Stat Comput (2017) 27:363–375
DOI 10.1007/s11222-016-9627-4

Parallel inference for massive distributed spatial data using low-rank models

Matthias Katzfuss[1] · **Dorit Hammerling**[2]

Received: 7 November 2014 / Accepted: 20 January 2016 / Published online: 9 February 2016

Abstract Due to rapid data growth, statistical analysis of massive datasets often has to be carried out in a distributed fashion, either because several datasets stored in separate physical locations are all relevant to a given problem, or simply to achieve faster (parallel) computation through a divide-and-conquer scheme. In both cases, the challenge is to obtain valid inference that does not require processing all data at a single central computing node. We show that for a very widely used class of spatial low-rank models, which can be written as a linear combination of spatial basis functions plus a fine-scale-variation component, parallel spatial inference and prediction for massive distributed data can be carried out exactly, meaning that the results are the same as for a traditional, non-distributed analysis. The communication cost of our distributed algorithms does not depend on the number of data points. After extending our results to the spatio-temporal case, we illustrate our methodology by carrying out distributed spatio-temporal particle filtering inference on total precipitable water measured by three different satellite sensor systems.

Keywords Distributed computing · Gaussian process · Particle filter · Predictive process · Spatial random effects model · Spatio-temporal statistics

✉ Matthias Katzfuss
katzfuss@gmail.com

[1] Department of Statistics, Texas A&M University, College Station, USA

[2] Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, USA

1 Introduction

While data storage capacity and data generation have increased by a factor of thousands in the past decade, the data transfer rate has increased by a factor of less than ten (Zhang 2013).

It is therefore of increasing importance to develop analysis tools that minimize the movement of data and perform necessary computations in parallel where the data reside (e.g., Fuller and Millett 2011). Here we consider two situations in which *distributed data* can arise:

Situation 1: Several massive datasets that are stored in separate data centers (servers) are all relevant to a given problem, and moving them to one central computing node for analysis is either too costly due to their large size or not desirable for other reasons such as unnecessary duplicated storage requirements. The goal then is to move the analysis to the data instead of the other way around (e.g., Shoshani et al. 2010).

Situation 2: All data relevant to a given problem are originally stored in the same location, but a "divide-and-conquer" approach with several nodes working in parallel on different chunks of the data is necessary, to achieve sufficiently fast computation or because the entire dataset is too large for a single machine to hold in working memory.

The goal in both of these situations is to obtain valid inference based on all data at a number of computers or servers, without moving the individual datasets between servers. The focus in this article is on Situation 1, but all results are also applicable to Situation 2 without modification. In the spatial and environmental sciences, both of the described distributed-data situations arise frequently. Because analy-

sis of a spatial dataset of size n usually involves the data covariance matrix that has n^2 elements, Situation 2 applies to datasets of even moderate size. Situation 1 arises when several datasets containing information about a particular environmental variable are stored in different data centers throughout the US or the world, and we aim to obtain spatial inference and prediction based on all of them. For example, hundreds of millions of remotely sensed measurements of sea surface temperature per day are available both from the National Oceanic and Atmospheric Administration's (NOAA's) Advanced Very High Resolution Radiometer and from the National Aeronautics and Space Administration's (NASA's) Moderate Resolution Imaging Spectroradiometer. Measurements of column-integrated carbon dioxide are obtained by NASA's Orbiting Carbon Observatory-2 and Atmospheric InfraRed Sounder, Japan's Greenhouse Gases Observing Satellite, and other instruments.

With such satellite data, analyzing the data where they reside is especially important. It not only makes costly data movement and duplicate storage unnecessary, but also avoids (re)transfers of large amounts of data after changes in the retrieval algorithms, which occur quite regularly.

In this article, we will illustrate our methodology by making on-line spatio-temporal inference on a spatial variable called total precipitable water, based on measurements made by three major sensor systems stored at three associated data centers.

We consider here spatial low-rank models that consist of a component that can be written as a linear combination of spatial basis functions and a spatially independent fine-scale-variation term. Despite some recent criticism of their ability to approximate the likelihood of spatial processes with parametric covariances in certain low-noise situations (Stein 2014), low-rank models are a very widely used class of models for large spatial datasets (see Sect. 2 below) because of their scalability for massive data sizes, and their predictive performance has been shown to compare favorably to other approaches in certain situations (Bradley et al. 2014). Note that here we do not advocate for or propose a particular spatial low-rank model—rather, we are presenting distributed algorithms for inference that are applicable to all members of the class of spatial low-rank models.

We show that basic inference for low-rank models can be carried out *exactly* for massive distributed spatial data, while only relying on (*parallel*) *local* computations at each server. In situations where a moderate, fixed number of basis functions is sufficient, the time complexity is linear in the number of measurements at each server, while the communication cost does not depend on the data size at all. Based on this main algorithm, we derive further algorithms for parameter inference and spatial prediction that are similarly well-suited for massive distributed data, and we extend the results to the spatio-temporal case. The results of our parallel distributed algorithms are exactly the same as those obtained by a traditional, non-distributed analysis with all data on one computational node, and so we do *not* ignore spatial dependence between the data at different servers.

General-purpose computer-science algorithms for massive distributed data are not well suited to the distributed-spatial-data problem described above, as solving the linear systems required for prediction and likelihood evaluation would involve considerable movement of data or intermediary results. In the engineering literature, there has been some work on distributed Kalman filters for spatial prediction based on measurements obtained by robotic sensors (Cortés 2009; Xu and Choi 2011; Graham and Cortés 2012), but because the sensors are typically assumed to collect only one measurement at a time, we are not aware of any treatment of the case where the individual datasets are massive.

In the statistics literature, we are also not aware of previous treatment of the distributed-spatial-data problem of Situation 1, although it is possible to adapt some approaches proposed for analyzing (non-distributed) massive spatial data to the distributed case—which is what we are doing with low-rank models in this article. The most obvious other approach is to simply approximate the likelihood for parameter estimation by dividing the data into blocks and then treating the blocks as independent, where in the distributed context each block would correspond to one of the distributed datasets. However, in most applications the distributed datasets were not necessarily collected in distinct spatial regions, and therefore block-independence approaches might ignore significant dependence between different blocks if there is substantial overlap in spatial coverage of the blocks. While methods such as composite likelihoods (e.g., Vecchia 1988; Curriero and Lele 1999; Stein et al. 2004; Caragea and Smith 2007, 2008; Bevilacqua et al. 2012; Eidsvik et al. 2014) have been proposed to allow for some dependence between blocks, it is not clear how well these methods would work in our context, and how spatial predictions at unobserved locations should be obtained (e.g., to which block does the prediction location belong?). Other efforts to implement parallel algorithms for large spatial datasets (e.g., Lemos and Sansó 2009) also exploit being able to split the data by spatial subregions and hence might not be suitable to distributed data in Situation 1.

This article is organized as follows. We begin with a brief review of low-rank spatial models in Sect. 2. We then focus on the distributed-data setting, describing a basic parallel algorithm for inference (Sect. 3), discussing inference on model parameters and presenting important simplifications for fixed basis functions (Sect. 4), describing how to do spatial prediction (Sect. 5), and extending the methodology to the spatio-temporal setting (Sect. 6). We present an application to total precipitable water measured by three sensor systems (Sect. 7), and a simulation study exploring the effect

of parallelization on computation time (Sect. 8). We conclude in Sect. 9.

2 Spatial low-rank models

We are interested in making inference on a spatial process $\{y(\mathbf{s}): \mathbf{s} \in \mathcal{D}\}$, or $y(\cdot)$, on a continuous (non-gridded) domain $\mathcal{D}$, based on a massive number of measurements, $\mathbf{z}_{1:J} := (\mathbf{z}_1', \ldots, \mathbf{z}_J')'$, stored on J different servers or data centers, where $\mathbf{z}_j := (z(\mathbf{s}_{j,1}), \ldots, z(\mathbf{s}_{j,n_j}))'$ is stored on server j (see Figure 1), and the total number of measurements at locations $\{\mathbf{s}_{j,i} \in \mathcal{D} : i = 1, \ldots, n_j;\ j = 1, \ldots, J\}$ is given by $n := \sum_{j=1}^{J} n_j$. Note that the ordering of the servers is completely arbitrary and does not affect the results in any way. We assume that we have additive and spatially independent measurement error, such that

$$z(\mathbf{s}_{j,i}) = y(\mathbf{s}_{j,i}) + \epsilon(\mathbf{s}_{j,i}), \tag{1}$$

for all $i = 1, \ldots, n_j$ and $j = 1, \ldots, J$, where $\epsilon(\mathbf{s}_{j,i}) \sim N(0, v_\epsilon(\mathbf{s}_{j,i}))$ is independent of $y(\cdot)$, and the function $v_\epsilon(\cdot)$ is known. In practice, if $v_\epsilon(\cdot)$ is unknown, one can set $v_\epsilon(\cdot) \equiv \sigma_\epsilon^2$, and then estimate σ_ϵ^2 by extrapolating the variogram to the origin (Kang et al. 2009). Because the measurements in (1) are at point level and not on a grid, we assume for simplicity that no two measurement locations coincide exactly.

The true process $y(\cdot)$ is assumed to follow a spatial low-rank model of the form,

$$y(\mathbf{s}) = \mathbf{b}(\mathbf{s})'\boldsymbol{\eta} + \delta(\mathbf{s}), \quad \mathbf{s} \in \mathcal{D}, \tag{2}$$

where $\mathbf{b}(\cdot)$ is a vector of r spatial basis functions with $r << n$, $\boldsymbol{\eta} \sim N_r(\boldsymbol{\nu}_0, \mathbf{K}_0)$, and often $\boldsymbol{\nu}_0 = \mathbf{0}$. The fine-scale variation $\delta(\mathbf{s}) \sim N(0, v_\delta(\mathbf{s}))$ is spatially independent and independent of $\boldsymbol{\eta}$. Note that we did not include a spatial trend in (2), as any linear trend of the form $\mathbf{x}(\cdot)'\boldsymbol{\beta}$, where $\mathbf{x}(\cdot)$ is a vector of spatial covariates, can simply be absorbed into $\mathbf{b}(\cdot)'\boldsymbol{\eta}$ if we assign a normal prior distribution to $\boldsymbol{\beta}$.

