Stochastic Local Interaction (SLI) Model: Interfacing Machine Learning and Geostatistics

Dionissios T. Hristopulos[*]

School of Mineral Resources Engineering,

Technical University of Crete, Chania 73100, Greece

(Dated: October 15, 2018)

Abstract

Machine learning and geostatistics are powerful mathematical frameworks for modeling spatial data. Both approaches, however, suffer from poor scaling of the required computational resources for large data applications. We present the Stochastic Local Interaction (SLI) model, which employs a local representation to improve computational efficiency. SLI combines geostatistics and machine learning with ideas from statistical physics and computational geometry. It is based on a joint probability density function defined by an energy functional which involves local interactions implemented by means of kernel functions with adaptive local kernel bandwidths. SLI is expressed in terms of an explicit, typically sparse, precision (inverse covariance) matrix. This representation leads to a semi-analytical expression for interpolation (prediction), which is valid in any number of dimensions and avoids the computationally costly covariance matrix inversion.

[*] dionisi@mred.tuc.gr

I. INTRODUCTION

Big data is expected to have a large impact in the geosciences given the abundance of remote sensing and earth-based observations related to climate [1, 34]. A similar data explosion is happening in other scientific and engineering fields [44]. This trend underscores the need for algorithms that can handle large data sets. Most current methods of data analysis, however, have not been designed with size as a primary consideration. This has inspired statements such as: "Improvements in data-processing capabilities are essential to make maximal use of state-of-the-art experimental facilities" [3]. Machine learning can extract information and "learn" characteristic patterns in the data. Thus, it is expected to play a significant role in the era of big data research. The application of machine learning methods in spatial data analysis has been spearheaded by Kanevski [25]. Machine learning and geostatistics are powerful frameworks for spatial data processing. A comparison of their performance using a set of radiological measurements is presented in [19]. The question that we address in this work is whether we can combine ideas from both fields to develop a computationally efficient framework for spatial data modeling.

Most data processing and visualization methods assume complete data sets, whereas in practice data often have gaps. Hence, it is necessary to fill missing values by means of imputation or interpolation methods. In geostatistics, such methods are based on various flavors of stochastic optimal linear estimation (kriging) [7]. In machine learning, methods such as k-nearest neighbors, artificial neural networks, and the Bayesian framework of Gaussian processes are used [31]. Both geostatistics and Gaussian process regression are based on the theory of random fields and share considerable similarities [2, 45]. The Gaussian process framework, however, is better suited for applications in higher than two dimensions. A significant drawback of most existing methods for interpolation and simulation of missing data is their poor scalability with the data size N, i.e., the $O(N^3)$ algorithmic complexity and the $O(N^2)$ memory requirements: An $O(N^p)$ dependence implies that the respective computational resource (time or memory) increases with N as a polynomial of degree at most equal to p.

Improved scaling with data size can be achieved by means of local approximations, dimensionality reduction techniques, and parallel algorithms. A recent review of available methods for large data geostatistical applications is given in [35]. These approaches employ clever

approximations to reduce the computational complexity of the standard geostatistical framework. Local approximations involve methods such as maximum composite likelihood [40] and maximum pseudo-likelihood [38]. Another approach involves covariance tapering which neglects correlations outside a specified range [10, 16, 26]. Dimensionality reduction includes methods such as fixed rank kriging which models the precision matrix by means of a fixed rank matrix $r \ll N$ [8, 30]. Markov random fields (MRFs) also take advantage of locality using factorizable joint densities. The application of MRFs in spatial data analysis was initially limited to structured grids [32]. However, a recently developed link between Gaussian random fields and MRFs via stochastic partial differential equations (SPDE) has extended the scope of MRFs to scattered data [28].

We propose a Stochastic Local Interaction (SLI) model for spatially correlated data which is based by construction on local correlations. SLI can be used for the interpolation and simulation of incomplete data in d-dimensional spaces, where d could be larger than 3. The SLI model incorporates concepts from statistical physics, computational geometry, and machine learning. We use the idea of local interactions from statistical physics to impose correlations between "neighboring" locations by means of an explicit precision matrix. The local geometry of the sampling network plays an important role in the expression of the interactions, since it determines the size of local neighborhoods. On regular grids, the SLI model becomes equivalent to a Gaussian MRF with specific structure. For scattered data, the SLI model provides an alternative to the SPDE approach that avoids the preprocessing cost involved in the latter.

