Multi-Scale Vecchia Approximations of Gaussian Processes

Jingjie ZHANG and Matthias KATZFUSS

Gaussian processes (GPs) are popular models for functions, time series, and spatial fields, but direct application of GPs is computationally infeasible for large datasets. We propose a multi-scale Vecchia (MSV) approximation of GPs for modeling and analysis of multi-scale phenomena, which are ubiquitous in geophysical and other applications. In the MSV approach, increasingly large sets of variables capture increasingly small scales of spatial variation, to obtain an accurate approximation of the spatial dependence from very large to very fine scales. For a given set of observations, the MSV approach decomposes the data into different scales, which can be visualized to obtain insights into the underlying processes. We explore properties of the MSV approximation and propose an algorithm for automatic choice of the tuning parameters. We provide comparisons to existing approaches based on simulated data and using satellite measurements of land-surface temperature.

Key Words: Covariance approximation; Computational complexity; Large datasets; Sparsity; Spatial statistics.

1. INTRODUCTION

Gaussian processes (GPs) are commonly used as function priors in many application areas such as geospatial analysis (e.g., Banerjee et al. 2004; Cressie and Wikle 2011) and machine learning (e.g., Rasmussen and Williams 2006). GPs are popular because they are flexible, interpretable, and naturally result in probabilistic uncertainty quantification. However, direct application of GPs is too computationally expensive for many modern datasets of interest, as the cost is cubic in the number of data points. Many GP approximations or simplifying assumptions have been proposed, some relying on sparsity (Furrer et al. 2006; Kaufman et al. 2008; Du et al. 2009; Lindgren et al. 2011), some relying on low-rank structure (e.g., Higdon 1998; Wikle and Cressie 1999; Quiñonero-Candela and Rasmussen 2005; Banerjee et al. 2008; Cressie and Johannesson 2008; Katzfuss and Cressie 2011), and some on a combination of the two (e.g., Snelson and Ghahramani 2007; Sang et al. 2011).

J. Zhang · M. Katzfuss (✉), Department of Statistics, Texas A&M University, College Station, USA (E-mail: *katzfuss@gmail.com*).

Journal of Agricultural, Biological, and Environmental Statistics
https://doi.org/10.1007/s13253-022-00488-0

Published online: 08 February 2022

We focus on approximations for a large number of observations of a multi-scale GP, which is defined here as a GP whose covariance function is a sum of covariance functions at different scales, or equivalently, as a sum of independent GPs at different scales. Multi-scale processes are ubiquitous in many geophysical and other applications. For example, environmental processes are often subject to diurnal, seasonal, and multi-year cycles over time (Kim et al. 2007); the atmosphere is affected by micro-scale systems such as clouds and thunderstorms, but also by extratropical cyclones that act on much larger scales (Cotton et al. 2010); and for soil moisture, short-range dependence is governed by surface characteristics such as soil texture, vegetation, and topography, while long-range dependence is due to precipitation (Skøien et al. 2003). GPs whose covariance functions are sums of kernels at different scales are also often used in GP emulation (e.g., Ba and Joseph 2012), astronomy (e.g., Sobolewska et al. 2014), and machine learning (e.g., Rasmussen and Williams 2006; Wilson and Adams 2013; Wilson et al. 2014). Further applications can be found in Ferreira and Lee (2007), for example.

Most of the GP approximations described above can be applied to multi-scale GPs, by simply considering the marginal distribution of the data, which implicitly collapses the processes or covariance functions at different scales into one. While its name might imply differently, this marginal approach is also the one considered in the multi-resolution approximation (Katzfuss 2017; Katzfuss and Gong 2020). In contrast, we will show here that it can be highly advantageous to exploit the multi-scale structure explicitly and to specify a suitable approximation for each scale.

Multi-scale approaches from engineering often do not result in consistent joint statistical models, and they usually focus on the development of coarser representations of the phenomenon of interest in order to obtain fast computational algorithms (e.g., Saquib et al. 1996; Comer and Delp 1999). In statistics, most existing multi-scale approaches use tree-structured models, and work on data collected at different scales. For example, Zhu et al. (2004) develop a multi-scale spatial model for soil data collected at varying resolutions and accuracies, and they define the neighborhood structure using a parent-child relationship in a multi-scale tree structure. Huang et al. (2002) propose a multi-resolution autoregressive tree-structured model for fast and resolution-consistent statistical prediction for satellite data measured at different resolutions. Similar tree-structured approaches can be found in Gotway and Young (2002) and Tzeng et al. (2005). There also exist some literature on multi-scale time-series models (Ferreira et al. 2006). These models couple standard linear models at different time scales via stochastic links across scales. Ferreira and Lee (2007) give an overview of these multi-scale models.

We propose here a multi-scale Vecchia (MSV) approximation for multi-scale GPs observed at point level, which essentially combines suitable Vecchia approximations at each of the different scales. The Vecchia approximation, originally proposed for the data vector directly (i.e., for a single level) in Vecchia (1988), replaces the high-dimensional joint distribution of the entire data vector with a product of univariate conditional distributions, in which each conditional distribution only conditions on a small subset of previous observations in some ordering. This can lead to tremendous computational savings if each conditioning set is small, which can be assumed if the so-called screening effect holds. For certain covariance functions, the prediction at a particular location only depends on

nearby observations, which is known as the screening effect. However, the screening effect is relatively weak when observations include a nugget or noise term. Katzfuss and Guinness (2021) and Katzfuss et al. (2020a) proposed a general Vecchia approach that treated the noise term separately from the continuous covariance component, and thus showed that the screening effect can largely be restored and the conditioning sets can be small.

Our MSV approach essentially extends the general-Vecchia idea to multiple levels: at each level, a suitable Vecchia approximation is found for the process acting at the corresponding scale. Roughly speaking, smooth large-scale processes can be approximated well using low-rank approaches, while non-smooth fine-scale processes often exhibit strong screening effects and thus can be approximated well using small conditioning sets. In this context, a nugget or noise term is the ultimate fine-scale process, which is independent over space and thus does not require any conditioning. As shown by Katzfuss and Guinness (2021), all of these different approximations are merely special cases of the Vecchia approach and thus can be combined into the MSV here.

We describe how to efficiently conduct inference using the MSV, and we provide an algorithm for automatic choice of the number of knot variables and the conditioning set size at each level. This algorithm is also applicable and useful for one-level (Vecchia 1988) or two-level (Katzfuss and Guinness 2021) Vecchia approximations. Our approach also leads to nice decompositions and visualizations of the different scales, which can be highly useful in many scientific contexts. We generally assume the covariance functions at the different levels (and the number of levels) to be known (e.g., from expert knowledge, or by using existing algorithms), and focus on accurate approximation of the resulting spatial dependence; however, we also provide a computationally cheap approximation to the integrated likelihood, which can be employed for parameter inference.

The remainder of this article is organized as follows. Section 2 introduces the multi-scale Vecchia approximation and an algorithm for automatic choice of the tuning parameters. In Sect. 3, we conduct numerical studies and comparisons to existing approaches. Section 4 provides an application of MSV to satellite measurements of land-surface temperature. We conclude in Sect. 5. The appendix contains further derivations and proofs.

2. METHODOLOGY

2.1. A MULTI-SCALE GAUSSIAN PROCESS

Consider a Gaussian process (GP) $z(\cdot) \sim GP(0, C)$ with covariance function C on a domain or spatial region $\mathcal{D} \subset \mathbb{R}^d$. Assume that $z(\cdot)$ is a multi-scale process in the sense that $z(\cdot) = \sum_{\ell=1}^{L} y^{(\ell)}(\cdot)$, where the processes at the individual levels, $y^{(\ell)}(\cdot) \overset{ind.}{\sim} GP(0, C^{(\ell)})$, $\ell = 1, \ldots, L$, are ordered from large scales to fine scales. We think of the scale of a process here in terms of the effective range of its covariance function (i.e., the distance beyond which the correlation drops below a small threshold, such as 0.05), and we assume throughout that $y^{(L)}(\cdot)$ is Gaussian white noise. For example, in spatial statistics, $z(\cdot)$ is often modeled as the sum of a large-scale, fine-scale, and nugget or noise component. A simple toy example is shown in Fig. 1.

Due to the independence assumption of the different processes, the covariance of $z(\cdot)$ is

$$C(\mathbf{s}_i, \mathbf{s}_j) = \sum_{\ell=1}^{L} C^{(\ell)}(\mathbf{s}_i, \mathbf{s}_j), \qquad \mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}. \tag{1}$$

Assume we have n observations $\mathbf{z} = (z_1, \ldots, z_n)^\top$ of $z(\cdot)$, such that $z_i = z(\mathbf{s}_i)$. In general, inference involving n observations of a GP requires $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ time. This is computationally infeasible when n is in the tens of thousands or more, and so for many datasets of interest, GP approximations are necessary.

2.2. MULTI-SCALE VECCHIA APPROXIMATION

To obtain a fast approximation of the GP $z(\cdot)$, one could simply apply an existing GP approximation to $z(\cdot)$ directly, using the "collapsed" covariance function C in (1). However, the main idea of our multi-scale Vecchia (MSV) approximation is that it can often be highly beneficial to consider each covariance $C^{(\ell)}$ separately, and tailor an approximation specifically to each of the L levels. Simply speaking, smooth large-scale components can often be approximated well by a low-rank process relying on a small number of anchoring points or knots, and non-smooth fine-scale components often exhibit strong screening or conditional-independence properties (see Sect. 2.3 for more details), while neither approximation might work well for the sum of the two components.