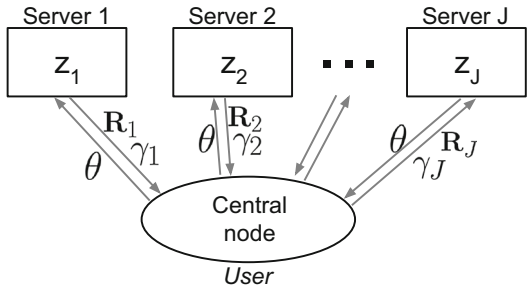

Fig. 1 An illustration of the set-up for distributed data with a central node and data stored at J servers. The quantities to be transferred are described in Algorithm 1

Low-rank models of the form (2) are popular because they do not assume stationarity, and, for a fixed number of basis functions, the time complexity to obtain exact spatial predictions is linear in the number of measurements, hence offering excellent scalability for massive datasets. Many widely used classes of spatial models are of the form (2), such as the spatial random effects model (Cressie and Johannesson 2008), discretized convolution models (e.g., Higdon 1998; Calder 2007; Lemos and Sansó 2009), and the predictive process (Banerjee et al. 2008; Finley et al. 2009). Basis functions that have been used in (2) include empirical orthogonal functions (e.g., Mardia et al. 1998; Wikle and Cressie 1999), Fourier basis functions (e.g., Xu et al. 2005), W-wavelets (e.g., Shi and Cressie 2007; Cressie et al. 2010; Kang and Cressie 2011), and bisquare functions (e.g., Cressie and Johannesson 2008; Katzfuss and Cressie 2011, 2012).

Note that our methodology described in the following sections is applicable to any of these parameterizations of (2), and we do not advocate for a particular model over others. Hence, we work with the general class of spatial low-rank models in (2), and we only assume that there is some parameter vector, $\boldsymbol{\theta}$, that determines $\mathbf{b}(\cdot)$, $\mathbf{K}_0$, and $v_\delta(\cdot)$.

3 Distributed spatial inference: main algorithm

We will now discuss how to obtain the posterior distribution, $[\boldsymbol{\eta}|\mathbf{z}_{1:J}]$, of the random-effects vector of basis-function weights, $\boldsymbol{\eta}$, by performing parallel computations at each server j that use only the local data $\mathbf{z}_j$. Throughout this section, we will treat the parameter vector $\boldsymbol{\theta}$ as fixed and known, with parameter inference to be discussed in Sect. 4.

First, define $\mathbf{B}_{1:J} = (\mathbf{B}_1', \ldots, \mathbf{B}_J')'$ and $\mathbf{V}_{1:J} = blockdiag(\mathbf{V}_1, \ldots, \mathbf{V}_J)$, where $\mathbf{z}_j$, $\mathbf{B}_j := (\mathbf{b}(\mathbf{s}_{j,1}), \ldots, \mathbf{b}(\mathbf{s}_{j,n_j}))'$, and $\mathbf{V}_j := diag(v_\delta(\mathbf{s}_{j,1}) + v_\epsilon(\mathbf{s}_{j,1}), \ldots, v_\delta(\mathbf{s}_{j,n_j}) + v_\epsilon(\mathbf{s}_{j,n_j}))$ are the local quantities at server j. Viewing the random-effects vector $\boldsymbol{\eta}$ as a Bayesian parameter with prior $\boldsymbol{\eta} \sim N_r(\boldsymbol{\nu}_0, \mathbf{K}_0)$ and linear Gaussian "likelihood" $\mathbf{z}_{1:J}|\boldsymbol{\eta} \sim N_n(\mathbf{B}_{1:J}\boldsymbol{\eta}, \mathbf{V}_{1:J})$, it is easy to see that the posterior distribution of $\boldsymbol{\eta}$ given the data at all servers is also multivariate normal, $\boldsymbol{\eta}|\mathbf{z}_{1:J} \sim N_r(\boldsymbol{\nu}_z, \mathbf{K}_z)$, where

$$\mathbf{K}_z^{-1} = \mathbf{K}_0^{-1} + \mathbf{R}, \quad \mathbf{R} := \mathbf{B}_{1:J}'\mathbf{V}_{1:J}^{-1}\mathbf{B}_{1:J}$$
$$\boldsymbol{\nu}_z = \mathbf{K}_z(\mathbf{K}_0^{-1}\boldsymbol{\nu}_0 + \boldsymbol{\gamma}), \quad \boldsymbol{\gamma} := \mathbf{B}_{1:J}'\mathbf{V}_{1:J}^{-1}\mathbf{z}_{1:J}.$$

The key to our distributed algorithms is that, due to the diagonal block structure of $\mathbf{V}_{1:J}$, we have

$$\begin{aligned} \mathbf{R} &= \textstyle\sum_{j=1}^{J} \mathbf{B}_j'\mathbf{V}_j^{-1}\mathbf{B}_j =: \sum_{j=1}^{J} \mathbf{R}_j \\ \boldsymbol{\gamma} &= \textstyle\sum_{j=1}^{J} \mathbf{B}_j'\mathbf{V}_j^{-1}\mathbf{z}_j =: \sum_{j=1}^{J} \boldsymbol{\gamma}_j. \end{aligned} \tag{3}$$

Thus, the posterior distribution of $\boldsymbol{\eta}$ can be obtained by properly combining quantities that each only depend on the data

and their spatial locations at one of the servers. This implies the following parallel algorithm to obtain the posterior distribution of $\boldsymbol{\eta}$:

Algorithm 1: Distributed Spatial Inference

1. Do the following *in parallel* for $j = 1, \ldots, J$:
   (a) Move $\boldsymbol{\theta}$ to server j (where data $\mathbf{z}_j$ is stored) and create the matrices $\mathbf{B}_j$ and $\mathbf{V}_j$ there.
   (b) At server j, calculate $\mathbf{R}_j = \mathbf{B}_j'\mathbf{V}_j^{-1}\mathbf{B}_j$ and $\boldsymbol{\gamma}_j = \mathbf{B}_j'\mathbf{V}_j^{-1}\mathbf{z}_j$.
   (c) Transfer the $r \times r$ matrix $\mathbf{R}_j$ and the $r \times 1$ vector $\boldsymbol{\gamma}_j$ back to the central node.
2. At the central node, calculate $\mathbf{K}_z^{-1} = \mathbf{K}_0^{-1} + \sum_{j=1}^J \mathbf{R}_j$ and $\boldsymbol{\nu}_z = \mathbf{K}_z(\mathbf{K}_0^{-1}\boldsymbol{\nu}_0 + \sum_{j=1}^J \boldsymbol{\gamma}_j)$. The posterior distribution of $\boldsymbol{\eta}$ is given by $\boldsymbol{\eta}|\mathbf{z}_{1:J} \sim N_r(\boldsymbol{\nu}_z, \mathbf{K}_z)$.

Algorithm 1 is illustrated in Fig. 1. The overall time complexity is $\mathcal{O}(r^3 + r^2 max_j n_j)$ (specifically, $\mathcal{O}(r^3)$ at the central node and $\mathcal{O}(r^2 n_j)$ at server j), the memory complexity is $\mathcal{O}(r^2)$ at the central node and $\mathcal{O}(rn_j)$ at server j, and we need to move only the $r(r/2+3/2)$ unique elements in $\mathbf{R}_j$ and $\boldsymbol{\gamma}_j$ from each server. Compare this to a non-distributed algorithm that has time complexity $\mathcal{O}(r^3 + r^2 \sum_j n_j)$, memory complexity $\mathcal{O}(rn)$, and requires moving the n measurements (plus their spatial coordinates) to the central node. In summary, if r remains fixed as the data size increases, Algorithm 1 has computational cost that is linear in each n_j, the communication cost does not depend on n at all, and hence it is scalable for massive distributed datasets.

3.1 Reducing communication via sparsity

The required amount of communication and computation for Algorithm 1 can be reduced further if the basis-function matrices $\mathbf{B}_j$ are sparse, resulting in sparse $\mathbf{R}_j$. Several approaches that impose sparsity in basis-function models have recently been proposed (e.g., Lindgren et al. 2011; Nychka et al. 2015). While sparsity in principle allows fast computation even for large r, having $r = \mathcal{O}(n)$ does not allow the reduction in communication desired in our Situation 1 from Sect. 1. Large r also creates problems in the spatio-temporal filtering context described below in Sect. 6, as sparsity in the precision matrix $\mathbf{K}_z^{-1}$ can generally not be maintained after propagation through time.

Returning to the low-rank case with small to moderate r, sparsity can be achieved, for example, by taking the predictive-process approach (see Banerjee et al. 2008 for a detailed definition) with a compactly supported parent covariance function, as follows. Assume a set of knots, $\mathcal{W} := \{\mathbf{w}_1, \ldots, \mathbf{w}_r\}$, and a parent covariance function

$$C(\mathbf{s}_1, \mathbf{s}_2) = \sigma(\mathbf{s}_1)\sigma(\mathbf{s}_2)\rho(\mathbf{s}_1, \mathbf{s}_2), \quad \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D},$$

where ρ is a correlation function. Then the predictive process can be written in the form (2) with

$$\mathbf{b}(\mathbf{s}) := \sigma(\mathbf{s})\left(\rho(\mathbf{s}, \mathbf{w}_1), \ldots, \rho(\mathbf{s}, \mathbf{w}_r)\right)', \ \mathbf{s} \in \mathcal{D}, \tag{4}$$

and the (i, j)th element of $\mathbf{K}_0^{-1}$ given by $\rho(\mathbf{w}_i, \mathbf{w}_j)$ (see, e.g., Katzfuss 2013).