The SLI model extends previous research on Spartan spatial random fields [13, 21, 22] to an explicitly discrete formulation and thus enables its application to scattered data without the approximations used in [23]. SLI is based on a joint probability density function (pdf) determined from local interactions. This is achieved by handling the irregularity of sampling locations in terms of kernel functions with locally adaptive bandwidth. Kernel methods are common in statistical machine learning [39] and in spatial statistics for the estimation of the variogram and the covariance function [14, 17, 20].

The remainder of the article is structured as follows: Section II briefly introduces useful definitions and terminology. In Section III we construct the SLI model, propose a computationally efficient parameter estimation approach, and formulate an explicit interpolation expression. In Section IV we investigate SLI interpolation using different types of simulated

and real data in one, two and four dimensional Euclidean spaces. Section V discusses potential extensions of the current SLI version and connections with machine learning. Finally, in Section VI we present our conclusions and point to future research.

II. BACKGROUND CONCEPTS AND NOTATION

A. Definition of the problem to be learned

a. Sampling grid The set of sampling points is denoted by $S_{\mathrm{N}} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$, where $\mathbf{s}_i$, $i = 1, \ldots, N$ are vectors in the Euclidean space $\mathbb{R}^d$ or in some abstract feature space that possesses a distance metric. In Euclidean spaces, the domain boundary is defined by the convex hull, $\mathcal{H}(S_N)$, of S_{N}.

b. Sample and predictions The sample data are denoted by the vector $\mathbf{x}_{\mathrm{S}} \equiv (x_1, \ldots, x_N)^T$, where the superscript "T" denotes the transpose. Interpolation aims to derive estimates of the observed field at the nodes of a regular grid $\mathcal{G} \subset \mathbb{Z}^d$, or at validation set points which may be scattered. The estimates (predictions) will be denoted by $\hat{x}(\mathbf{s}_p)$, $p = 1, \ldots, P$, i.e., $\hat{\mathbf{x}}_{\mathrm{P}} = (\hat{x}_1, \ldots, \hat{x}_P)^T$.

c. Spatial random field model The data $\mathbf{x}_{\mathrm{S}}$ are assumed to represent samples from a spatial random field (SRF) $X_i(\omega)$, where the index $i = 1, \ldots, N$ denotes the spatial location $\mathbf{s}_i \in S_{\mathrm{N}}$. The expectation over the ensemble of probable states is denoted by $\mathbb{E}[X_i(\omega)]$, and the autocovariance function is given by $C_{i,j} := \mathbb{E}[X_i(\omega)\,X_j(\omega)] - \mathbb{E}[X_i(\omega)]\,\mathbb{E}[X_j(\omega)]$.

The pdf of *Gibbs SRFs* can be expressed in terms of an energy functional $H(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a set of *model parameters*, according to the Gibbs pdf [42, p. 51]

$$f_{\mathrm{X}}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}) = \frac{\mathrm{e}^{-H(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}. \tag{1}$$

The constant $Z(\boldsymbol{\theta})$, called the *partition function*, is the pdf normalization factor obtained by integrating $\mathrm{e}^{-H(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta})}$ over all the probable states $\mathbf{x}_{\mathrm{S}}$.

B. From continuum spaces to scattered data

The formulation based on (1) has its origins in statistical physics, and it has found applications in pattern analysis [6, 18] and Bayesian field theory, e.g. [15, 27]. In statistics,

TABLE I: Definitions of kernel functions used in Section IV below. The first three have compact support. Notation: $u = \|\mathbf{r}\|/h$ where $\|\mathbf{r}\|$ is the distance and h the bandwidth; $\mathbb{1}_A(u)$ is the indicator function of the set A, i.e., $\mathbb{1}_A(u) = 1$, $u \in A$ and $\mathbb{1}_A(u) = 0$, $u \notin A$.