To specify the MSV, define $\mathbf{y}^{(\ell)} = (y^{(\ell)}(\mathbf{s}_1), \ldots, y^{(\ell)}(\mathbf{s}_n))^\top = (y_1^{(\ell)}, \ldots, y_n^{(\ell)})^\top$ for each level $\ell = 1, \ldots, L-1$. The vector of anchoring points or knots at level ℓ, denoted by $\mathbf{y}_\ell = (y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)})^\top$, is assumed to consist of the latent process evaluated at the first n_ℓ observation locations in the chosen ordering. Stack all variables into a vector $\mathbf{x} = (\mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_{L-1}^\top, \mathbf{z}^\top)^\top$.

The exact distribution of the observation vector $\mathbf{z}$ is given by $f(\mathbf{z}) = \int f(\mathbf{x}) d\mathbf{y}_{1:L-1}$, where

$$f(\mathbf{x}) = \big(\textstyle\prod_{\ell=1}^{L-1} \prod_{i=1}^{n_\ell} f(y_i^{(\ell)} | y_1^{(\ell)}, \ldots, y_{i-1}^{(\ell)})\big)\big(\prod_{i=1}^{n} f(z_i | \mathbf{y}_1, \ldots, \mathbf{y}_{L-1}, z_1, \ldots, z_{i-1})\big).$$

When $i = 1$, the conditioning set of $y_i^{(\ell)}$ is empty, and the conditioning set of z_1 does not include any other z_j. Our MSV approximation is essentially a Vecchia approximation applied to the distribution $f(\mathbf{x})$,

$$\hat{f}(\mathbf{x}) = \Big(\prod_{\ell=1}^{L-1} \prod_{i=1}^{n_\ell} f(y_i^{(\ell)} | N_{\mathbf{y}_i^{(\ell)}})\Big)\Big(\prod_{i=1}^{n} f(z_i | N_{z_i})\Big),$$

where for each $y_i^{(\ell)}$ the full conditioning set $y_1^{(\ell)}, \ldots, y_{i-1}^{(\ell)}$ is replaced by a subset $N_{\mathbf{y}_i^{(\ell)}}$, which denotes the nearest $\min\{m_\ell, i-1\}$ variables in space to variable $y_i^{(\ell)}$ among the previously ordered knot variables $\{y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)}\}$. For each z_i the full conditioning set is replaced by a subset $N_{z_i} = \{N_{z_i}^{(1)}, \ldots, N_{z_i}^{(L-1)}\}$, where $N_{z_i}^{(\ell)} = \{y_i^{(\ell)}\}$ for $i \le n_\ell$, and

$N_{z_i}^{(\ell)}$ consists of the nearest m_ℓ variables in space to variable z_i among the knot variables $\{y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)}\}$ for $i > n_\ell$.

Thus, to specify an MSV approximation, first an ordering of the locations must be chosen, resulting in $\mathbf{s}_1, \ldots, \mathbf{s}_n$. Then, for each level $\ell = 1, \ldots, L-1$, based on having selected n_ℓ and m_ℓ, the Vecchia conditioning set for each variable consists of the nearest m_ℓ previously ordered variables among the n_ℓ knots (i.e., the variables corresponding to the first n_ℓ locations in the ordering). For the data level $\mathbf{z}$, the Vecchia conditioning set for each variable consists of the nearest m_ℓ variables among the n_ℓ knots, $\ell = 1, \ldots, L-1$.

2.3. EXAMPLES OF COVARIANCE APPROXIMATIONS

Many types of covariances can be approximated very well using special cases of the Vecchia approximation. We give some examples here, mostly with isotropic covariance functions, which are specified as functions of the distance $r = \|\mathbf{s}_i - \mathbf{s}_j\|$ between two locations. However, our approach does not require isotropy.

Polynomial Consider a polynomial $y^{(\ell)}(\mathbf{s}) = \mathbf{p}(\mathbf{s})^\top \boldsymbol{\beta}$ as a function of location $\mathbf{s}$ with p coefficients $\boldsymbol{\beta} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$, which is often used to capture a large-scale trend term in spatial applications. Such a polynomial can be approximated exactly using a Vecchia approximation with $n_\ell = m_\ell = p$ (see Proposition 1 in Appendix C). For example, in two-dimensional space, we might set $\mathbf{p}(\mathbf{s}) = (1, s_1, s_2, s_1^2, s_2^2, s_1 s_2)^\top$ for $\mathbf{s} = (s_1, s_2)^\top$ and thus $p = 6$.

Squared exponential The squared exponential covariance function, $C^{(\ell)}(r) \propto \exp(-r^2/\lambda^2)$, leads to covariance matrices with exponentially decaying spectrum. Thus, the resulting covariance matrices are approximately low-rank. A process of rank n_ℓ can be approximated using Vecchia with a coarse grid of n_ℓ knots over $\mathcal{D}$, and by conditioning on all previous variables in the knot set (i.e., $m_\ell = n_\ell$).

Exponential Covariance matrices based on the exponential covariance, $C^{(\ell)}(r) \propto \exp(-r/\lambda)$, have a slowly decaying spectrum, and so a good approximation requires the knot set to be essentially equal to the set of observed locations (i.e., $n_\ell \approx n$). However, the precision matrix (i.e., the inverse of the covariance matrix) is typically approximately sparse, meaning that a strong screening effect holds. In the context of a Vecchia approximation, this allows us to choose the conditioning set to consist of only a small number m_ℓ of nearby locations. For example, in one dimension one can achieve an exact approximation by ordering locations from left to right and only conditioning on the $m_\ell = 1$ previous variable.

Matérn The Matérn class of covariance functions has a smoothness parameter ν, with realizations being k times differentiable if $\nu > k$. It includes the exponential ($\nu = 0.5$) and squared exponential ($\nu = \infty$) covariance as special cases on (almost) opposite ends of the smoothness spectrum. For covariance functions in between these extreme cases, we generally need fewer and fewer knots but (relatively) larger and larger conditioning sets (i.e., smaller n_ℓ but larger m_ℓ/n_ℓ) as ν increases.

Nugget A spatially independent noise term, also called a nugget, is the ultimate fine-scale process with covariance function $C^{(\ell)}(r) \propto \mathbb{1}_{[r=0]}$. Due to the independence, the knot set has to be equal to the observed locations (i.e., $n_\ell = n$), but Vecchia is exact even if $m_\ell = 0$.

2.4. INFERENCE

2.4.1. Matrices Needed for Inference

Similarly to Proposition 1 in Katzfuss and Guinness (2021), we can write the MSV as

$$\begin{aligned}\hat{f}(\mathbf{x}) &= \prod_{\ell=1}^{L-1}\left(\prod_{i=1}^{n_\ell}\mathcal{N}\left(y_i^{(\ell)}|B_i^{(\ell)}N_{\mathbf{y}_i^{(\ell)}}, D_i^{(\ell)}\right)\right)\left(\prod_{i=1}^{n}\mathcal{N}(z_i|B_i^{(L)}N_{z_i}, D_i^{(L)})\right)\\ &= \mathcal{N}_{n+\sum_{\ell=1}^{L-1}n_\ell}(\mathbf{x}|\mathbf{0}, \hat{\mathbf{C}}),\end{aligned}$$

where $\hat{\mathbf{C}}^{-1} = \mathbf{U}\mathbf{U}^\top$, $Cov(y_i^{(\ell)}, y_j^{(\ell)}) = C^{(\ell)}(\mathbf{s}_i, \mathbf{s}_j)$, and for $\ell = 1, 2, \ldots, L-1$,

$$\begin{aligned}B_i^{(\ell)} &= Cov(y_i^{(\ell)}, N_{\mathbf{y}_i^{(\ell)}})\, Cov(N_{\mathbf{y}_i^{(\ell)}}, N_{\mathbf{y}_i^{(\ell)}})^{-1},\\ D_i^{(\ell)} &= Cov(y_i^{(\ell)}, y_i^{(\ell)}) - B_i^{(\ell)}Cov(N_{\mathbf{y}_i^{(\ell)}}, y_i^{(\ell)}),\end{aligned} \tag{2}$$

and for $\ell = L$,

$$\begin{aligned}B_i^{(L)} &= Cov\left(z_i, N_{z_i}\right) Cov\left(N_{z_i}, N_{z_i}\right)^{-1},\\ D_i^{(L)} &= Cov\left(z_i, z_i\right) - B_i^{(L)}Cov\left(N_{z_i}, z_i\right).\end{aligned} \tag{3}$$

The sparse upper-triangular matrix $\mathbf{U}$ can be specified based on the $B_i^{(\ell)}$ and $D_i^{(\ell)}$ as detailed in Appendix A.

