.gov/pubmed/20016667`.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

Roman Garnett, Michael A Osborne, and Philipp Hennig. Active learning of linear embeddings for gaussian processes. *arXiv preprint arXiv:1310.6740*, 2013.

Joseph Guinness. Permutation and grouping methods for sharpening Gaussian process approximations. *Technometrics*, 60(4):415–429, 2018. doi: 10.1080/00401706.2018.1437476. URL `http://arxiv.org/abs/1609.05372`.

Joseph Guinness. Gaussian process learning via Fisher scoring of Vecchia's approximation. *Statistics and Computing*, 31(25), 2021.

Matthew J. Heaton, Abhirup Datta, Andrew O. Finley, Reinhard Furrer, Joseph Guinness, Rajarshi Guhaniyogi, Florian Gerber, Robert B. Gramacy, Dorit M. Hammerling, Matthias Katzfuss, Finn Lindgren, Douglas W. Nychka, Furong Sun, and Andrew Zammit-Mangion. A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological, and Environmental Statistics*, 24(3):398–425, 2019. doi: 10.1007/s13253-018-00348-w. URL http://arxiv.org/abs/1710.05013.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

Matthias Katzfuss. A multi-resolution approximation for massive spatial datasets. *Journal of the American Statistical Association*, 112(517):201–214, 2 2017. ISSN 0162-1459. doi: 10.1080/01621459.2015.1123632. URL http://www.tandfonline.com/doi/full/10.1080/01621459.2015.1123632.

Matthias Katzfuss and Wenlong Gong. A class of multi-resolution approximations for large spatial datasets. *Statistica Sinica*, 30(4):2203–2226, 2020. doi: 10.1007/s13253-020-00401-7.

Matthias Katzfuss and Joseph Guinness. A general framework for Vecchia approximations of Gaussian processes. *Statistical Science*, 36(1):124–141, 2021. doi: 10.1214/19-STS755. URL http://arxiv.org/abs/1708.06302.

Matthias Katzfuss and Florian Schäfer. Scalable Bayesian transport maps for high-dimensional non-Gaussian spatial fields. *arXiv:2108.04211*, 2021.

Matthias Katzfuss, Joseph Guinness, Wenlong Gong, and Daniel Zilber. Vecchia approximations of Gaussian-process predictions. *Journal of Agricultural, Biological, and Environmental Statistics*, 25(3):383–414, 2020. doi: 10.1007/s13253-020-00401-7.

Matthias Katzfuss, Joseph Guinness, and Earl Lawrence. Scaled Vecchia approximation for fast computer-model emulation. *SIAM/ASA Journal on Uncertainty Quantification*, accepted, 2022. URL http://arxiv.org/abs/2005.00386.

Daniel Kressner. Advanced numerical analysis, May 2015. URL https://www.epfl.ch/labs/anchp/wp-content/uploads/2018/05/AdvancedNA2015.pdf. Lecture Notes, Swiss Federal Institute of Technology Lausanne, https://www.epfl.ch/labs/anchp/wp-content/uploads/2018/05/AdvancedNA2015.pdf.

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

Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian process meets big data: A review of scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. doi: 10.1109/TNNLS.2019.2957109. URL http://arxiv.org/abs/1807.01065.

Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

R. M. Neal. Bayesian learning for neural networks. *Lecture Notes in Statistics*, 1996.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Konstantin Posch, Maximilian Arbeiter, Martin Pleschberger, and Juergen Pilz. Variable selection using nearest neighbor gaussian processes. *arXiv preprint arXiv:2103.14315*, 2021.

J Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005. URL http://dl.acm.org/citation.cfm?id=1194909.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 026218253X. doi: 10.1142/S0129065704001899. URL http://www.gaussianprocess.org/gpml/chapters/RW.pdf.

Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

Huiyan Sang, Mikyoung Jun, and Jianhua Z Huang. Covariance approximation for large multivariate spatial datasets with an application to multiple climate model errors. *Annals of Applied Statistics*, 5(4):2519–2548, 2011. URL http://www.stat.tamu.edu/~mjun/paper/Sang_Jun_Huang.pdf.

Florian Schäfer, Matthias Katzfuss, and Houman Owhadi. Sparse Cholesky factorization by Kullback-Leibler minimization. *SIAM Journal on Scientific Computing*, 43(3):A2019–A2046, 2021a. doi: 10.1137/20M1336254.

Florian Schäfer, T. J. Sullivan, and Houman Owhadi. Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. *Multiscale Modeling & Simulation*, 19(2):688–730, 2021b. doi: 10.1137/19M129526X.

Edward Snelson and Zoubin Ghahramani. Variable noise and dimensionality reduction for sparse gaussian processes. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, page 461–468. AUAI Press, 2006. ISBN 0974903922.

Edward Snelson and Zoubin Ghahramani. Local and global sparse Gaussian process approximations. In *Artificial Intelligence and Statistics 11 (AISTATS)*, 2007.

Michael L. Stein. When does the screening effect hold? *Annals of Statistics*, 39(6):2795–2819, 12 2011. ISSN 0090-5364. doi: 10.1214/11-AOS909. URL http://projecteuclid.org/euclid.aos/1327413769.

Michael L. Stein. Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics*, 8:1–19, 5 2014. ISSN 22116753. doi: 10.1016/j.spasta.2013.06.003. URL http://linkinghub.elsevier.com/retrieve/pii/S2211675313000390https://linkinghub.elsevier.com/retrieve/pii/S2211675313000390.

Michael L. Stein, Zhiyi Chi, and L.J. Welty. Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B*, 66(2):275–296, 2004. URL http://www3.interscience.wiley.com/journal/118808457/abstract.

Wei Sun, Joseph G Ibrahim, and Fei Zou. Genomewide multiple-loci mapping in experimental crosses by iterative adaptive penalized regression. *Genetics*, 185(1):349–359, 2010.

S. Surjanovic and Derek Bingham. Virtual Library of Simulation Experiments: Test Functions and Datasets. http://www.sfu.ca/~ssurjano, 2013. URL http://www.sfu.ca/~ssurjano.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996. doi: 10.1126/science.6254144. URL http://www.nature.com/leu/journal/v27/n3/pdf/leu2012253a.pdf%0Ahttp://www.ncbi.nlm.nih.gov/pmc/articles/PMC3824131/pdf/main.pdf%0Ahttp://science.sciencemag.org/content/sci/210/4470/604.full.pdf%0Ahttp://science.sciencemag.org/content/210/4470/604.long%0Ah.

AV Vecchia. Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society, Series B*, 50(2):297–312, 1988. URL http://www.jstor.org/stable/10.2307/2345768.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784. PMLR, 2015.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

G Yi, JQ Shi, and T Choi. Penalized gaussian process regression and classification for high-dimensional nonlinear data. *Biometrics*, 67(4):1285–1294, 2011.

Florian Ziel. Iteratively reweighted adaptive lasso for conditional heteroscedastic time series with applications to ar–arch type processes. *Computational Statistics & Data Analysis*, 100:773–793, 2016.