Now, if C is compactly supported with range h, then the (l, m)th element of the matrix $\mathbf{R}_j = \mathbf{B}_j'\mathbf{V}_j^{-1}\mathbf{B}_j$ in (3) can only be nonzero if $||\mathbf{w}_l - \mathbf{w}_m|| < 2h$. Hence, if for a given set of knots, at most v other knots are within a distance of $2h$ of any knot, at most $r(v/2 + 2)$ numbers (including $\boldsymbol{\gamma}_j$) need to be transferred from each server.

4 Parameter inference

So far, we have treated the parameter vector $\boldsymbol{\theta}$ (containing the parameters determining $\mathbf{b}(\cdot)$, $\mathbf{K}_0$, and $v_\delta(\cdot)$) as fixed and known. In practice, of course, this is usually not the case. Fortunately, several commonly used inference approaches can be implemented in a distributed and parallel fashion by extending Algorithm 1 (while still producing the same results as in the traditional, non-distributed setting).

4.1 Parsimonious parameterizations

If the parameter vector $\boldsymbol{\theta}$ is of low dimension (e.g., there are only three parameters in the predictive-process model in (4) with a Matérn parent covariance function), and estimates or posterior distributions of the parameters are not available in closed form, standard numerical likelihood-based inference is one possibility for parameter inference.

As shown in Appendix 1, the likelihood (up to a normalization constant) for the spatial low-rank model in Sect. 2 can be written as,

$$\begin{aligned} -2\log L(\boldsymbol{\theta}) &:= -2\log[\mathbf{z}_{1:J}|\boldsymbol{\theta}] \\ &= -\log|\mathbf{K}_0^{-1}| + \boldsymbol{\nu}_0'\mathbf{K}_0^{-1}\boldsymbol{\nu}_0 \\ &\quad + \log|\mathbf{K}_z^{-1}| - \boldsymbol{\nu}_z'\mathbf{K}_z^{-1}\boldsymbol{\nu}_z + \textstyle\sum_{j=1}^J a_j, \end{aligned} \tag{5}$$

where $a_j := \log|\mathbf{V}_j| + \mathbf{z}_j'\mathbf{V}_j^{-1}\mathbf{z}_j$. This allows carrying out both frequentist and Bayesian inference for distributed data (e.g., by numerical maximization of the likelihood, Metropolis-Hasting sampling, or other approaches). Each iteration of such a parameter-inference procedure consists of carrying out Algorithm 1 (with the addition of calculating

a_j at server j and moving this scalar quantity to the central node), combining the results to evaluate the likelihood (5) at the central node, updating the parameters $\boldsymbol{\theta}$, and sending out the new value of $\boldsymbol{\theta}$ to the servers. This results in a sequential algorithm, for which the (major) calculations at each iteration can be carried out in parallel.

To avoid servers being idle in such a sequential algorithm, we recommend instead the use of an importance or particle sampler. Any of the various such algorithms proposed in the literature can be carried out in the distributed context (with the exact same results), by evaluating the likelihood as in (5). Here is an example of such an algorithm:

Algorithm 2: Distributed Importance Sampler

1. Generate a number of parameter vectors or particles, $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}$, from a suitably chosen proposal distribution, $q(\boldsymbol{\theta})$.
2. Do the following *in parallel* for $j = 1, \ldots, J$ and $m = 1, \ldots, M$:
   (a) Move $\boldsymbol{\theta}^{(m)}$ to server j and create the matrices $\mathbf{B}_j^{(m)}$ and $\mathbf{V}_j^{(m)}$.
   (b) Calculate
   $$\mathbf{R}_j^{(m)} = \mathbf{B}_j^{(m)\prime}(\mathbf{V}_j^{(m)})^{-1}\mathbf{B}_j^{(m)}$$
   $$\boldsymbol{\gamma}_j^{(m)} = \mathbf{B}_j^{(m)\prime}(\mathbf{V}_j^{(m)})^{-1}\mathbf{z}_j$$
   $$a_j^{(m)} = \log|\mathbf{V}_j^{(m)}| + \mathbf{z}_j'(\mathbf{V}_j^{(m)})^{-1}\mathbf{z}_j.$$
   (c) Transfer $\mathbf{R}_j^{(m)}$, $\boldsymbol{\gamma}_j^{(m)}$, and $a_j^{(m)}$ back to the central node.
3. At the central node, for $m = 1, \ldots, M$, calculate $(\mathbf{K}_z^{(m)})^{-1} = (\mathbf{K}_0^{(m)})^{-1} + \sum_{j=1}^J \mathbf{R}_j^{(m)}$, $\boldsymbol{\nu}_z^{(m)} = \mathbf{K}_z^{(m)}((\mathbf{K}_0^{(m)})^{-1}\boldsymbol{\nu}_0^{(m)} + \sum_{j=1}^J \boldsymbol{\gamma}_j^{(m)})$, and
   $$-2\log L(\boldsymbol{\theta}^{(m)}) = -\log|(\mathbf{K}_0^{(m)})^{-1}| + \boldsymbol{\nu}_0^{(m)\prime}(\mathbf{K}_0^{(m)})^{-1}\boldsymbol{\nu}_0^{(m)} + \log|(\mathbf{K}_z^{(m)})^{-1}| - \boldsymbol{\nu}_z^{(m)\prime}(\mathbf{K}_z^{(m)})^{-1}\boldsymbol{\nu}_z^{(m)} + \sum_{j=1}^J a_j^{(m)}.$$
4. The particle approximation of the posterior distribution of $\boldsymbol{\theta}$ takes on the value $\boldsymbol{\theta}^{(m)}$ with probability $w^{(m)} \propto p(\boldsymbol{\theta}^{(m)})L(\boldsymbol{\theta}^{(m)})/q(\boldsymbol{\theta}^{(m)})$ for $m = 1, \ldots, M$, where $p(\boldsymbol{\theta})$ is the prior distribution of the parameters.

The advantage of this parameter-inference approach is that we can carry out calculations for the likelihood evaluations for all particles completely in parallel at all servers (while getting the same results as in the traditional, non-distributed setting).

4.2 Spatial random effects model

The spatial random effects model (Cressie and Johannesson 2008; Katzfuss and Cressie 2009; Kang and Cressie 2011) is a low-rank model of the form (1)–(2), for which the basis functions are known functions (e.g., bisquare functions) that do not depend on unknown parameters, $\mathbf{K}_0$ is a general covariance matrix (i.e., it contains $r(r+1)/2$ parameters), and often $v_\delta(\cdot) \equiv \sigma_\delta^2$. If we also assume $v_\epsilon(\cdot) \equiv \sigma_\epsilon^2$ (or we have transformed the data such that these assumptions hold), we have $\mathbf{V}_j^{-1} = \frac{1}{\sigma_\delta^2+\sigma_\epsilon^2}\mathbf{I}_{n_j}$, and so $\mathbf{R}_j = \frac{1}{\sigma_\delta^2+\sigma_\epsilon^2}\mathbf{B}_j'\mathbf{B}_j$ and $\boldsymbol{\gamma}_j = \frac{1}{\sigma_\delta^2+\sigma_\epsilon^2}\mathbf{B}_j'\mathbf{z}_j$. Since the $\mathbf{B}_j$ in the spatial random effects model are fixed, all that is required for inference on $\boldsymbol{\eta}$ in Algorithm 1 from server j are the fixed quantities $\mathbf{B}_j'\mathbf{z}_j$ and $\mathbf{B}_j'\mathbf{B}_j$, making multiple passes over the servers for parameter inference unnecessary. The only additional information required from server j for evaluating the likelihood (5) is n_j and $\mathbf{z}_j'\mathbf{z}_j$.

If the basis functions do contain unknown parameters, or $v_\epsilon(\cdot)$ is not constant, maximum likelihood estimates can be obtained by deriving a distributed version of the expectation-maximization algorithm of Katzfuss and Cressie (2009, 2011). Each step of the resulting algorithm consists of carrying out Algorithm 1, and then updating the estimates of $\mathbf{K}_0^{-1}$ and σ_δ^2 as

$$\hat{\mathbf{K}}_0^{-1} = (\mathbf{K}_z + \boldsymbol{\nu}_z\boldsymbol{\nu}_z')^{-1} = \mathbf{K}_z^{-1} - \mathbf{q}\mathbf{q}'/(1+\boldsymbol{\nu}_z'\mathbf{q})$$

$$\hat{\sigma}_\delta^2 = \sigma_\delta^2 + \sum_{j=1}^J \frac{\sigma_\delta^4}{n_j}\left(||\mathbf{V}_j^{-1}(\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_z)||^2 - tr(\boldsymbol{\Omega}_j^{-1})\right),$$

where $\mathbf{q}{:=}\mathbf{K}_0^{-1}\boldsymbol{\nu}_0 + \boldsymbol{\gamma}$ and $\boldsymbol{\Omega}_j{:=}\mathbf{B}_j\mathbf{K}_z\mathbf{B}_j' + \mathbf{V}_j$. The expression for $\hat{\sigma}_\delta^2$ above can be derived by obtaining $[\boldsymbol{\delta}_j|\boldsymbol{\eta}, \mathbf{z}_{1:J}]$ and then applying the laws of total expectation and total variance.

By assuming conjugate prior distributions (i.e., an inverse-Wishart distribution for $\mathbf{K}_0$ and an inverse-Gamma distribution for σ_δ^2), Bayesian inference using a Gibbs sampler is also possible.

5 Spatial prediction

The goal in spatial statistics is typically to make spatial predictions of $y(\cdot)$ at a set of prediction locations, $\mathbf{s}_1^P, \ldots, \mathbf{s}_{n_P}^P$, based on all data $\mathbf{z}_{1:J}$, which in technical terms amounts to finding the posterior predictive distribution $[\mathbf{y}^P|\mathbf{z}_{1:J}]$, where $\mathbf{y}^P{:=}(y(\mathbf{s}_1^P), \ldots, y(\mathbf{s}_{n_P}^P))'$. Note that prediction can

be carried out separately, after parameter inference has been completed, and so it suffices to obtain the predictive distribution for the final parameter estimates in a frequentist procedure, or for thinned MCMC samples or for particles with nonzero weight in a Bayesian context.