2.4.2. Likelihood

Similarly to Katzfuss and Guinness (2021), the likelihood $\hat{f}(\mathbf{z}) = \int \hat{f}(\mathbf{x})d\mathbf{y}_{1:L-1}$ can be computed based on $\mathbf{U}$ as

$$\begin{aligned}-2\log\hat{f}(\mathbf{z}) &= \textstyle\sum_{\ell=1}^{L}\sum_{i=1}^{n_\ell}\log D_i^{(\ell)} + 2\sum_{i=1}^{\sum_{\ell=1}^{L-1}n_\ell}\log\mathbf{V}_{ii} + \tilde{\mathbf{z}}^\top\tilde{\mathbf{z}}\\ &\quad -(\mathbf{V}^{-1}\mathbf{U}_y\tilde{\mathbf{z}})^\top(\mathbf{V}^{-1}\mathbf{U}_y\tilde{\mathbf{z}}) + n\log(2\pi),\end{aligned} \tag{4}$$

where $\mathbf{V} = chol(\mathbf{W})$ is the upper triangular Cholesky factor of $\mathbf{W}{:=}\mathbf{U}_y\mathbf{U}_y^\top$, $\tilde{\mathbf{z}} := \mathbf{U}_z^\top\mathbf{z}$, and $\mathbf{U}_y$ and $\mathbf{U}_z$ are the matrices consisting only of the rows of $\mathbf{U}$ corresponding to $(\mathbf{y}_1, \ldots, \mathbf{y}_{L-1})$ and $\mathbf{z}$, respectively.

This expression of the MSV likelihood can be evaluated cheaply. Thus, while we generally assume model parameters (e.g., in the covariance functions $C^{(\ell)}$) to be fixed here, the MSV likelihood in (4) allows us to carry out frequentist and Bayesian inference on unknown model parameters. Note that there might be identifiability issues when the number of levels

L is large. To avoid this, the parameter spaces are often restricted, sometimes through prior distributions, to ensure identifiability, with lower levels accounting for longer-range dependence (e.g., Ba and Joseph 2012).

2.4.3. Prediction

For prediction at observed and unobserved locations, we first consider the posterior distribution of $\mathbf{y}_{1:L-1} = (\mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_{L-1}^\top)^\top$ given $\mathbf{z}$. In an adaptation of the results in Katzfuss et al. (2020a), we have

$$\mathbf{y}_{1:L-1}|\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}^{-1}),$$

where $\boldsymbol{\mu} = -(\mathbf{V}^\top)^{-1}\mathbf{V}^{-1}\mathbf{U}_y\tilde{\mathbf{z}}$ and $\mathbf{W} = \mathbf{U}_y\mathbf{U}_y^\top$ can be computed cheaply based on $\mathbf{U}$ and $\mathbf{V}$.

Now consider linear combinations of the form $\mathbf{H}\mathbf{y}_{1:L-1}$. For example, we might be interested in inference on each scale $\mathbf{y}_\ell = \mathbf{H}_\ell\mathbf{y}_{1:L-1}$, where $\mathbf{H}_\ell$ is a submatrix of the identity, for $\ell = 1, \ldots, L-1$. Similar to Katzfuss et al. (2020a, Sect. 3.3), we have

$$\mathbf{H}\mathbf{y}_{1:L-1}|\mathbf{z} \sim \mathcal{N}(\mathbf{H}\boldsymbol{\mu}, \boldsymbol{\Sigma}_\mathbf{H}),$$

where the covariance matrix can be computed as $\boldsymbol{\Sigma}_\mathbf{H} = (\mathbf{V}^{-1}\mathbf{H}^\top)^\top(\mathbf{V}^{-1}\mathbf{H}^\top)$, and its diagonal elements can be computed as $diag(var(\mathbf{H}\mathbf{y}_{1:L-1}|\mathbf{z})) = ((\mathbf{V}^{-1}\mathbf{H}^\top)\circ(\mathbf{V}^{-1}\mathbf{H}^\top))^\top\mathbf{1}$, where $\circ$ denotes element-wise multiplication and $\mathbf{1}$ is a vector of ones.

For prediction of $y^{(\ell)}(\cdot)$ at any unobserved location $\mathbf{s}_0$ based on observed location set $\mathcal{S}_\ell$, we show in Appendix B that

$$E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) = C^{(\ell)}(\mathbf{s}_0, \mathcal{S}_\ell)\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}\mathbf{H}_\ell\boldsymbol{\mu} \tag{5}$$

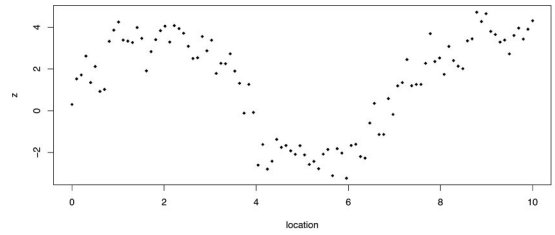
and

$$var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) = C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - \mathbf{c}_\ell(\mathbf{s}_0)^\top\mathbf{c}_\ell(\mathbf{s}_0) + \tilde{\mathbf{c}}_\ell(\mathbf{s}_0)^\top\tilde{\mathbf{c}}_\ell(\mathbf{s}_0), \tag{6}$$

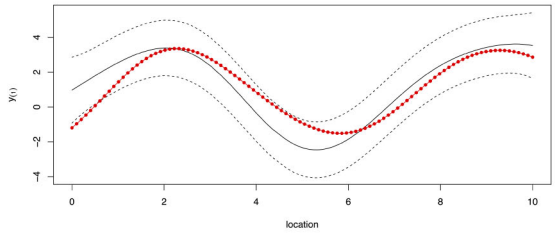
where $\mathbf{c}_\ell(\mathbf{s}_0) = \mathbf{U}^{(\ell)\top}C^{(\ell)}(\mathcal{S}_\ell, \mathbf{s}_0)$, $\tilde{\mathbf{c}}_\ell(\mathbf{s}_0) = \mathbf{V}^{-1}\mathbf{H}_\ell^\top\mathbf{U}^{(\ell)}\mathbf{c}_\ell(\mathbf{s}_0)$, and $\mathbf{U}^{(\ell)}$ is the block of $\mathbf{U}$ corresponding to knot variables at level ℓ. These posterior distributions are illustrated in Fig. 1.

2.5. AUTOMATIC CHOICE OF KNOTS AND CONDITIONING SETS

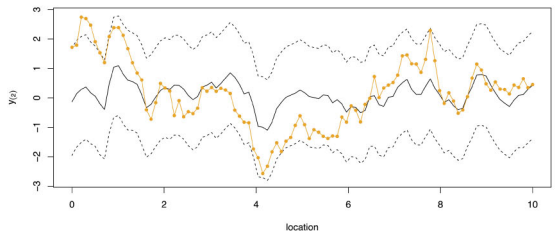
To specify the MSV for a given dataset, for each level we need to determine the knot and conditioning sets, based on an ordering of the locations. To simplify this problem, assume that the locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ are ordered using a maximum–minimum distance (maxmin) ordering (Guinness 2018; Schäfer et al. 2017) and that the variables in each $\mathbf{y}^{(\ell)} = (y^{(\ell)}(\mathbf{s}_1), \ldots, y^{(\ell)}(\mathbf{s}_n))^\top = (y_1^{(\ell)}, \ldots, y_n^{(\ell)})^\top$ are ordered accordingly. The maxmin ordering sequentially selects each variable in the ordering to maximize the minimum distance to all previously ordered variables, and thus attempts to spread out the first k locations in



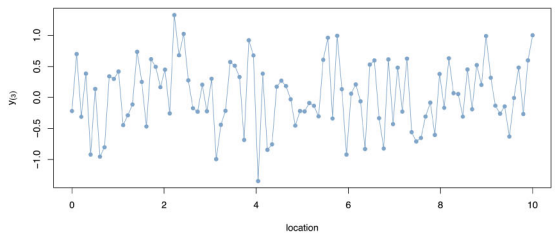
(a) Observations $\mathbf{z}$



(b) $y^{(1)}(\cdot)$; $n_1 = m_1 = 12$



(c) $y^{(2)}(\cdot)$; $m_2 = 1$

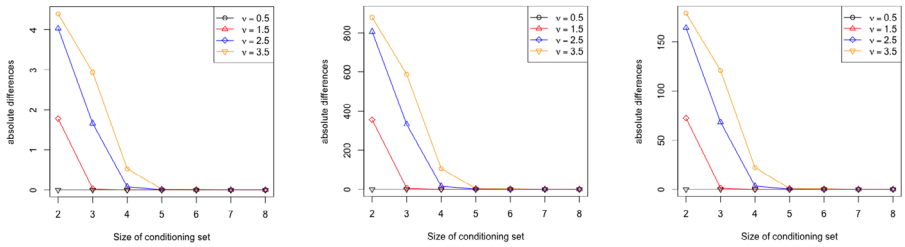


(d) $y^{(3)}(\cdot)$

Figure 1. A simple toy example of **a** observations $\mathbf{z}$ of a multi-scale process obtained as the sum of components (colored dots and lines) with **b** squared exponential, **c** exponential, and **d** nugget covariance, respectively, on a one-dimensional domain $\mathcal{D} = [0, 10]$. Posterior means (black solid lines) and 95% intervals (black dashed lines) for levels 1 and 2 were obtained using MSV as discussed in Sect. 2.4.3. Knot sets and conditioning set sizes were computed using Algorithms 1 and 2 and led to a virtually exact approximation, so that the approximate posterior summaries are basically identical to those obtained using the exact GP .

the ordering as much as possible, for any k. As described in Sect. 2.2, we specify the knot variables $\mathbf{y}_\ell = (y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)})^\top$ as the first n_ℓ variables in this ordering. The conditioning vector $N_{\mathbf{y}_i^{(\ell)}}$ consists of the nearest m_ℓ variables in space to variable $y_i^{(\ell)}$ among previously ordered variables in $\mathbf{y}_\ell$.