Triangular	$K(u) = (1-u)\,\mathbb{1}_{|u|\leq 1}(u)$
Tricube	$K(u) = (1-u^3)^3\,\mathbb{1}_{|u|\leq 1}(u)$
Quadratic	$K(u) = (1-u^2)\,\mathbb{1}_{|u|\leq 1}(u)$
Gaussian	$K(u) = \exp(-u^2)$
Exponential	$K(u) = \exp(-|u|)$

this general model belongs to the exponential family of distributions that have desirable mathematical properties [5]. Our group used the exponential density in connection with a specific energy functional to develop Spartan spatial random fields (SSRF's) [13, 21–23]. In Section III we construct an explicitly discrete model motivated by SSRFs which adapts local interactions to general sampling networks and prediction grids by means of kernel functions.

C. Kernel weights

Let $K(\mathbf{r})$ be a non-negative-valued kernel that is either compactly supported or decays exponentially fast at large distances (e.g., the Gaussian or exponential function). We define kernel weights associated with the sampling points $\mathbf{s}_i$ and $\mathbf{s}_j$ as follows

$$K_{i,j} \doteq K\left(\frac{\mathbf{s}_i - \mathbf{s}_j}{h_i}\right) = K\left(\frac{\|\mathbf{s}_i - \mathbf{s}_j\|}{h_i}\right), \tag{2}$$

where $\|\mathbf{s}_i - \mathbf{s}_j\|$ is the distance (Euclidean or other) between two points $\mathbf{s}_i$ and $\mathbf{s}_j$, whereas h_i is the respective *kernel bandwidth* that adapts to local variations of the sampling pattern. The kernel weight $K_{j,i}$ is defined in terms of a bandwidth h_j. Hence, $K_{i,j} \neq K_{j,i}$ if the bandwidths h_i and h_j are different. Examples of kernel functions are given in Table I.

Let $D_{i,[k]}(S_N)$ denote the distance between $\mathbf{s}_i$ and its k-nearest neighbor in S_N ($k = 0$ corresponds to zero distance). We choose the local bandwidth associated with $\mathbf{s}_i$ according to

$$h_i = \mu\, D_{i,[k]}(S_N), \tag{3}$$

where $\mu > 1$ and $k > 1$ are model parameters. In several case studies involving Euclidean spaces of dimension $d = 1, 2, 3, 4$, we determined that $k = 2$ (second nearest neighbors)

performs well for compactly supported kernels and $k = 1$ (nearest neighbors) for infinitely supported kernels. Using $k = 2$ for compact kernels avoids zero bandwidth problems which result from $k = 1$ for collocated sampling and prediction points. Since the sampling point configuration is fixed, μ and $D_{i,[k]}(S_N)$ determine the local bandwidths. $D_{i,[k]}(S_N)$ depends purely on the sampling point configuration, but μ also depends on the sample values. For compactly supported kernels setting $k = 1$ only makes sense if $\mu > 1$; otherwise $h_i = D_{i,[k=1]}(S_N)$ implying that the kernel vanishes even for the nearest-neighbor pairs and thus fails to implement interactions.

D. Kernel averages

For any two-point function $\Phi(\cdot)$, we use a local-bandwidth extension of the Nadaraya-Watson kernel-weighted average over the network of sampling points [29, 41]

$$\langle \Phi(\cdot) \rangle_{\mathbf{h}} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} K_{i,j} \, \Phi(\cdot)}{\sum_{i=1}^{N} \sum_{j=1}^{N} K_{i,j}},$$

where $\mathbf{h} = (h_1, \ldots, h_N)^T$ is the vector of local bandwidths. The function $\Phi(\cdot)$ represents the distance between two points $\|\mathbf{s}_i - \mathbf{s}_j\|$ or the difference $x_i - x_j$ of the field values, or any other function that depends on the locations or the values of the field. The kernel average is normalized so as to preserve unity, i.e., $\langle 1 \rangle_{\mathbf{h}} = 1$ for all possible point configurations.

III. THE STOCHASTIC LOCAL INTERACTION (SLI) MODEL

The SLI joint pdf is determined below by means of the energy functional (4). This leads to a precision matrix which is explicitly defined in terms of local interactions and thus avoids the covariance matrix inversion. The prediction of missing data is based on maximizing the joint pdf of the data and the predictand, which is equivalent to minimizing the corresponding energy functional. This leads to the mode predictor (21), which involves a calculation with linear algorithmic complexity.