Because we can write

$$\mathbf{y}^P = \mathbf{B}^P \boldsymbol{\eta} + \boldsymbol{\delta}^P, \tag{6}$$

where $\mathbf{B}^P := (\mathbf{b}(\mathbf{s}_1^P), \ldots, \mathbf{b}(\mathbf{s}_{n_P}^P))'$ and $\boldsymbol{\delta}^P := (\delta(\mathbf{s}_1^P), \ldots, \delta(\mathbf{s}_{n_P}^P))'$, the desired predictive distribution is determined by the joint posterior distribution $[\boldsymbol{\eta}, \boldsymbol{\delta}^P | \mathbf{z}_{1:J}]$.

First, assume that none of the prediction locations exactly coincide with any of the observed locations. This is a reasonable assumption when measurements have point support on a continuous spatial domain, as we have assumed throughout this manuscript. Then it is easy to see that $\boldsymbol{\delta}^P | \mathbf{z}_{1:J} \sim N_{n_P}(\mathbf{0}, \mathbf{V}_\delta^P)$, with $\mathbf{V}_\delta^P := diag\{v_\delta(\mathbf{s}_1^P), \ldots, v_\delta(\mathbf{s}_{n_P}^P)\}$, is conditionally independent of $\boldsymbol{\eta}$ given $\mathbf{z}_{1:J}$. Therefore, spatial prediction reduces to obtaining $\boldsymbol{\nu}_z$ and $\mathbf{K}_z$ using Algorithm 1, and then calculating

$$\mathbf{y}^P | \mathbf{z}_{1:J} \sim N_{n_P}(\mathbf{B}^P \boldsymbol{\nu}_z, \mathbf{B}^P \mathbf{K}_z \mathbf{B}^{P\prime} + \mathbf{V}_\delta^P) \tag{7}$$

at the central node.

Appendix 2 describes how to do spatial prediction when a small number of the observed locations coincide with the desired prediction locations.

6 Spatio-temporal inference

To extend our results to the spatio-temporal case, we consider a spatio-temporal low-rank model in discrete time. In our hierarchical state-space model, the process of interest is given by,

$$y_t(\mathbf{s}) = \mathbf{b}_t(\mathbf{s})' \boldsymbol{\eta}_t + \delta_t(\mathbf{s}), \quad \mathbf{s} \in \mathcal{D};\ t = 1, 2, \ldots,$$

where $\delta_t(\cdot)$ is assumed to be independent over space and time with variance function $v_{\delta,t}(\cdot)$, and the temporal evolution of the low-rank component is given by,

$$\boldsymbol{\eta}_t | \boldsymbol{\theta}_t, \boldsymbol{\eta}_{t-1}, \boldsymbol{\eta}_{t-2}, \ldots \sim N_r(\mathbf{H}_t \boldsymbol{\eta}_{t-1}, \mathbf{U}_t), \quad t = 1, 2, \ldots,$$

where $\boldsymbol{\eta}_0 \sim N_r(\boldsymbol{\nu}_{0,0}, \mathbf{K}_{0,0})$ is the initial state, and $\boldsymbol{\theta}_t$ is a time-varying parameter vector with generic transition equation $p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1})$ and initial value $\boldsymbol{\theta}_0$. The data at server j at time t are given by $\mathbf{z}_{j,t} := (z_t(\mathbf{s}_{1,j,t}), \ldots, z_t(\mathbf{s}_{n_{j,t},j,t}))'$, with

$$z_t(\mathbf{s}_{i,j,t}) = y_t(\mathbf{s}_{i,j,t}) + \epsilon_t(\mathbf{s}_{i,j,t}),$$

for all $i = 1, \ldots, n_{j,t}$, $j = 1, \ldots, J$, and $t = 1, 2, \ldots$, where $\epsilon_t(\mathbf{s}_{i,j,t}) \sim N(0, v_\epsilon(\mathbf{s}_{i,j,t}))$ is independent in space, time, and of $y(\cdot)$. Cressie et al. (2010) called this the spatio-temporal random effects model, but as in the spatial-only case, many different ways of parameterizing such a spatio-temporal low-rank model are possible (see Sect. 7 for an example). We again merely assume that $\mathbf{H}_t$, $\mathbf{U}_t$, $\mathbf{b}_t(\cdot)$, and $v_{\delta,t}(\cdot)$ are known up to the parameter vector $\boldsymbol{\theta}_t$.

6.1 Filtering and smoothing for known parameters

We temporarily assume the parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots$ to be known, or held at a particular set of values at one step of a parameter-inference procedure (see Sect. 6.2 below). We first take an on-line, filtering perspective in time, which means that we are interested at time point t in obtaining the filtering distribution $\boldsymbol{\eta}_t | \mathbf{z}_{1:t} \sim N_r(\boldsymbol{\nu}_{t|t}, \mathbf{K}_{t|t})$, where $\mathbf{z}_{1:t}$ denotes the vector of all data collected at the first t time points. We can obtain $\boldsymbol{\nu}_{t|t}$ and $\mathbf{K}_{t|t}$ using a Kalman filter, for which each update step essentially requires carrying out Algorithm 1:

Algorithm 3: Distributed Spatio-Temporal Filtering

1. For $t = 0$, initialize the algorithm by calculating $\boldsymbol{\nu}_{0|0}$ and $\mathbf{K}_{0|0}$ based on $\boldsymbol{\theta}_0$.
2. At time $t = 1, 2, \ldots$, once the new data $\mathbf{z}_{1,t}, \ldots, \mathbf{z}_{J,t}$ become available:
   (a) Do the following *in parallel* for $j = 1, \ldots, J$:
      i. Move $\boldsymbol{\theta}_t$ to server j and create the matrices $\mathbf{B}_{j,t}$ and $\mathbf{V}_{j,t}$ based on the observed locations at time t.
      ii. At server j, calculate $\mathbf{R}_{j,t} = \mathbf{B}_{j,t}' \mathbf{V}_{j,t}^{-1} \mathbf{B}_{j,t}$ and $\boldsymbol{\gamma}_{j,t} = \mathbf{B}_{j,t}' \mathbf{V}_{j,t}^{-1} \mathbf{z}_{j,t}$.
      iii. Transfer $\mathbf{R}_{j,t}$ and $\boldsymbol{\gamma}_{j,t}$ back to the central node.
   (b) At the central node, calculate the forecast quantities $\boldsymbol{\nu}_{t|t-1} := \mathbf{H}_t \boldsymbol{\nu}_{t-1|t-1}$, $\mathbf{K}_{t|t-1} := \mathbf{H}_t \mathbf{K}_{t-1|t-1} \mathbf{H}_t' + \mathbf{U}_t$, and then the filtering quantities $\mathbf{K}_{t|t}^{-1} = \mathbf{K}_{t|t-1}^{-1} + \sum_{j=1}^J \mathbf{R}_{j,t}$ and $\boldsymbol{\nu}_{t|t} = \mathbf{K}_{t|t}(\mathbf{K}_{t|t-1}^{-1} \boldsymbol{\nu}_{t|t-1} + \sum_{j=1}^J \boldsymbol{\gamma}_{j,t})$. We have $\boldsymbol{\eta}_t | \mathbf{z}_{1:t} \sim N_r(\boldsymbol{\nu}_{t|t}, \mathbf{K}_{t|t})$.

It is interesting to note that Algorithm 1 in Sect. 3 can itself be viewed as a decentralized Kalman filter (Rao et al. 1993) over servers applied to our spatial low-rank model written as a state-space model with an identity evolution equation. Thus, Algorithm 3 is actually the combination of two nested

filters, where each "outer" filtering step over time essentially consists of an "inner" filter over servers as in (3).

In some applications, retrospective smoothing inference based on data collected at T time points might be of interest. Obtaining the smoothing distribution $\boldsymbol{\eta}_t|\mathbf{z}_{1:T} \sim N_r(\boldsymbol{\nu}_{t|T}, \mathbf{K}_{t|T})$ for $t = 1, \ldots, T$, requires forward-filtering using Algorithm 3 and then backward-smoothing at the central node by calculating iteratively for $t = T-1, T-2, \ldots, 1$:

$$\boldsymbol{\eta}_{t|T} = \boldsymbol{\eta}_{t|t} + \mathbf{J}_t(\boldsymbol{\eta}_{t+1|T} - \boldsymbol{\eta}_{t+1|t}),$$
$$\mathbf{K}_{t|T} = \mathbf{K}_{t|t} + \mathbf{J}_t(\mathbf{K}_{t+1|T} - \mathbf{K}_{t+1|t})\mathbf{J}_t',$$

where $\mathbf{J}_t := \mathbf{K}_{t|t}\mathbf{H}_{t+1}'\mathbf{K}_{t+1|t}^{-1}$ (see, e.g., Cressie et al. 2010 p. 732, for more details). Also, note that in the smoothing context, it is not actually necessary to "consolidate" the information at the end of each time point as in Step 2(b) of Algorithm 3 before moving on to the next time point; instead, we can calculate $\mathbf{R}_{j,1}, \ldots, \mathbf{R}_{j,T}$ and $\boldsymbol{\gamma}_{j,1}, \ldots, \boldsymbol{\gamma}_{j,T}$ at each server j, and then directly calculate $\mathbf{K}_{T|T}$ and $\boldsymbol{\nu}_{T|T}$ at the central node.

Because $\delta_t(\cdot)$ is *a priori* independent over time, spatial prediction for each t in the filtering and smoothing context can be carried out as described in Sect. 5 using the filtering or smoothing distribution of $\boldsymbol{\eta}_t$ (i.e., $\boldsymbol{\nu}_{t|t}, \mathbf{K}_{t|t}$ or $\boldsymbol{\nu}_{t|T}, \mathbf{K}_{t|T}$, respectively).