Given these constraints, we only need to choose n_ℓ and m_ℓ for each level $\ell = 1, \ldots, L-1$. If we simply set $n_\ell = m_\ell = n$, the approximation will be exact, but this choice leads to computational infeasibility when n is large. Hence, we propose to pick the smallest



(a) $KL(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)}))$ (b) $\sum_{i=1}^{n} \log(D_i^{(\ell)})$ (c) $\sum_{i \in T} \log(D_i^{(\ell)})$

Figure 2. For the KL divergence and two computationally cheaper alternative quantities, differences for subsequent values of the size m_ℓ of the conditioning sets, for a Matérn covariance function with range 1 and different smoothness values ν. Numerically, we show that convergence of KL divergence as a function of m_ℓ is equivalent to convergence of the sum of all log conditional variances, which in turn is closely approximated by the sum of log conditional variances for the last $t = 20$ locations in maxmin ordering .

n_ℓ and m_ℓ at each level such that the improvement in accuracy by increasing n_ℓ and m_ℓ further is negligible. We consider the Kullback–Leibler (KL) divergence between the exact and approximate distribution as a measure of accuracy. Explicit computation of the KL divergence requires $\mathcal{O}(n^3)$ time and is hence impractical for large n, but it turns out that we do not have to calculate it explicitly to implement our desired algorithm.

Theorem 1. *For each level $\ell = 1, \ldots, L-1$, the KL divergence between the true distribution $f(\mathbf{y}^{(\ell)})$ and a Vecchia approximation $\hat{f}(\mathbf{y}^{(\ell)})$ is given by*

$$KL\left(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)})\right) = \frac{1}{2}\sum_{i=1}^{n} \log(D_i^{(\ell)}) - c(f(\mathbf{y}^{(\ell)}))$$

where $D_i^{(\ell)} = Var(y_i^{(\ell)}|N_{\mathbf{y}_i^{(\ell)}})$ is the conditional variance given in (2), and $c(f(\mathbf{y}^{(\ell)}))$ depends on the exact distribution $f(\mathbf{y}^{(\ell)})$ but is constant with respect to m_ℓ, n_ℓ.

The proof can be found in Appendix C. Thus, minimizing (as a function of n_ℓ and m_ℓ) this KL divergence at each level $l = 1, \ldots, L-1$ is equivalent to minimizing the sum of (or each of) the $\log(D_i^{(\ell)})$ over all variables or locations. In practice, to achieve further speed-ups for large datasets, we minimize the conditional variances for a systematically chosen subset of locations, specifically the last t locations in the maxmin ordering, with indices $T = \{n - t + 1, \ldots, n\}$. Figure 2 illustrates this can be a valid approach.

The resulting proposed procedure for automatically choosing the tuning parameters n_ℓ and m_ℓ is described in Algorithm 2, which relies on Algorithm 1 for choosing m_ℓ for a fixed n_ℓ. Note that Algorithm 2 can be run in parallel for each $\ell = 1, 2, \ldots, L-1$.

In Algorithm 1, for given n_ℓ, we choose m_ℓ based on $\left|\frac{\log D_j^{(\ell)}(m_\ell+1) - \log D_j^{(\ell)}(m_\ell)}{\log D_j^{(\ell)}(m_\ell)}\right|$, which is the relative difference of the logarithm of conditional variance $D_j^{(\ell)}$ for $j \in T$. In practice, we choose the size of T as $\min\{1000, n\}$. Note that, especially for large m_ℓ, $Cov(\mathbf{y}_{N_j^{(\ell)}}, \mathbf{y}_{N_j^{(\ell)}})$ can become numerically singular, in which case we have $D_j^{(\ell)} = \texttt{NA}$. But this would also

Algorithm 1 `chooseM`: Automatic choice of m_ℓ for given n_ℓ

Input: Covariance $C^{(\ell)}$, tolerance $\varepsilon > 0$, maximum conditioning set size $m_{\max}$, knot set size n_ℓ
for $m_\ell = 1, 2, \ldots, \min(m_{\max}, n_\ell)$ **do**
  Compute $D_j^{(\ell)}(m_\ell) = var(y_j^{(\ell)} | \mathbf{y}_{N_j^{(\ell)}})$ using n_ℓ, m_ℓ for all $j \in T$
  if $\forall j \in T, \left| \frac{\log D_j^{(\ell)}(m_\ell+1) - \log D_j^{(\ell)}(m_\ell)}{\log D_j^{(\ell)}(m_\ell)} \right| < \varepsilon$ or $D_j^{(\ell)}(m_\ell + 1) = NA$ **then**
    Break
  end if
end for
return m_ℓ and corresponding $D_{j\in T}^{(\ell)}$

Algorithm 2 Automatic choice of n_ℓ and m_ℓ

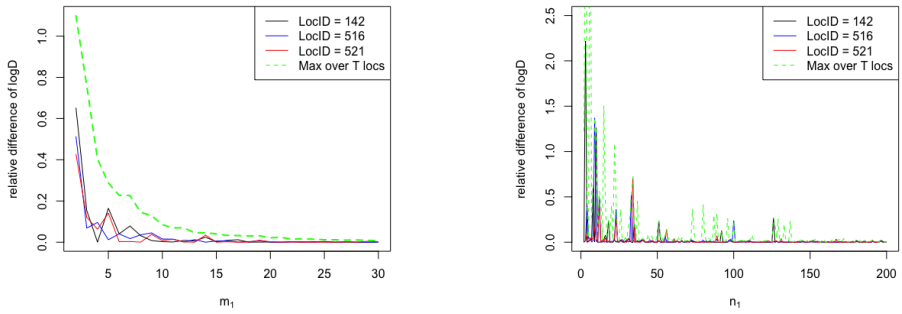
Input: $C^{(\ell)}$, n, ε, $m_{\max}$, T. Default: $m_{\max} = 30$
$n_\ell = 0$, `stepsize` $= 1$, `MinSum` $= \infty$
while $n_\ell \leq n$ **do**
  $n_\ell = n_\ell +$ `stepsize`
  $[m_\ell, D_{j\in T}^{(\ell)}] =$ `chooseM`$(C^{(\ell)}, \varepsilon, m_{\max}, n_\ell)$ (Algorithm 1)
  if `last`$D_j^{(\ell)}$ exists and $\forall j \in T, \left| \frac{\log D_j^{(\ell)} - \log \texttt{last}D_j^{(\ell)}}{\log \texttt{last}D_j^{(\ell)}} \right| < \varepsilon$ or $D_j^{(\ell)} =$ `NA` **then**
    Break
  end if
  if $\sum_{j\in T} D_j^{(\ell)} <$ `MinSum` **then**
    `MinSum` $= \sum_{j\in T} D_j^{(\ell)}$
    `best_m`$_\ell = m_\ell$
    `best_n`$_\ell = n_\ell$
  end if
  `last`$D_{j\in T}^{(\ell)} = D_{j\in T}^{(\ell)}$
  `stepsize` $= 2 *$ `stepsize`
end while
return `best_n`$_\ell$, and `best_m`$_\ell$

imply that enlarging the conditioning set does not result in any improvement in the conditional variance, and so the algorithm will stop. Similarly, we also terminate the algorithm if $D_j^{(\ell)}$ is extremely small (smaller than some specified threshold ε).

In Algorithm 2, we start with $n_\ell = 1$, and compute $D_{j\in T}^{(\ell)}$ using Algorithm 1. To speed up the algorithm, we double the step size of n_ℓ and compute the corresponding $D_{j\in T}^{(\ell)}$ at each iteration until n_ℓ reaches the data size n or the relative difference of logarithm of conditional variance for each location in T converges.

We illustrate Algorithms 1 and 2 for a Matérn covariance in Fig. 3. We also applied Algorithm 2 to the toy example of $n = 100$ simulated observations in Fig. 1, for which a virtually exact approximation was obtained for $n_1 = m_1 = 12$, $n_2 = n = 100$, and $m_2 = 1$.

When optimizing the MSV likelihood in (4) with respect to unknown covariance parameters using an iterative procedure, we recommend not carrying out Algorithm 2 at every iteration, to lower the computational cost. Instead, Algorithm 2 could be carried out only



(a) Algorithms 1 (given $n_1 = 63$)

(b) Algorithm 2

Figure 3. Illustration of Algorithms 1 and 2 for a Matérn covariance with effective range 1, variance 1 and smoothness 3.5 on a two-dimensional domain $\mathcal{D} = [0, 1]^2$ with sample size 900. For illustration purposes, we show the relative difference of logD at three locations, and include the maxima over all locations in the last t locations in the maxmin ordering. **a** Shows that given $n_1 = 63$, the relative difference of log conditional variances converges at $m_1 = 20$. **b** Shows that the relative difference of log conditional variances converges at $n_1 = 148$, which in turn results in a corresponding $m_1 = 21$.

based on the initial and final values parameter values, or at increasing intervals during the parameter optimization (e.g., at iterations 2, 4, 8, 16, …).