A. The energy functional

Consider a sample $\mathbf{x}_S$ on an unstructured sampling grid with sample mean μ_X. We propose the following energy functional $H_X(\mathbf{x}_S; \boldsymbol{\theta})$

$$H_X(\mathbf{x}_S; \boldsymbol{\theta}) = \frac{1}{2\lambda} \left[\mathcal{S}_0(\mathbf{x}_S) + \alpha_1\, \mathcal{S}_1(\mathbf{x}_S; \mathbf{h}_1) + \alpha_2\, \mathcal{S}_2(\mathbf{x}_S; \mathbf{h}_2) \right], \tag{4}$$

where $\boldsymbol{\theta} = (\mu_X, \alpha_1, \alpha_2, \lambda, \mu, k)$ is the SLI parameter vector and the parameters μ, k are defined in Section II C above.

The terms $\mathcal{S}_0(\mathbf{x}_S)$, $\mathcal{S}_1(\mathbf{x}_S; \mathbf{h}_1)$, and $\mathcal{S}_2(\mathbf{x}_S; \mathbf{h}_2)$ correspond to the averages of the square fluctuations, the square gradient and the square curvature in a Euclidean space of dimension d. The latter two are given by kernel-weighted averages that involve the *field increments* $x_{i,j} = x_i - x_j$.

$$\mathcal{S}_0(\mathbf{x}_S) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_X)^2, \tag{5}$$

$$\mathcal{S}_1(\mathbf{x}_S; \mathbf{h}_1) = d\, \langle x_{i,j}^2 \rangle_{\mathbf{h}_1}, \tag{6}$$

$$\mathcal{S}_2(\mathbf{x}_S; \mathbf{h}_2) = c_{2,1}\, \langle x_{i,j}^2 \rangle_{\mathbf{h}_2} - c_{2,2}\, \langle x_{i,j}^2 \rangle_{\mathbf{h}_3} - c_{2,3}\, \langle x_{i,j}^2 \rangle_{\mathbf{h}_4}, \tag{7a}$$

$$\text{where } c_{2,1} = 4d(d+2),\ c_{2,2} = 2d(d-1),\ c_{2,3} = d. \tag{7b}$$

The $c_{2,j}$ $(j = 1, 2, 3)$ values in $\mathcal{S}_2$ are motivated by discrete approximations of the square gradient and curvature [23]. We use two vector bandwidths, $\mathbf{h}_1$ and $\mathbf{h}_2$, to determine the range of influence of the kernel function around each sampling point for the gradient $\mathcal{S}_1(\mathbf{x}_n; \mathbf{h}_1)$ and curvature $\mathcal{S}_2(\mathbf{x}_n; \mathbf{h}_2)$ terms respectively. Additional bandwidths used in (7a) for $\mathcal{S}_2(\mathbf{x}_n; \mathbf{h}_2)$ are defined by $\mathbf{h}_3 = \sqrt{2}\,\mathbf{h}_2$, $\mathbf{h}_4 = 2\,\mathbf{h}_2$. These definitions are motivated by the formulation of SSRFs [21, 23].

B. SLI parameters and permissibility

To obtain realistic kernel bandwidths, k should be a positive integer larger than one, and μ should be larger than one. The parameter μ_X is set equal to the sample mean. The coefficients α_1, α_2 control the relative contributions of the mean square gradient and mean square curvature terms. The coefficient λ controls the overall amplitude of the fluctuations. Finally, μ and k control the bandwidth values as described in Section II C.

The SLI energy functional (4) is permissible if $H_X(\mathbf{x}_S; \boldsymbol{\theta}) \geq 0$ for all $\mathbf{x}_S$, a condition which ensures that $\mathrm{e}^{-H(\mathbf{x}_S; \boldsymbol{\theta})}$ is bounded and thus the existence of the partition function in (1). Assuming that $\mathcal{S}_2 \geq 0$ ($\mathcal{S}_0$ and $\mathcal{S}_1$ are always non-negative by construction), a sufficient permissibility condition, independently of the distance metric used, is $\alpha_1, \alpha_2, \lambda > 0$. In all the case studies that we have investigated, however, we have not encountered permissibility problems so long as $\alpha_1, \alpha_2, \lambda > 0$. Intuitively, the justification for the permissibility of (4) is that the first average, i.e., $\langle x_{i,j}^2 \rangle_{\mathbf{h}_2}$ in (7a) has a positive sign and is multiplied by $c_{2,1}$, which is significantly larger (especially as d increases) than the coefficients $c_{2,2}$ and $c_{2,3}$ multiplying the negative-sign averages $\langle x_{i,j}^2 \rangle_{\mathbf{h}_3}$ and $\langle x_{i,j}^2 \rangle_{\mathbf{h}_4}$. This property is valid for geodesic distances on the globe and for other metric spaces as well.