6.2 Spatio-temporal parameter inference

In the filtering context, inference on the parameter vector $\boldsymbol{\theta}_t$ at time point t is typically based on the filtering likelihood,

$$\begin{aligned} -2\log L_t(\boldsymbol{\theta}_t) &:= -2\log[\mathbf{z}_t|\mathbf{z}_{1:t-1}, \boldsymbol{\theta}_t] \\ &= -\log|\mathbf{K}_{t|t-1}^{-1}| + \boldsymbol{\nu}_{t|t-1}'\mathbf{K}_{t|t-1}^{-1}\boldsymbol{\nu}_{t|t-1} \\ &\quad + \log|\mathbf{K}_{t|t}^{-1}| - \boldsymbol{\nu}_{t|t}'\mathbf{K}_{t|t}^{-1}\boldsymbol{\nu}_{t|t} + \textstyle\sum_{j=1}^{J} a_{j,t}, \end{aligned} \tag{8}$$

where $a_{j,t} := \log|\mathbf{V}_{j,t}| + \mathbf{z}_{j,t}'\mathbf{V}_{j,t}^{-1}\mathbf{z}_{j,t}$. This expression of the likelihood can be derived similarly as in the spatial-only case described in Appendix 1. If there are a small number of unknown parameters in the spatio-temporal low-rank model, we again advocate the use of a particle-filtering approach for parameter estimation. Sequential importance sampling with resampling (Gordon et al. 1993) is a natural inference procedure for on-line inference over time. With distributed data, it can be carried out using a straightforward combination of Algorithms 2 and 3:

Algorithm 4: Distributed Spatio-Temporal Particle Filter

1. For $t = 0$, calculate $\boldsymbol{\nu}_{0|0}$ and $\mathbf{K}_{0|0}$ based on initial parameter value $\boldsymbol{\theta}_0$. Then sample M particles $\boldsymbol{\theta}_1^{(1)}, \ldots, \boldsymbol{\theta}_1^{(M)}$ from a suitably chosen proposal distribution $q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_0)$.
2. At time $t = 1, 2, \ldots$, once new data $\mathbf{z}_{1,t}, \ldots, \mathbf{z}_{J,t}$ become available:
   (a) Do the following *in parallel* for $j = 1, \ldots, J$ and $m = 1, \ldots, M$:
      i. Move $\boldsymbol{\theta}_t^{(m)}$ to server j and create the matrices $\mathbf{B}_{j,t}^{(m)}$ and $\mathbf{V}_{j,t}^{(m)}$ based on the observed locations at time t.
      ii. At server j, calculate
      $$\begin{aligned} \mathbf{R}_{j,t}^{(m)} &= \mathbf{B}_{j,t}^{(m)\prime}(\mathbf{V}_{j,t}^{(m)})^{-1}\mathbf{B}_{j,t}^{(m)} \\ \boldsymbol{\gamma}_{j,t} &= \mathbf{B}_{j,t}^{(m)\prime}(\mathbf{V}_{j,t}^{(m)})^{-1}\mathbf{z}_{j,t} \\ a_{j,t}^{(m)} &= \log|\mathbf{V}_{j,t}^{(m)}| + \mathbf{z}_{j,t}'(\mathbf{V}_{j,t}^{(m)})^{-1}\mathbf{z}_{j,t}. \end{aligned}$$
      iii. Transfer $\mathbf{R}_{j,t}^{(m)}$, $\boldsymbol{\gamma}_{j,t}^{(m)}$, $a_{j,t}^{(m)}$ and back to the central node.
   (b) At the central node, do the following in parallel for $m = 1, \ldots, M$:
      i. Based on $\boldsymbol{\theta}^{(m)}$, calculate $\boldsymbol{\nu}_{t|t-1}^{(m)} := \mathbf{H}_t^{(m)}\boldsymbol{\nu}_{t-1|t-1}^{(m)}$, $\mathbf{K}_{t|t-1}^{(m)} := \mathbf{H}_t^{(m)}\mathbf{K}_{t-1|t-1}^{(m)}\mathbf{H}_t^{(m)\prime} + \mathbf{U}_t^{(m)}$, $(\mathbf{K}_{t|t}^{(m)})^{-1} = (\mathbf{K}_{t|t-1}^{(m)})^{-1} + \sum_{j=1}^{J}\mathbf{R}_{j,t}^{(m)}$, $\boldsymbol{\nu}_{t|t}^{(m)} = \mathbf{K}_{t|t}^{(m)}((\mathbf{K}_{t|t-1}^{(m)})^{-1}\boldsymbol{\nu}_{t|t-1}^{(m)} + \sum_{j=1}^{J}\boldsymbol{\gamma}_{j,t}^{(m)})$, and the filtering likelihood $L_t(\boldsymbol{\theta}_t^{(m)})$ as in (8).
   (c) The particle-filter approximation of the filtering distribution of $\boldsymbol{\theta}_t$ takes on the value $\boldsymbol{\theta}_t^{(m)}$ with probability $w_t^{(m)} \propto p(\boldsymbol{\theta}_t^{(m)}|\boldsymbol{\theta}_{t-1}^{(m)})L_t(\boldsymbol{\theta}_t^{(m)})/q(\boldsymbol{\theta}_t^{(m)}|\boldsymbol{\theta}_{t-1}^{(m)})$.
   (d) Using a resampling scheme (see, e.g., Douc et al. 2005), generate resampled particles $\widetilde{\boldsymbol{\theta}}_t^{(1)}, \ldots, \widetilde{\boldsymbol{\theta}}_t^{(M)}$ (and the associated $\mathbf{K}_{t|t}^{(m)}$ and $\boldsymbol{\nu}_{t|t}^{(m)}$) from $\boldsymbol{\theta}_t^{(1)}, \ldots, \boldsymbol{\theta}_t^{(M)}$, and obtain M particles for time $t+1$ using a suitable proposal distribution $q(\boldsymbol{\theta}_{t+1}|\widetilde{\boldsymbol{\theta}}_t^{(m)})$.

In a smoothing context, parameter inference is based on the likelihood, $[\mathbf{z}_{1:T}|\boldsymbol{\theta}_{1:T}] = \prod_{t=1}^{T}[\mathbf{z}_t|\mathbf{z}_{1:t-1}, \boldsymbol{\theta}_t]$, of all data in a specific time window $\{1, \ldots, T\}$, where $[\mathbf{z}_t|\mathbf{z}_{1:t-1}, \boldsymbol{\theta}_t]$ is given in (8).

7 Application: total precipitable water measured by three sensor systems

We applied our methodology to hourly measurements from three sensor systems to obtain spatio-temporal filtering inference on an atmospheric variable called total precipitable water. Total precipitable water is the integrated amount of water vapor in a column from the surface of the earth to space in kilograms per square meter or, equivalently, in millimeters of condensate. The sensor systems are ground-based GPS, the Geostationary Operational Environmental Satellite (GOES) infrared sounders, and Microwave Integrated Retrieval System (MIRS) satellites. These data products are retrieved and stored at different data centers, and so our Situation 1 described in Sect. 1 applies. The sensor systems also feature varying spatial coverage and precision. The measurement-error standard deviations are 0.75, 2, and 4.5 mm, respectively, and so the function $v_\epsilon(\cdot)$ from (1) varies by server (i.e., by j) but not over space.

Since March 2009, an operational blended multisensor water vapor product based on these three sensor systems has been produced by the National Environmental Satellite, Data, and Information Service of NOAA (Kidder and Jones 2007; Forsythe et al. 2012). This product is sent to National Weather Service offices, where it is used by forecasters to track the movement of water vapor in the atmosphere and to detect antecedent conditions for heavy precipitation. The operational product is created by overlaying the existing field with the latest available data, which can lead to unphysical features in the form of abrupt boundaries. The goal of our analysis was to illustrate our methodology using a simple version of a spatio-temporal low-rank model, and to create spatially more coherent predictive maps with associated uncertainties based on data from all three systems, without having to transfer the data to a central processor.

We consider here a dataset consisting of a total of 3,351,860 measurements assumed to be collected at point-level support in January 2011 over a period of 47 h by the three sensor systems over a spatial domain covering the United States. The top three rows of Fig. 2 show the three sensor data products at time points (hours) 7, 8 and 9. As is evident from these plots, total precipitable water exhibits considerable variability at the considered spatial and temporal scales.

We made filtering inference based on a spatio-temporal low-rank model, parameterized by a predictive-process approach inspired by Finley et al. (2012). Specifically, we assumed the model in Sect. 6 with $v_{\delta,t}(\cdot) \equiv \sigma^2_{\delta,t}$, $\mathbf{H}_t = \alpha_t \mathbf{I}_r$,

$$\mathbf{K}_{0,0}^{-1} = (\rho(\mathbf{w}_i, \mathbf{w}_j|\boldsymbol{\theta}_0))_{i,j=1,\ldots,r}$$
$$\mathbf{U}_t^{-1} = (1-\alpha_t^2)^{-1}(\rho(\mathbf{w}_i, \mathbf{w}_j|\boldsymbol{\theta}_t))_{i,j=1,\ldots,r}$$
$$\mathbf{b}_t(\mathbf{s}) = \sigma_t(\mathbf{s})\left(\rho(\mathbf{s}, \mathbf{w}_1|\boldsymbol{\theta}_t), \ldots, \rho(\mathbf{s}, \mathbf{w}_r|\boldsymbol{\theta}_t)\right)', \ \mathbf{s} \in \mathcal{D}.$$

The parent correlation function was chosen to be

$$\rho(\mathbf{s}_1, \mathbf{s}_2|\boldsymbol{\theta}_t) = \mathcal{M}(\|\mathbf{s}_1 - \mathbf{s}_2\|/\kappa_t) \cdot \mathcal{T}(\|\mathbf{s}_1 - \mathbf{s}_2\|/10),$$

where $\mathcal{M}$ is the Matérn correlation function (e.g., Stein 1999, p. 50) with smoothness $\upsilon = 1.25$,

$$\mathcal{M}(h) = (2h\sqrt{\upsilon})^{\upsilon}\mathcal{K}_{\upsilon}(2h\sqrt{\upsilon})2^{1-\upsilon}/\Gamma(\upsilon)$$

and multiplication by the compactly supported Kanter's function $\mathcal{T}$ (Kanter 1997) led to considerable sparsity in the matrices $\mathbf{R}_{j,t}$, as described in Sect. 3.1. The set of knots, $\mathcal{W}$:=$\{\mathbf{w}_1, \ldots, \mathbf{w}_{84}\}$, was a regular $5^\circ \times 5^\circ$ latitude/longitude grid over the domain. The trend consisted of an intercept term with a Gaussian random-walk prior with initial value 13.2 and variance 15.9 and was absorbed into the basis-function vector. While we chose this relatively simple model here for illustration, we would like to reiterate that neither the communication cost nor the computational complexity of the algorithm changes if a more elaborate parameterization of the general spatio-temporal low-rank model in Sect. 6 is chosen.