2.6. SPARSITY AND COMPUTATIONAL COMPLEXITY

The matrix $\mathbf{U}$ is upper triangular and sparse. The columns of $\mathbf{U}$ corresponding to $\mathbf{y}_\ell = (y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)})^\top$ have at most m_ℓ nonzero off-diagonal entries per column, and so they can be computed in $\mathcal{O}(n_\ell m_\ell^3)$ time. Each z_i may condition on m_ℓ variables at each level $\ell = 1, \ldots, L-1$, but the levels are independent, and so the matrix $Cov(N_{z_i}, N_{z_i})$ in $B_i^{(L)}$ in (3) is block-diagonal. Hence, computing the columns of $\mathbf{U}$ corresponding to $\mathbf{z}$ takes at most $\mathcal{O}(n \sum_{\ell=1}^{L-1} m_\ell^3)$ time; however, the actual computing time can be much lower, because for any $i \leq n_\ell$, we can simply use $N_{z_i}^{(\ell)} = \{y_i^{(\ell)}\}$.

Thus, $\mathbf{U}$ is highly sparse and can be calculated quickly. This often also results in a sparse Cholesky factor $\mathbf{V}$ of $\mathbf{U}\mathbf{U}^\top$, based on the use of ordering algorithms such as approximate minimum degree. In addition, in-fill can be avoided completely through the use of an incomplete Cholesky algorithm, often without introducing significant additional error (Schäfer et al. 2020).

The complexity of each iteration in Algorithm 1 is $\mathcal{O}(tm_\ell^3)$, and so the overall complexity of Algorithm 1 is $\mathcal{O}(tm_\ell^4)$, which is Line 5 in Algorithm 2. Because the step size is doubled at each iteration (Line 15), the overall complexity of Algorithm 2 is $\mathcal{O}(tm_\ell^4 \log n)$.

3. NUMERICAL COMPARISON

We considered simulated data from Gaussian processes with $L = 3$ levels with a Matérn, exponential, and nugget covariance, respectively. We compared the following approaches:

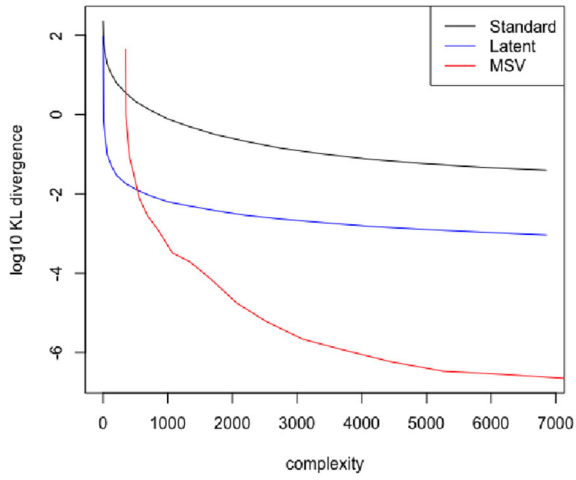


Figure 4. Comparison of KL divergence (on a log scale) against computational complexity for simulated data on a one-dimensional domain $\mathcal{D} = [0, 10]$ with $n = 900$ from a 3-level GP with Matérn (smoothness 2.5, variance 1 and effective range 5), exponential (variance 0.3^2 and effective range 2.996), and nugget (0.1^2) covariance .

Standard: The original Vecchia approximation (Vecchia 1988), which from our perspective is a 1-level Vecchia approximation that is applied directly to the covariance function of the data, obtained by collapsing all levels into one as in (1).

Latent: The latent Vecchia approach (e.g., Datta et al. 2016; Katzfuss and Guinness 2021) can be viewed as a 2-level Vecchia approximation, for which the second level must be Gaussian white noise, $\tilde{C}^{(2)} = C^{(L)}$, and so all other levels in our model are collapsed into one, $\tilde{C}^{(1)} = \sum_{\ell=1}^{L-1} C^{(\ell)}$.

MSV: The multi-scale Vecchia approximation proposed in previous sections, here with $L = 3$ levels.

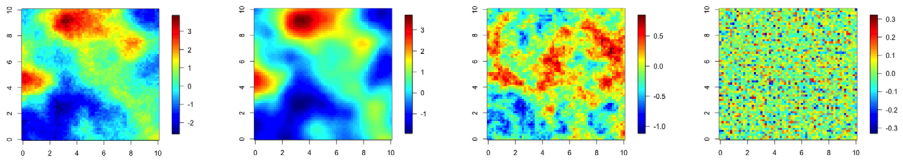
As all approaches can be highly accurate but also slow for large conditioning-set sizes, we compared the KL divergence to the true distribution as a function of computational complexity, which was taken to be nm^3, $n(m^3 + 1)$, and $n \sum_{\ell=1}^{L-1} m_\ell^3$ for Standard, Latent, and MSV, respectively. Standard and Latent only use a single conditioning-set size m; for MSV, we ran Algorithm 2 for various ε values, and then computed the complexity and KL divergence based on the resulting values of n_ℓ and m_ℓ.

Figure 4 shows a comparison on a one-dimensional domain with a relatively small data size of $n = 900$, which allowed us to compute the exact KL divergence. MSV clearly outperformed the other approaches, except for the very-low-complexity setting.

Then, we considered larger datasets of size $n = 6{,}400$ on a two-dimensional domain. One simulated dataset is illustrated in Fig. 5. Figure 6 shows comparisons in terms of KL divergence; to avoid the high computational cost of repeatedly calculating the exact KL divergence, we approximated it by subtracting each method's loglikelihood from the loglikelihood for MSV with the largest possible conditioning sets.

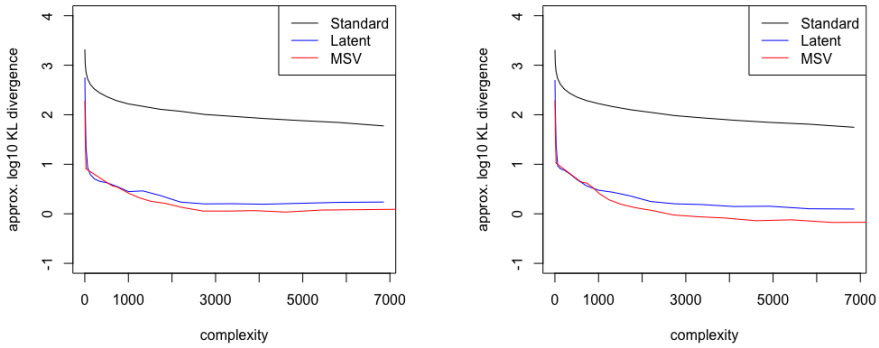
MSV was again more accurate than Latent for a given computational complexity, and both methods strongly outperformed Standard Vecchia.

The focus of our paper is on approximating a given covariance structure, including a given number of levels L, and hence, this is what we examined in our simulation study. MSV can



(a) Observations $\mathbf{z}$ (b) $\mathbf{y}^{(1)}$ (c) $\mathbf{y}^{(2)}$ (d) $\mathbf{y}^{(3)}$

Figure 5. 2D example of **a** observations $\mathbf{z}$ of a three-scale process based on components with **b** Matérn (smoothness 2.5, variance 1 and effective range 5), **c** exponential (variance 0.3^2 and effective range 3), and **d** nugget(0.1^2) covariance, respectively, on a two-dimensional domain $\mathcal{D} = [0, 10]^2$.



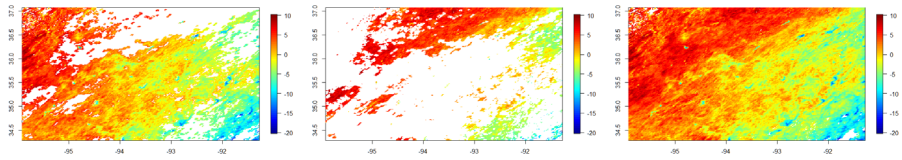
(a) Matérn effective range $\lambda_1 = 5$ (b) Matérn effective range $\lambda_1 = 8$

Figure 6. Comparison of KL divergence (on a log scale) against computational complexity for simulated data on a two-dimensional domain $\mathcal{D} = [0, 10]^2$ with $n = 6{,}400$ from a 3-level GP with Matérn (smoothness 2.5, variance 1, and effective range 5 in (**a**) and 8 in (**b**)), exponential (variance 0.3^2 and effective range 2.996), and nugget (0.1^2) covariance .

have any number of levels $L \geq 1$. While we considered $L = 3$ in our numerical examples, this is not necessary. If, for example, in practice the data were generated using $L = 2$ levels, MSV would ideally also use $L = 2$ and be equivalent to Latent; if we artificially forced MSV to use $L = 3$, the results would depend completely on how the "wrong" additional level was specified (i.e., how strong the model misspecification is), and less on how accurately the MSV is approximating this misspecified model. In this paper, we do not address the issue of model misspecification, and we instead focus on the setting of approximating a known multi-level covariance.

4. APPLICATION

We applied the MSV method to 148,309 satellite measurements of daytime land-surface temperatures from Heaton et al. (2019). The observations are Level-3 data obtained by the Terra instrument onboard the MODIS satellite on August 4, 2016, over a latitudinal range of 34.29519 to 37.06811, and a longitude range from -95.91153 to -91.28381. According to the split in Heaton et al. (2019), the training dataset has 105,569 observations, and the testing dataset has 42,740 observations. We considered the centered data obtained by subtracting



(a) training data (b) testing data (c) full data

Figure 7. Centered daytime land surface temperature data measured by the Terra instrument onboard the MODIS satellite on August 4, 2016 .

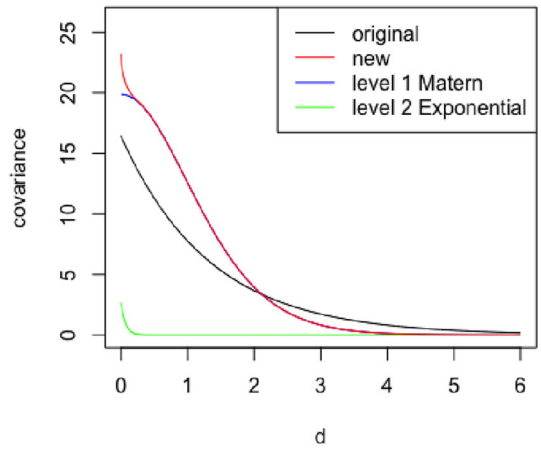
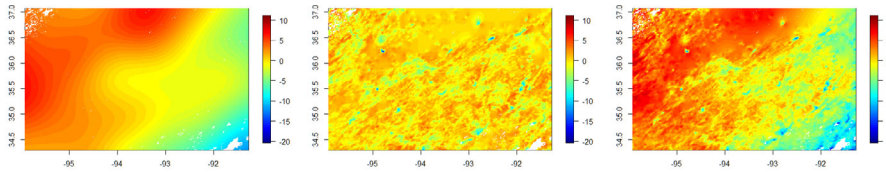


Figure 8. Illustration of the first two levels of the new estimated 3-level covariance function as a function of distance. The original exponential covariance (black curve) was estimated in Heaton et al. (2019), with an estimated variance of 16.40771 and a range of 4/3 .

an overall (constant) mean, which are shown in Fig. 7. For these centered data, we assumed a 3-level Gaussian process model with mean zero and with Matérn, exponential, and nugget covariance, with six unknown parameters. As in Heaton et al. (2019), these six parameters were estimated (jointly) based on a subsample of size 2,500 using the exact GP, resulting in the following parameter estimates: for the Matérn level, variance 19.8656, range 0.3573, smoothness 4.9894; for the exponential level, variance 2.6772, range 0.0665; and nugget variance 0.6917. The resulting covariance function is illustrated in Fig. 8.

For the full training dataset, we used Algorithm 2, with $m_{\max} = 30$, $\varepsilon = 0.001$, and $T = 1{,}000$ (i.e., the last 1000 locations in the maxmin ordering). The algorithm selected $m_1 = 13, n_1 = 16{,}383$ for level 1 (Matérn), and $m_2 = 23, n_2 = n = 105{,}569$ for level 2 (exponential covariance). Given these knots and conditioning sets, we computed the posterior predictive distribution using MSV. To give a rough idea of computing time, it took around eight minutes to compute point predictions at the 42,740 test locations (Intel Core i7-7700K, 4.2GHz, 32GB RAM). Note that both training data and testing data were noisy observations, with the true underlying process unknown. Hence, MSV predictions were obtained at both training locations and testing locations.

The prediction results are shown in Fig. 9. We can see that the first level (Fig. 9a) captures the large-scale spatial dependence, and the second level (Fig. 9b) captures smaller-scale spatial dependence. The overall predicted values (i.e., level 1 plus level 2) given in Fig. 9c are very close to the full original observations in Fig. 7c.



(a) prediction at level 1 (b) prediction at level 2 (c) prediction level 1 + level 2

Figure 9. MSV predictions for MODIS temperature data .

Table 1. Comparison of prediction accuracy scores for MSV and the best-performing existing methods from Heaton et al. (2019) on the MODIS temperature data

Method	MAE	RMSE	INT	CVG
MSV	1.11	1.42	7.32	0.88
FRK	1.96	2.44	14.08	0.79
Gapfill	1.33	1.86	34.78	0.36
LatticeKrig	1.22	1.68	7.55	0.96
LAGP	1.65	2.08	10.81	0.83
Metakriging	2.08	2.50	10.77	0.89
MRA	1.33	1.85	8.00	0.92
NNGP	1.21	1.64	7.57	0.95
Partition	1.41	1.80	10.49	0.86
Pred.Proc.	2.15	2.64	15.51	0.83
SPDE	1.10	1.53	8.85	0.97
Tapering	1.87	2.45	10.31	0.93
Periodic Embedding	1.29	1.79	7.44	0.93

The smaller of MAE, RMSE, INT, the better; the closer of CVG to 0.95, the better

As the dataset considered here is the same one used for the comparison study of many recent methods for large spatial data in Heaton et al. (2019), we also compared the MSV prediction accuracy on the test data to the accuracy of existing methods reported in Heaton et al. (2019). We considered the mean absolute error (MAE), the root mean squared error (RMSE), and the interval score (INT, e.g., Gneiting and Katzfuss 2014) and prediction interval coverage (CVG, i.e., the proportion of intervals containing the test value) for 95% prediction intervals. The results, shown in Table 1, indicate that the multi-level approach (MSV) was highly competitive with the leading existing methods in Heaton et al. (2019).

5. CONCLUSIONS

We proposed a multi-scale Vecchia (MSV) approximation of Gaussian processes for modeling multi-scale phenomena. Our MSV method can tailor suitable Vecchia approximations to the processes acting at different scales. Increasingly large sets of variables capture increasingly small scales of spatial variation, to obtain an accurate approximation of the spatial dependence from very large to very fine scales. We conducted inference using the MSV method, explored approximation properties, and provided an algorithm for automatic choice of the number of knot variables and the conditioning set size at each level. We

compared our method to existing variants of the Vecchia approximation using simulated data. In an application to MODIS daytime land-surface temperature data, our multi-scale method exhibited highly competitive performance relative to a large set of existing methods for large spatial data in Heaton et al. (2019). Our approach also leads to nice visualizations of different scales, which can be highly useful in many scientific contexts.

Our algorithm for determining tuning parameters for the Vecchia approximations at different levels or scales is also applicable and useful for single-level (Vecchia 1988) or two-level (Katzfuss and Guinness 2021) Vecchia approximations. It is also possible to consider hybrids between MSV and, say, single-level Vecchia, by including nearby previously ordered z_j in the conditioning set of z_i. While we have assumed here for simplicity that the data are obtained as an unweighted sum of the latent processes at different scales, extending our methodology to observations that are modeled as (different) linear combinations of the individual scales (including some with zero weight) is straightforward. Our method could also be combined with compositional kernel search (Duvenaud et al. 2013), which expresses the covariance function or kernel of a GP as a sum of kernels, which are obtained using a greedy search over sums and products of a number of base kernels.

ACKNOWLEDGEMENTS

Katzfuss' research was partially supported by National Science Foundation (NSF) Grants DMS–1521676, DMS–1654083, and DMS–1953005. We would like to thank David Jones for helpful comments. Part of our MSV implementation was inspired by the `R` package `GPvecchia` (Katzfuss et al. 2020b), and it relies on a fast maximum–minimum distance ordering algorithm by Florian Schaefer.

[Received February 2021. Revised December 2021. Accepted December 2021.]

A. COMPUTING U

Extending the derivations in Section 2.4.1, the sparse upper triangular matrix $\mathbf{U}$ can be specified by the following rules:
(1) For each $\ell = 1, 2, ..., L-1$, denote $\mathbf{U}^{(\ell)}$ as the block of $\mathbf{U}$ corresponding to level ℓ with size $n_\ell \times n_\ell$. For each $i = 1, 2, ..., n_\ell$,

$$\mathbf{U}^{(\ell)}_{ii} = (D^{(\ell)}_i)^{-1/2}.$$

For the conditioning set of $y^{(\ell)}_i$, suppose the s-th element in its conditioning set is $y^{(\ell)}_{i'}$, then

$$\mathbf{U}^{(\ell)}_{i'i} = -\{B^{(\ell)}_i\}^s (D^{(\ell)}_i)^{-1/2},$$

where $\{B^{(\ell)}_i\}^s$ is the s-th element of $B^{(\ell)}_i$.
(2) For the data level L, first denote an $n \times n$ diagonal matrix by

$$\mathbf{U}^{(L)(L)} = diag\left((D^{(L)}_1)^{-1/2}, (D^{(L)}_2)^{-1/2}, ..., (D^{(L)}_n)^{-1/2}\right).$$

Next, for $\ell = 1, 2, ..., L-1$, denote an $n_\ell \times n$ matrix $\mathbf{U}^{(L)(\ell)}$ as the block of $\mathbf{U}$ corresponding to $\{N_{z_i}^{(\ell)}, i = 1, 2, ..., n\}$. Then, for each i, suppose the s-th element in $N_{z_i}^{(\ell)}$ is $y_{i'}^{(\ell)}$, then

$$\mathbf{U}_{i'i}^{(L)(\ell)} = -\{B_i^{(L)}\}^s (D_i^{(L)})^{-1/2}.$$

(3) Finally, the matrix $\mathbf{U}$ is

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}^{(1)} & & & \mathbf{U}^{(L)(1)} \\ & \mathbf{U}^{(2)} & & \mathbf{U}^{(L)(2)} \\ & & \ddots & \vdots \\ & & \mathbf{U}^{(L-1)} & \mathbf{U}^{(L)(L-1)} \\ & & & \mathbf{U}^{(L)(L)} \end{pmatrix}$$

All unmentioned entries in $\mathbf{U}$ are 0.