The transition distribution of the parameter vector

$$\boldsymbol{\theta}_t = (\Phi^{-1}(\alpha_t), \quad \log(\sigma_t), \quad \log(\kappa_t), \quad \log(\sigma^2_{\delta,t}))'$$

was taken to be a Gaussian random walk with $\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1} \sim N_4(\boldsymbol{\theta}_{t-1}, 0.01 \times \mathbf{I}_4)$, for $t = 1, \ldots, T = 47$. The initial parameter vector $\boldsymbol{\theta}_0$ was specified as $\alpha_0 = 0.8$, $\sigma_0 = 5$, $\kappa_0 = 15$, and $\sigma^2_{\delta,0} = 0.5$. Here, α_t determines the strength of the temporal dependence, while the scale parameter κ_t determines the strength of the spatial dependence.

We implemented the sequential importance sampling algorithm with residual resampling as described in Algorithm 4 with $M = 6000$ particles, using the prior distribution as the proposal distribution for simplicity. The resulting filtering posterior means and posterior standard deviations for total precipitable water for time periods 7, 8, and 9 (i.e., $t \in \{7, 8, 9\}$) on a regular $0.5^\circ \times 0.5^\circ$ latitude/longitude grid of size 6,283 are shown in the bottom two rows of Fig. 2. We were able to calculate the filtering distribution based on the 3,351,860 measurements collected over 47 hours by the three sensor systems in about 7 hours using parallel computations on the Geyser data analysis cluster on the high-performance computing facility Yellowstone at the National Center for Atmospheric Research. Geyser uses 10-core 2.4-GHz Intel

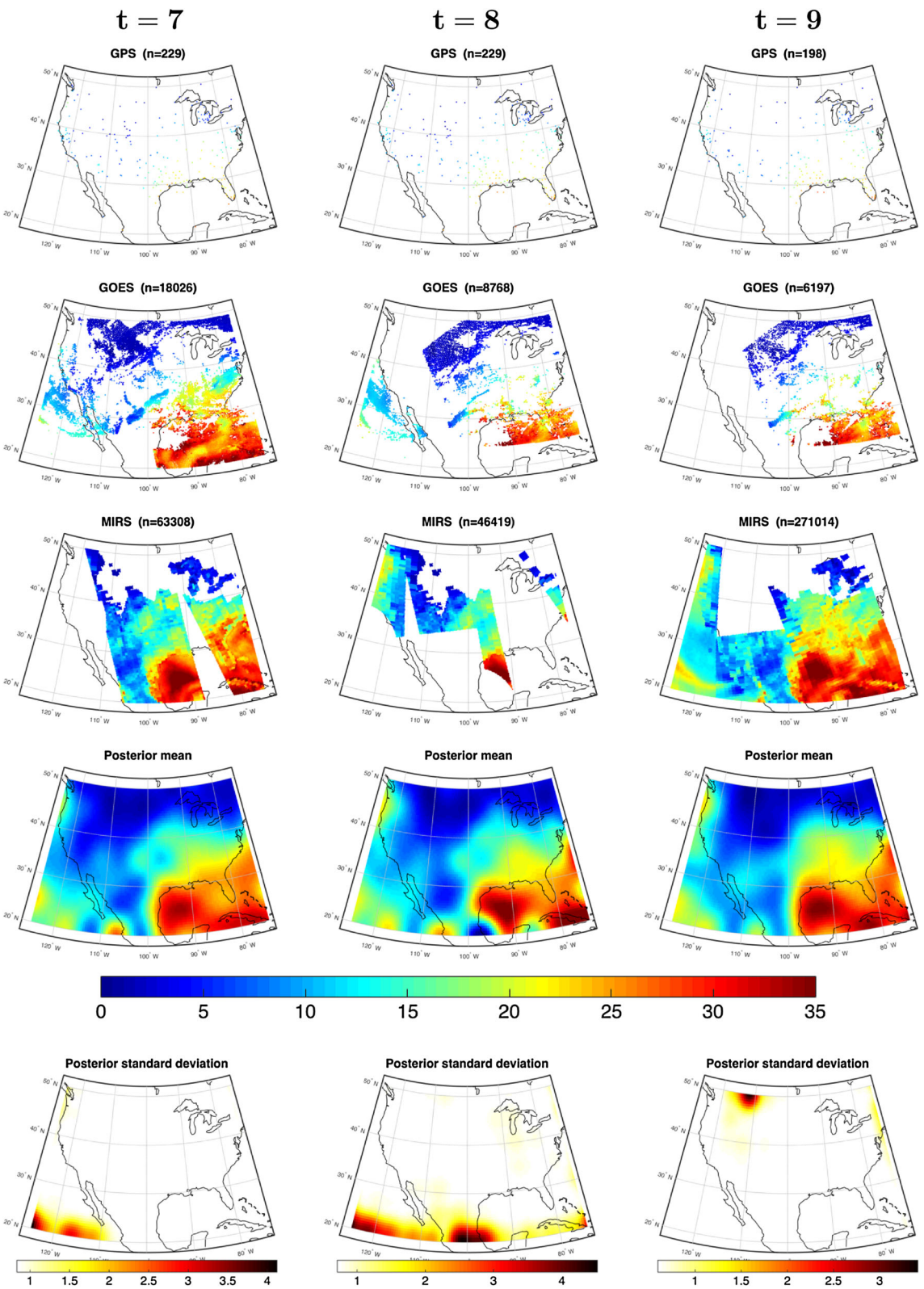

Fig. 2 *Top three rows* Hourly observations of total precipitable water by the GPS system, GOES infrared sounders, and MIRS, respectively, over the larger continental United States in January 2011. *Bottom two rows* Corresponding filtering posterior means and posterior standard deviations, respectively, of total precipitable water based on all three data products. The *columns* represent time points 7, 8, and 9, respectively. The scale for the posterior standard deviation plots varies between time periods. All units are in millimeters

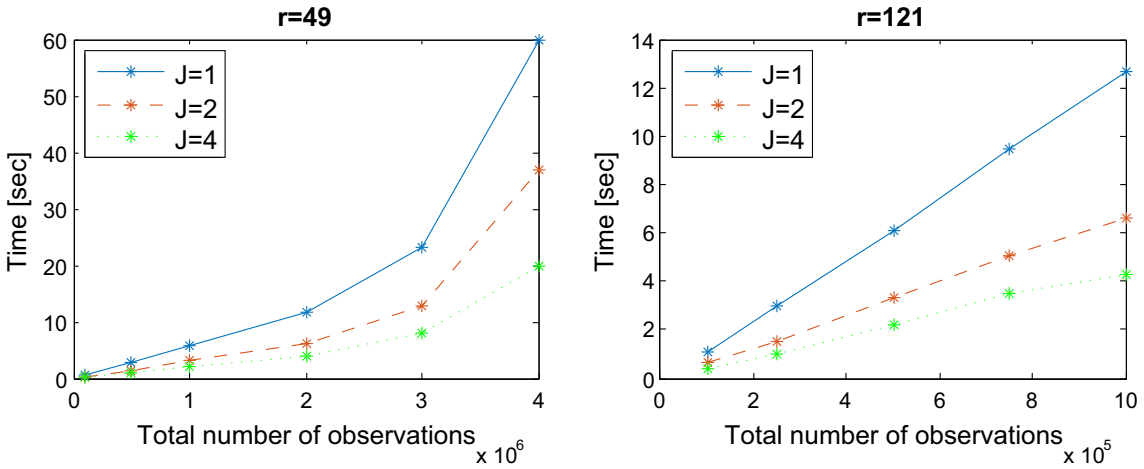

Fig. 3 Computation time for one likelihood evaluation for simulated spatial data with a varying number of observations and different numbers of computational nodes (J), for $r = 49$ (*left*) and $r = 121$ (*right*) knots

Xeon E7-4870 (Westmere EX) processors and has 25GB of memory/core.

8 Timing study

While the focus of this article is on avoiding data movement and duplicate storage for distributed data (Situation 1 from Sect. 1), the outlined methodology is also applicable without modification to a divide-and-conquer inference scheme in the case of centrally stored data (Situation 2). We briefly investigated the benefits in terms of computational speed by parallelizing one spatial-only likelihood evaluation for a predictive-process model with a Matérn covariance function similar to the one in Sect. 7, for $r = 49$ and $r = 121$ knots, and varying numbers of simulated observations and numbers of servers. The timing results shown in Fig. 3 are the means of ten replicates (the variation between replicates is very small). The study was conducted on a MacBook Pro with an Intel quad-core 2.6 GHz i7 processor and 8 GB of memory. The specific results are dependent on the characteristics of the processor, but provide a relative sense of computational improvement potential. We see that parallelizing over several processors leads to speed-ups, as expected. We would like to emphasize that, while not investigated further here, the divide-and-conquer scheme made possible by our methodology can also lead to memory advantages by splitting up the analysis in a distributed-memory environment. This is crucial when, for example, analyzing sea-surface temperature with hundreds of millions of measurements per day.