B. PREDICTION AT UNOBSERVED LOCATIONS

For simplicity in the proof, denote $\mathcal{S}$ as the locations corresponding to the knot set at level ℓ. First, we show $\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} = \mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}$. When $\ell = 1$, denote $\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{0} & \mathbf{U}_3 \end{pmatrix}$, where $\mathbf{U}_1 = \mathbf{U}^{(1)}$ is the block of $\mathbf{U}$ corresponding to knot variables at level 1. Then,

$$\hat{\mathbf{C}}^{-1} = \mathbf{U}\mathbf{U}^\top = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{0} & \mathbf{U}_3 \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^\top & \mathbf{0} \\ \mathbf{U}_2^\top & \mathbf{U}_3^\top \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1\mathbf{U}_1^\top + \mathbf{U}_2\mathbf{U}_2^\top & \mathbf{U}_2\mathbf{U}_3^\top \\ \mathbf{U}_3\mathbf{U}_2^\top & \mathbf{U}_3\mathbf{U}_3^\top \end{pmatrix}.$$

Since we can also write $\hat{\mathbf{C}}^{-1}$ as $\hat{\mathbf{C}}^{-1} = \begin{pmatrix} C^{(1)}(\mathcal{S}, \mathcal{S}) & A \\ A^\top & B \end{pmatrix}^{-1} = \begin{pmatrix} E & F \\ F^\top & G \end{pmatrix}$, by the property of matrix inverse in block form, $C^{(1)}(\mathcal{S}, \mathcal{S})^{-1} = E - FG^{-1}F^\top$. Thus we have

$$C^{(1)}(\mathcal{S}, \mathcal{S})^{-1} = \mathbf{U}_1\mathbf{U}_1^\top + \mathbf{U}_2\mathbf{U}_2^\top - \mathbf{U}_2\mathbf{U}_3^\top(\mathbf{U}_3\mathbf{U}_3^\top)^{-1}\mathbf{U}_3\mathbf{U}_2^\top = \mathbf{U}_1\mathbf{U}_1^\top = \mathbf{U}^{(1)}\mathbf{U}^{(1)\top}.$$

For any $\ell > 1$, similar results hold: $C^{(\ell)}(\mathcal{S}, \mathcal{S})^{-1} = \mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}$.
Using Algorithm 2 and achieving a KL divergence of (almost) zero, the MSV approximation based on the knot set $\mathbf{y}_\ell$ at level ℓ is (almost) exact, and so we assume that all information about the process at level ℓ is captured by knot set $\mathbf{y}_\ell$. Then, the posterior mean in (5) at an unobserved location $\mathbf{s}_0$ can be computed as

$$\begin{aligned} E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) =& E\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell, \mathbf{z})|\mathbf{z}\right) = E\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell)|\mathbf{z}\right) \\ =& C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} E(\mathbf{y}_\ell|\mathbf{z}) \\ =& C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}\mathbf{H}_\ell\boldsymbol{\mu}. \end{aligned}$$

The posterior variance in (6) can be computed as

$$
\begin{aligned}
&var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) \\
=&E\left(var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell, \mathbf{z})|\mathbf{z}\right) + var\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell, \mathbf{z})|\mathbf{z}\right) \\
=&E\left(var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell)|\mathbf{z}\right) + var\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell)|\mathbf{z}\right) \\
=&E\left(C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} C^{(\ell)}(\mathcal{S}, \mathbf{s}_0)|\mathbf{z}\right) \\
&+ var\left(C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} \mathbf{y}_\ell|\mathbf{z}\right) \\
=&C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} C^{(\ell)}(\mathcal{S}, \mathbf{s}_0) \\
&+ C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} var(\mathbf{y}_\ell|\mathbf{z})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} C^{(\ell)}(\mathbf{s}_0, \mathcal{S})^\top \\
=&C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top} C^{(\ell)}(\mathcal{S}, \mathbf{s}_0) \\
&+ C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}\boldsymbol{\Sigma}_{\mathbf{H}_\ell}\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top} C^{(\ell)}(\mathbf{s}_0, \mathcal{S})^\top. \qquad (7)
\end{aligned}
$$

The posterior predictive variance for the entire latent process is

$$
\begin{aligned}
&var(y^{(1)}(\mathbf{s}_0) + y^{(2)}(\mathbf{s}_0)|\mathbf{z}) \\
&= E\left(var\left(y^{(1)}(\mathbf{s}_0) + y^{(2)}(\mathbf{s}_0)|\mathbf{y}_1, \mathbf{y}_2, \mathbf{z}\right)|\mathbf{z}\right) + var\left(E\left(y^{(1)}(\mathbf{s}_0) + y^{(2)}(\mathbf{s}_0)|\mathbf{y}_1, \mathbf{y}_2, \mathbf{z}\right)|\mathbf{z}\right) \\
&= E\left(var\left(y^{(1)}(\mathbf{s}_0) + y^{(2)}(\mathbf{s}_0)|\mathbf{y}_1, \mathbf{y}_2\right)|\mathbf{z}\right) + var\left(E\left(y^{(1)}(\mathbf{s}_0) + y^{(2)}(\mathbf{s}_0)|\mathbf{y}_1, \mathbf{y}_2\right)|\mathbf{z}\right) \\
&= E\left(\left(var(y^{(1)}(\mathbf{s}_0)|\mathbf{y}_1) + var(y^{(2)}(\mathbf{s}_0)|\mathbf{y}_2)\right)|\mathbf{z}\right) + var\left(\left(E(y^{(1)}(\mathbf{s}_0)|\mathbf{y}_1) + E(y^{(2)}(\mathbf{s}_0)|\mathbf{y}_2)\right)|\mathbf{z}\right) \\
&= E\left(var(y^{(1)}(\mathbf{s}_0)|\mathbf{y}_1)|\mathbf{z}\right) + E\left(var(y^{(2)}(\mathbf{s}_0)|\mathbf{y}_2)|\mathbf{z}\right) \\
&\quad + var\left(C^{(1)}(\mathbf{s}_0, \mathcal{S}_1)\left(C^{(1)}(\mathcal{S}_1, \mathcal{S}_1)\right)^{-1}\mathbf{y}_1 + C^{(2)}(\mathbf{s}_0, \mathcal{S}_2)\left(C^{(2)}(\mathcal{S}_2, \mathcal{S}_2)\right)^{-1}\mathbf{y}_2|\mathbf{z}\right)
\end{aligned}
$$

The first two terms can be computed similar to (7). The last term is a linear combination of $\mathbf{y}_{1:L-1}$, and so it can be calculated by $var(\mathbf{H}\mathbf{y}_{1:L-1}|z) = (\mathbf{V}^{-1}\mathbf{H}^\top)^\top(\mathbf{V}^{-1}\mathbf{H}^\top)$.

C. PROOFS

Proposition 1. *For a polynomial $y^{(\ell)}(\mathbf{s}) = \mathbf{p}(\mathbf{s})^\top\boldsymbol{\beta}$ as a function of spatial location $\mathbf{s}$ with p coefficients $\boldsymbol{\beta} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$, the corresponding covariance function $C^{(\ell)}(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{p}(\mathbf{s}_i)^\top\boldsymbol{\Sigma}_\beta\mathbf{p}(\mathbf{s}_j)$ can be captured exactly by setting the knot and conditioning set to be any distinct p locations.*

Proof of Proposition 1. Denote any p distinct locations as $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_p\}$. For polynomial $y(\mathbf{s}) = \mathbf{p}(\mathbf{s})^\top\boldsymbol{\beta}$ with $\boldsymbol{\beta} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$, the system of equations $\{\mathbf{p}(\mathbf{s}_1)^\top\boldsymbol{\beta} = y(\mathbf{s}_1), \mathbf{p}(\mathbf{s}_2)^\top\boldsymbol{\beta} = y(\mathbf{s}_2), ..., \mathbf{p}(\mathbf{s}_p)^\top\boldsymbol{\beta} = y(\mathbf{s}_p), \mathbf{p}(\mathbf{s})^\top\boldsymbol{\beta} = y(\mathbf{s})\}$ is equivalent to the system of equations $\{\mathbf{p}(\mathbf{s}_1)^\top\boldsymbol{\beta} = y(\mathbf{s}_1), \mathbf{p}(\mathbf{s}_2)^\top\boldsymbol{\beta} = y(\mathbf{s}_2), ..., \mathbf{p}(\mathbf{s}_p)^\top\boldsymbol{\beta} = y(\mathbf{s}_p)\}$, thus