9 Conclusions and future work

As datasets are becoming larger, so is the cost of moving them to a central computer for analysis, necessitating algorithms designed to work on distributed data that keep analysis operations as close to the stored data as possible. We showed how distributed spatial inference, including likelihood-based parameter inference, can be carried out in a computationally feasible way for massive distributed datasets under the assumption of a low-rank model, while producing the same results as traditional, non-distributed inference. Our approach is scalable in that the computational cost is linear in each n_j (the number of measurements at server j) and the communication cost does not depend on the n_j at all. Inference, especially when done based on a particle sampler, is also "embarassingly parallel," allowing a divide-and-conquer analysis of massive spatial data with little communication overhead. In addition, if the selected low-rank model has fixed basis functions that do not depend on parameters (see Sect. 4.2), our methodology can be used for data reduction in situations where it is not possible to store all measurements.

After extending the results to the spatio-temporal case, we demonstrated the applicability of our model to massive real-world data in Sect. 7, and showed that we can obtain sensible results in a fast manner. However, getting the best possible results for this particular application is part of ongoing research and will likely require a more refined and complicated model.

The methodology described in this article can be extended to the full-scale approximation of Sang et al. (2011), where the fine-scale variation is assumed to be dependent within subregions of the spatial domain, resulting in nondiagonal $\mathbf{V}_j$. This idea is explored for a multi-resolutional extension of the full-scale approximation in Katzfuss (2015).

Another natural extension of our methodology is to the increasingly important multivariate data-fusion case involving inference on multiple processes based on data from multiple measuring instruments. Multivariate analysis can in principle be carried out as described here by stacking

the basis function weights for the individual processes into one big vector $\boldsymbol{\eta}$ (see,e.g., Nguyen et al. 2012, 2014), but it will likely require more complicated inference on $\delta(\cdot)$ due to different instrument footprints and overlaps. While the combined size of the low-rank components for multiple processes will become prohibitive in highly multivariate settings, the hope is that the processes can be written as linear combinations of a smaller number of processes.

Acknowledgements This material was based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Katzfuss was partially supported by NASA's Earth Science Technology Office AIST-14 program and by National Science Foundation (NSF) Grant DMS-1521676. Hammerling's research also had partial support from the NSF Research Network on Statistics in the Atmosphere and Ocean Sciences (STATMOS) through Grant DMS-1106862. We would like to acknowledge high-performance computing support from Yellowstone (ark:/85065/d7wd3xhc) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation. We would like to thank Amy Braverman for making us aware of the problem of distributed spatial data; John Forsythe and Stan Kidder for the datasets and helpful advice; Yoichi Shiga for support with preprocessing and visualizing the data; and Andrew Zammit Mangion, Emtiyaz Khan, Kirk Borne, Jessica Matthews, Emily Kang, several anonymous reviewers, and the SAMSI Massive Datasets Environment and Climate working group for helpful comments and discussions.

Appendix 1: Derivation of the likelihood

We derive here the expression of the likelihood in (5). First, note that $\mathbf{z}_{1:J}|\boldsymbol{\theta} \sim N_n(\mathbf{B}_{1:J}\boldsymbol{\nu}_0, \boldsymbol{\Sigma}_{1:J})$, where $\boldsymbol{\Sigma}_{1:J} = \mathbf{B}_{1:J}\mathbf{K}_0\mathbf{B}'_{1:J} + \mathbf{V}_{1:J}$. Hence, the likelihood is given by,

$$\begin{aligned}-2\log[\mathbf{z}_{1:J}|\boldsymbol{\theta}] &= \log|\boldsymbol{\Sigma}_{1:J}| \\ &\quad + (\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0)'\boldsymbol{\Sigma}_{1:J}^{-1}(\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0) \\ &\quad - (n/2)\log(2\pi).\end{aligned}$$

Applying a matrix determinant lemma (e.g., Harville 1997 Thm. 18.1.1), we can write the log determinant as,

$$\begin{aligned}\log|\boldsymbol{\Sigma}_{1:J}| &= \log|\mathbf{V}_{1:J}| + \log|\mathbf{K}_0| \\ &\quad + \log|\mathbf{B}'_{1:J}\mathbf{V}_{1:J}^{-1}\mathbf{B}_{1:J} + \mathbf{K}_0^{-1}| \\ &= \textstyle\sum_{j=1}^J \log|\mathbf{V}_j| - \log|\mathbf{K}_0^{-1}| + \log|\mathbf{K}_z^{-1}|.\end{aligned}$$

Further, using the Sherman-Morrison-Woodbury formula (Sherman and Morrison 1950; Woodbury 1950; Henderson and Searle 1981), we can show that $\boldsymbol{\Sigma}_{1:J}^{-1} = \mathbf{V}_{1:J}^{-1} - \mathbf{V}_{1:J}^{-1}\mathbf{B}_{1:J}\mathbf{K}_z\mathbf{B}'_{1:J}\mathbf{V}_{1:J}^{-1}$, and so

$$\begin{aligned}&(\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0)'\boldsymbol{\Sigma}_{1:J}^{-1}(\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0) \\ &\quad = \textstyle\sum_{j=1}^J (\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0)'\mathbf{V}_j^{-1}(\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0) \\ &\qquad - \big(\sum_{j=1}^J \mathbf{B}'_j\mathbf{V}_j^{-1}(\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0)\big)'\mathbf{K}_z \\ &\qquad \times \big(\textstyle\sum_{j=1}^J \mathbf{B}'_j\mathbf{V}_j^{-1}(\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0)\big) \\ &\quad = \sum_j \mathbf{z}'_j\mathbf{V}_j^{-1}\mathbf{z}_j - 2\boldsymbol{\nu}'_0(\mathbf{K}_z^{-1}\boldsymbol{\nu}_z - \mathbf{K}_0^{-1}\boldsymbol{\nu}_0) \\ &\qquad + \boldsymbol{\nu}'_0(\mathbf{K}_z^{-1} - \mathbf{K}_0^{-1})\boldsymbol{\nu}_0 \\ &\qquad - \big((\mathbf{K}_z^{-1}\boldsymbol{\nu}_z - \mathbf{K}_0^{-1}\boldsymbol{\nu}_0) - (\mathbf{K}_z^{-1} - \mathbf{K}_0^{-1})\boldsymbol{\nu}_0\big)'\mathbf{K}_z \\ &\qquad \times \big((\mathbf{K}_z^{-1}\boldsymbol{\nu}_z - \mathbf{K}_0^{-1}\boldsymbol{\nu}_0) - (\mathbf{K}_z^{-1} - \mathbf{K}_0^{-1})\boldsymbol{\nu}_0\big) \\ &\quad = \sum_j \mathbf{z}'_j\mathbf{V}_j^{-1}\mathbf{z}_j - 2\boldsymbol{\nu}'_0\mathbf{K}_z^{-1}\boldsymbol{\nu}_z + \boldsymbol{\nu}'_0\mathbf{K}_0^{-1}\boldsymbol{\nu}_0 + \boldsymbol{\nu}'_0\mathbf{K}_z^{-1}\boldsymbol{\nu}_0 \\ &\qquad - (\mathbf{K}_z^{-1}\boldsymbol{\nu}_z)'\mathbf{K}_z(\mathbf{K}_z^{-1}\boldsymbol{\nu}_z) - \boldsymbol{\nu}'_0\mathbf{K}_z^{-1}\mathbf{K}_z\mathbf{K}_z^{-1}\boldsymbol{\nu}_0 \\ &\qquad + 2(\mathbf{K}_z^{-1}\boldsymbol{\nu}_z)'\mathbf{K}_z\mathbf{K}_z^{-1}\boldsymbol{\nu}_0 \\ &\quad = \textstyle\sum_j \mathbf{z}'_j\mathbf{V}_j^{-1}\mathbf{z}_j + \boldsymbol{\nu}'_0\mathbf{K}_0^{-1}\boldsymbol{\nu}_0 - \boldsymbol{\nu}'_z\mathbf{K}_z^{-1}\boldsymbol{\nu}_z,\end{aligned}$$

where $\sum_{j=1}^J \mathbf{B}'_j\mathbf{V}_j^{-1}\mathbf{B}_j = \mathbf{K}_z^{-1} - \mathbf{K}_0^{-1}$ and $\sum_{j=1}^J \mathbf{B}'_j\mathbf{V}_j^{-1}\mathbf{z}_j = \mathbf{K}_z^{-1}\boldsymbol{\nu}_z - \mathbf{K}_0^{-1}\boldsymbol{\nu}_0$ both follow from (3).

Appendix 2: Spatial prediction when observed and prediction locations coincide

Here we describe how to do spatial prediction when a small number, q say, of the observed locations are also in the set of desired prediction locations. Define $\boldsymbol{\delta}_{P,O}$ to be the vector of the first q elements of $\boldsymbol{\delta}^P$, which we assume to correspond to the q observed prediction locations, and let $\mathbf{P}_j$ be a sparse $n_j \times q$ matrix with $(\mathbf{P}_j)_{k,l} = I(\mathbf{s}_{j,k} = \mathbf{s}_l^P)$. We write our model in state-space form with identity evolution equation, $\mathbf{z}_j = \tilde{\mathbf{B}}_j\tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\xi}}_j$, where $\tilde{\mathbf{B}}_j{:=}(\mathbf{B}_j, \mathbf{P}_j)$, $\tilde{\boldsymbol{\eta}}{:=}(\boldsymbol{\eta}', \boldsymbol{\delta}'_{P,O})' \sim N(\tilde{\boldsymbol{\nu}}_0, \tilde{\mathbf{K}}_0)$, $\tilde{\boldsymbol{\nu}}_0{:=}(\boldsymbol{\nu}'_0, \mathbf{0}'_q)'$, $\tilde{\mathbf{K}}_0$ is blockdiagonal with first block $\mathbf{K}_0$ and second block $diag\{v_\delta(\mathbf{s}_1^P), \ldots, v_\delta(\mathbf{s}_q^P)\}$, $\tilde{\boldsymbol{\xi}}_j \sim N_{n_j}(\mathbf{0}, \tilde{\mathbf{V}}_j)$, and $\tilde{\mathbf{V}}_j$ is the same as $\mathbf{V}_j$ except that the ith diagonal element is now $v_\epsilon(\mathbf{s}_{j,i})$ if $\mathbf{s}_{j,i}$ is one of the prediction locations.