$P\left(y(\mathbf{s})|y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_p)\right) = 1$. Then, the exact distribution for $\mathbf{y} = (y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_n))$ can be written as

$$\begin{aligned} f(\mathbf{y}) &= \prod_{i=1}^{n} f(y(\mathbf{s}_i)|y(\mathbf{s}_{h_i})) \\ &= f(y(\mathbf{s}_1)) f(y(\mathbf{s}_2)|y(\mathbf{s}_1)) f(y(\mathbf{s}_3)|y(\mathbf{s}_1), y(\mathbf{s}_2)) \cdots f(y(\mathbf{s}_p)|y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_{p-1})) \cdot \\ &\quad \prod_{i=p+1}^{n} f(y(\mathbf{s}_i)|y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_p)), \end{aligned}$$

which equals $\hat{f}(\mathbf{y})$ in Vecchia by setting the knot and conditioning set to be $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_p\}$. Thus, the covariance can be captured exactly. □

Proof of Theorem 1. The following proof is related to Guinness (2018, Thm. 1). Suppose the true covariance is Σ_0 and the approximated covariance is $\hat{\Sigma}$. At each level ℓ, the KL divergence between the two normal distributions can be written as $KL\left(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)})\right) = \frac{1}{2}E\left(-(\mathbf{y}^{(\ell)})^\top \Sigma_0^{-1}\mathbf{y}^{(\ell)}\right) + \frac{1}{2}E\left((\mathbf{y}^{(\ell)})^\top \hat{\Sigma}^{-1}\mathbf{y}^{(\ell)}\right) + \frac{1}{2}\log\frac{|\hat{\Sigma}|}{|\Sigma_0|}$. Since Σ_0 is the true covariance, the first term $E\left(-(\mathbf{y}^{(\ell)})^\top \Sigma_0^{-1}\mathbf{y}^{(\ell)}\right) = -n$. Based on MSV, we have $\log|\hat{\Sigma}| = \sum_{i=1}^{n} \log D_i^{(\ell)}$. Suppose L_0 is the Cholesky factor of Σ_0, then $E\left((\mathbf{y}^{(\ell)})^\top \hat{\Sigma}^{-1}\mathbf{y}^{(\ell)}\right) = tr(UU^T\Sigma_0) = \sum_{i,j}(L_0^T U)_{ij}^2 = n$. Thus, the KL divergence can be written as $KL\left(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)})\right) = \frac{1}{2}\left(-n + n + \sum_{i=1}^{n}\log D_i^{(\ell)} - \log|\Sigma_0|\right) = \frac{1}{2}\sum_{i=1}^{n}\log D_i^{(\ell)} - constant.$ □

REFERENCES

Ba S, Joseph VR (2012) Composite Gaussian process models for emulating expensive functions. Ann Appl Stat 6(4):1838–1860

Banerjee S, Carlin BP, Gelfand AE (2004) Hierarchical modeling and analysis for spatial data. Chapman & Hall, Cambridge

Banerjee S, Gelfand AE, Finley AO, Sang H (2008) Gaussian predictive process models for large spatial data sets. J Roy Stat Soc B 70(4):825–848

Comer ML, Delp EJ (1999) Segmentation of textured images using a multiresolution gaussian autoregressive model. IEEE Trans Image Process 8(3):408–420

Cotton WR, Bryan G, Van den Heever SC (2010) Storm and cloud dynamics, volume 99. Academic Press

Cressie N, Johannesson G (2008) Fixed rank kriging for very large spatial data sets. J Roy Stat Soc B 70(1):209–226

Cressie N, Wikle CK (2011) Statistics for spatio-temporal data. Wiley, Hoboken, NJ

Datta A, Banerjee S, Finley AO, Gelfand AE (2016) Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. J Am Stat Assoc 111(514):800–812

Du J, Zhang H, Mandrekar VS (2009) Fixed-domain asymptotic properties of tapered maximum likelihood estimators. Ann Stat 37:3330–3361

Duvenaud D, Lloyd JR, Grosse R, Tenenbaum JB, Ghahramani Z (2013) Structure discovery in nonparametric regression through compositional kernel search. In: Proceedings of the 30th international conference on machine learning, vol 28, pp 1166–1174

Ferreira MA, Lee HK (2007) Multiscale modeling: a Bayesian perspective. Springer Science & Business Media, Berlin

Ferreira MA, West M, Lee HK, Higdon DM et al (2006) Multi-scale and hidden resolution time series models. Bayesian Anal 1(4):947–967

Furrer R, Genton MG, Nychka D (2006) Covariance tapering for interpolation of large spatial datasets. J Comput Graph Stat 15(3):502–523

Gneiting T, Katzfuss M (2014) Probabilistic forecasting. Ann Rev Stat Appl 1(1):125–151

Gotway CA, Young LJ (2002) Combining incompatible spatial data. J Am Stat Assoc 97(458):632–648

Guinness J (2018) Permutation and grouping methods for sharpening Gaussian process approximations. Technometrics 60(4):415–429

Heaton MJ, Datta A, Finley AO, Furrer R, Guinness J, Guhaniyogi R, Gerber F, Gramacy RB, Hammerling D, Katzfuss M, Lindgren F, Nychka DW, Sun F, Zammit-Mangion A (2019) A case study competition among methods for analyzing large spatial data. J Agric Biol Environ Stat 24(3):398–425

Higdon D (1998) A process-convolution approach to modelling temperatures in the North Atlantic Ocean. Environ Ecol Stat 5(2):173–190

Huang H-C, Cressie N, Gabrosek J (2002) Fast, resolution-consistent spatial prediction of global processes from satellite data. J Comput Graph Stat 11(1):63–88

Katzfuss M (2017) A multi-resolution approximation for massive spatial datasets. J Am Stat Assoc 112(517):201–214

Katzfuss M, Cressie N (2011) Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets. J Time Ser Anal 32(4):430–446

Katzfuss M, Gong W (2020) A class of multi-resolution approximations for large spatial datasets. Stat Sin 30(4):2203–2226

Katzfuss M, Guinness J (2021) A general framework for Vecchia approximations of Gaussian processes. Stat Sci 36(1):124–141

Katzfuss M, Guinness J, Gong W, Zilber D (2020) Vecchia approximations of Gaussian-process predictions. J Agric Biol Environ Stat 25(3):383–414

Katzfuss M, Jurek M, Zilber D, Gong W, Guinness J, Zhang J, Schäfer F (2020b) GPvecchia: fast Gaussian-process inference using Vecchia approximations. R package version 0.1.3

Kaufman CG, Schervish MJ, Nychka DW (2008) Covariance tapering for likelihood-based estimation in large spatial data sets. J Am Stat Assoc 103(484):1545–1555

Kim S-W, Yoon S-C, Kim J, Kim S-Y (2007) Seasonal and monthly variations of columnar aerosol optical properties over east asia determined from multi-year modis, lidar, and aeronet sun/sky radiometer measurements. Atmos Environ 41(8):1634–1651

Lindgren F, Rue H, Lindström J (2011) An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. J Roy Stat Soc B 73(4):423–498

Quiñonero-Candela J, Rasmussen CE (2005) A unifying view of sparse approximate Gaussian process regression. J Mach Learn Res 6:1939–1959

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. MIT Press, Cambridge

Sang H, Jun M, Huang JZ (2011) Covariance approximation for large multivariate spatial datasets with an application to multiple climate model errors. Ann Appl Stat 5(4):2519–2548

Saquib SS, Bouman CA, Sauer K (1996) A non-homogeneous mrf model for multiresolution bayesian estimation. In: Proceedings of 3rd IEEE international conference on image processing, vol 2, pp 445–448. IEEE

Schäfer F, Katzfuss M, Owhadi H (2020) Sparse Cholesky factorization by Kullback-Leibler minimization. arXiv:2004.14455

Schäfer F, Sullivan TJ, Owhadi H (2017) Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. arXiv:1706.02205

Skøien JO, Blöschl G, Western A (2003) Characteristic space scales and timescales in hydrology. Water Resour Res, 39(10)

Snelson E, Ghahramani Z (2007) Local and global sparse Gaussian process approximations. In: Artif Intell Stati 11 (AISTATS)

Sobolewska MA, Siemiginowska A, Kelly BC, Nalewajko K (2014) Stochastic modeling of the Fermi/LAT γ-ray blazar variability. Astrophys J, 786(143)

Tzeng S, Huang H-C, Cressie N (2005) A fast, optimal spatial-prediction method for massive datasets. J Am Stat Assoc 100(472):1343–1357

Vecchia A (1988) Estimation and model identification for continuous spatial processes. J Roy Stat Soc B 50(2):297–312

Wikle CK, Cressie N (1999) A dimension-reduced approach to space-time Kalman filtering. Biometrika 86(4):815–829

Wilson AG, Adams RP (2013) Gaussian process kernels for pattern discovery and extrapolation. In: Proceedings of the 30th international conference on machine learning

Wilson AG, Gilboa E, Nehorai A, Cunningham JP (2014) Fast kernel learning for multidimensional pattern extrapolation. In: Advances in neural information processing systems, pp 3626–3634

Zhu J, Morgan CL, Norman JM, Yue W, Lowery B (2004) Combined mapping of soil properties using a multi-scale tree-structured spatial model. Geoderma 118(3–4):321–334