The decentralized Kalman filter (Rao et al. 1993) gives $\tilde{\mathbf{K}}_z^{-1} = \tilde{\mathbf{K}}_0^{-1} + \sum_{j=1}^J \tilde{\mathbf{R}}_j$ and $\tilde{\boldsymbol{\nu}}_z = \tilde{\mathbf{K}}_z(\tilde{\mathbf{K}}_0^{-1}\tilde{\boldsymbol{\nu}}_0 + \sum_{j=1}^J \tilde{\boldsymbol{\gamma}}_j)$, where $\tilde{\mathbf{R}}_j{:=}\tilde{\mathbf{B}}'_j\tilde{\mathbf{V}}_j^{-1}\tilde{\mathbf{B}}_j$ and $\tilde{\boldsymbol{\gamma}}_j{:=}\tilde{\mathbf{B}}'_j\tilde{\mathbf{V}}_j^{-1}\mathbf{z}_j$ are the only quantities that need to be calculated at and transfered from server j, which is feasible due to sparsity if q is not too large. The predictive distribution is then given by $\mathbf{y}^P|\mathbf{z}_{1:J} \sim N(\tilde{\mathbf{B}}^P\tilde{\boldsymbol{\nu}}_J, \tilde{\mathbf{B}}^P\tilde{\mathbf{K}}_z\tilde{\mathbf{B}}^{P\prime} + \tilde{\mathbf{V}}_\delta^P)$, where $\tilde{\mathbf{B}}^P{:=}(\mathbf{B}^P, (\mathbf{I}_q, \mathbf{0})')$ and $\tilde{\mathbf{V}}_\delta^P{:=}diag\{\mathbf{0}'_q, v_\delta(\mathbf{s}_{q+1}^P), \ldots, v_\delta(\mathbf{s}_{n_P}^P)\}$.

References

Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H.: Gaussian predictive process models for large spatial data sets. J. R. Stat. Soc. Ser. B **70**(4), 825–848 (2008). doi:10.1111/j.1467-9868.2008.00663.x

Bevilacqua, M., Gaetan, C., Mateu, J., Porcu, E.: Estimating space and space-time covariance functions for large data sets: a weighted composite likelihood approach. J. Am. Stat. Assoc. **107**(497), 268–280 (2012). doi:10.1080/01621459.2011.646928

Bradley, J.R., Cressie, N., Shi, T.: A comparison of spatial predictors when datasets could be very large. (2014) arXiv:1410.7748

Calder, C.A.: Dynamic factor process convolution models for multivariate space-time data with application to air quality assessment. Environ. Ecol. Stat. **14**(3), 229–247 (2007). doi:10.1007/s10651-007-0019-y

Caragea, P.C., Smith, R.L.: Asymptotic properties of computationally efficient alternative estimators for a class of multivariate normal models. J. Multivar. Anal. **98**(7), 1417–1440 (2007). doi:10.1016/j.jmva.2006.08.010

Caragea, P.C., Smith, R.L.: Approximate likelihoods for spatial processes. Technical Report, University of North Carolina, Chapel Hill, NC (2008)

Cortés, J.: Distributed kriged Kalman filter for spatial estimation. IEEE Trans. Autom. Control **54**(12), 2816–2827 (2009)

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

Cressie, N., Shi, T., Kang, E.L.: Fixed rank filtering for spatio-temporal data. J Comput. Graph. Stat. **19**(3), 724–745 (2010)

Curriero, F., Lele, S.: A composite likelihood approach to semivariogram estimation. J. Agric. Biol. Environ. Stat. **4**(1), 9–28 (1999)

Douc R, Cappé O, Moulines E (2005) Comparison of resampling schemes for particle filtering. In: ISPA 2005 Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005 pp 64–69. doi:10.1109/ISPA.2005.195385

Eidsvik, J., Shaby, B.A., Reich, B.J., Wheeler, M., Niemi, J.: Estimation and prediction in spatial models with block composite likelihoods using parallel computing. J. Comput. Graph. Stat. **23**(2), 295–315 (2014)

Finley, A., Banerjee, S., Gelfand, A.E.: Bayesian dynamic modeling for large space-time datasets using Gaussian predictive processes. J. Geogr. Syst. **14**, 29–47 (2012)

Finley, A.O., Sang, H., Banerjee, S., Gelfand, A.E.: Improving the performance of predictive process modeling for large datasets. Comput. Stat. Data Anal. **53**(8), 2873–2884 (2009). doi:10.1016/j.csda.2008.09.008

Forsythe, J.M., Dodson, J.B., Partain, P.T., Kidder, S.Q., Haar, T.H.V.: How total precipitable water vapor anomalies relate to cloud vertical structure. J. Hydrometeorol. **13**(2), 709–721 (2012)

Fuller, S.H., Millett, L.I. (eds.): The Future of Computing Performance: Game Over or Next Level?. Committee on Sustaining Growth in Computing Performance; National Research Council, Washington, DC (2011)

Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc. Radar Signal Process. **140**(2), 107–113 (1993)

Graham, R., Cortés, J.: Cooperative adaptive sampling of random fields with partially known covariance. Int. J. Robust Nonlinear Control **22**(5), 504–534 (2012)

Harville, D.A.: Matrix Algebra from a Statistician's Perspective. Springer, New York (1997)

Henderson, H., Searle, S.: On deriving the inverse of a sum of matrices. SIAM Rev. **23**(1), 53–60 (1981)

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

Kang, E.L., Cressie, N.: Bayesian inference for the spatial random effects model. J. Am. Stat. Assoc. **106**(495), 972–983 (2011)

Kang, E.L., Liu, D., Cressie, N.: Statistical analysis of small-area data based on independence, spatial, non-hierarchical, and hierarchical models. Comput. Stat. Data Anal. **53**(8), 3016–3032 (2009). doi:10.1016/j.csda.2008.07.033

Kanter, M.: Unimodal spectral windows. Stat. Probab. Lett. **34**(4), 403–411 (1997). http://linkinghub.elsevier.com/retrieve/pii/S0167715296002088

Katzfuss, M.: Bayesian nonstationary spatial modeling for very large datasets. Environmetrics **24**(3), 189–200 (2013)

Katzfuss, M.: A multi-resolution approximation for massive spatial datasets. J. Am. Stat. Assoc. (2015). doi:10.1080/01621459.2015.1123632

Katzfuss, M., Cressie, N.: Maximum likelihood estimation of covariance parameters in the spatial-random-effects model. In: Proceedings of the Joint Statistical Meetings, American Statistical Association, Alexandria, VA, pp 3378–3390 (2009)

Katzfuss, M., Cressie, N.: Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets. J. Time Ser. Anal. **32**(4), 430–446 (2011). doi:10.1111/j.1467-9892.2011.00732.x

Katzfuss, M., Cressie, N.: Bayesian hierarchical spatio-temporal smoothing for very large datasets. Environmetrics **23**(1), 94–107 (2012)

Kidder, S.Q., Jones, A.S.: A blended satellite total precipitable water product for operational forecasting. J. Atmos. Ocean. Technol. **24**(1), 74–81 (2007)

Lemos, R.T., Sansó, B.: A spatio-temporal model for mean, anomaly, and trend fields of North Atlantic sea surface temperature. J. Am. Stat. Assoc. **104**(485), 5–18 (2009). doi:10.1198/jasa.2009.0018

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

Mardia, K., Goodall, C., Redfern, E., Alonso, F.: The kriged Kalman filter. Test **7**(2), 217–282 (1998)

Nguyen, H., Cressie, N., Braverman, A.: Spatial statistical data fusion for remote sensing applications. J. Am. Stat. Assoc. **107**(499), 1004–1018 (2012)

Nguyen, H., Katzfuss, M., Cressie, N., Braverman, A.: Spatio-temporal data fusion for very large remote sensing datasets. Technometrics **56**(2), 174–185 (2014)

Nychka, D.W., Bandyopadhyay, S., Hammerling, D., Lindgren, F., Sain, S.R.: A multi-resolution Gaussian process model for the analysis of large spatial data sets. J. Comput. Graph. Stat. **24**(2), 579–599 (2015)

Rao, B., Durrant-Whyte, H., Sheen, J.: A fully decentralized multi-sensor system for tracking and surveillance. Int. J. Robot. Res. **12**(1), 20–44 (1993). doi:10.1177/027836499301200102

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

Sherman, J., Morrison, W.: Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. Ann. Math. Stat. **21**(1), 124–127 (1950)

Shi, T., Cressie, N.: Global statistical analysis of MISR aerosol data: a massive data product from NASA's Terra satellite. Environmetrics **18**, 665–680 (2007)

Shoshani, A., Klasky, S., Ross, R.: Scientific data management: Challenges and approaches in the extreme scale era. In: Proceedings of the 2010 Scientific Discovery through Advanced Computing (SciDAC) Conference, Chattanooga, TN, 1, pp 353–366 (2010)

Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging. Springer, New York (1999)

Stein, M.L.: Limitations on low rank approximations for covariance matrices of spatial data. Spat. Stat. **8**, 1–19 (2014). doi:10.1016/j.spasta.2013.06.003

Stein, M.L., Chi, Z., Welty, L.: Approximating likelihoods for large spatial data sets. J. Roy. Stat. Soc. B **66**(2), 275–296 (2004)

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

Wikle, C.K., Cressie, N.: A dimension-reduced approach to space-time Kalman filtering. Biometrika **86**(4), 815–829 (1999)
Woodbury, M.: Inverting modified matrices. Memorandum Report 42, Statistical Research Group, Princeton University (1950)
Xu, B., Wikle, C.K., Fox, N.: A kernel-based spatio-temporal dynamical model for nowcasting radar precipitation. J. Am. Stat. Assoc. **100**(472), 1133–1144 (2005)
Xu, Y., Choi, J.: Adaptive sampling for learning gaussian processes using mobile sensor networks. Sensors **11**(3), 3051–3066 (2011). doi:10.3390/s110303051
Zhang, K.: ISSCC 2013: Memory trends. http://www.electroiq.com/articles/sst/2013/02/isscc-2013-memory-trends.html (2013). Accessed 12 June 2013