C. Precision matrix representation

We express (4) in terms of the *precision matrix* $\hat{J}_{i,j}(\boldsymbol{\theta})$ $(i, j = 1, \ldots, N)$

$$H_X(\mathbf{x}_S; \boldsymbol{\theta}) = \frac{1}{2}(\mathbf{x}_S - \boldsymbol{\mu}_X)^T \, \mathbf{J}(\boldsymbol{\theta}) \, (\mathbf{x}_S - \boldsymbol{\mu}_X). \tag{8}$$

The symmetric precision matrix $\mathbf{J}(\boldsymbol{\theta})$ follows from expanding the squared differences in (4), leading to the following expression

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{1}{\lambda} \left\{ \frac{\mathbf{I}_N}{N} + \alpha_1 \, d \, \mathbf{J}_1(\mathbf{h}_1) + \alpha_2 \left[c_{2,1} \, \mathbf{J}_2(\mathbf{h}_2) - c_{2,2} \, \mathbf{J}_3(\mathbf{h}_3) - c_{2,3} \, \mathbf{J}_4(\mathbf{h}_4) \right] \right\}, \tag{9}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix: $[\mathbf{I}_N]_{i,j} = 1$ if $i = j$ and $[\mathbf{I}_N]_{i,j} = 0$ otherwise, and $\mathbf{J}_q(\mathbf{h}_q)$, $q = 1, 2, 3, 4$ are *network matrices* that are determined by the sampling pattern, the kernel function, and the bandwidths. The index q defines the gradient network matrix for

$q = 1$, whereas the values $q = 2, 3, 4$ specify the curvature network matrices that correspond to the three terms in $\mathcal{S}_2(\mathbf{x}_S; \mathbf{h}_2)$ given by (7a). The elements of the network matrices $\mathbf{J}_q(\mathbf{h}_q)$ are given by the following equations

$$[\mathbf{J}_q(\mathbf{h}_q)]_{i,j} = -u_{i,j}(h_{q;i}) - u_{i,j}(h_{q;j}) + [\mathbf{I}_N]_{i,j} \sum_{l=1}^{N} [u_{i,l}(h_{q;i}) + u_{l,i}(h_{q;l})], \qquad (10a)$$

$$u_{i,j}(h_{q;i}) = \frac{K\left(\frac{\mathbf{s}_i - \mathbf{s}_j}{h_{q,i}}\right)}{\sum_{i=1}^{N}\sum_{j=1}^{N} K\left(\frac{\mathbf{s}_i - \mathbf{s}_j}{h_{q,i}}\right)}, \qquad q = 1, \ldots, 4. \qquad (10b)$$

The network matrices defined by (10) are symmetric by construction. It follows from (10) that the row and column sums vanish, i.e.,

$$\sum_{j=1}^{N} [\mathbf{J}_q(\mathbf{h}_q)]_{i,j} = 0. \qquad (11)$$

Based on (10a), the diagonal elements are given by the following expression

$$[\mathbf{J}_q(\mathbf{h}_q)]_{i,i} = \sum_{l=1,\neq i}^{N} [u_{i,l}(h_{q;i}) + u_{l,i}(h_{q;l})]. \qquad (12)$$

Since the kernel weights are non-negative, it follows that the sub-matrices $\mathbf{J}_q(\mathbf{h}_q)$ are *diagonally dominant*, i.e., $\left|[\mathbf{J}_q(\mathbf{h}_q)]_{i,i}\right| \geq \sum_{j\neq i} \left|[\mathbf{J}_q(\mathbf{h}_q)]_{i,j}\right|$. It also follows from (9) and (11) that

$$\sum_{j=1}^{N} [\mathbf{J}(\boldsymbol{\theta})]_{i,j} = \frac{1}{N\lambda}. \qquad (13a)$$

D. Parameter inference

We have experimented both with maximum likelihood estimation and leave-one-out cross validation. The former requires the calculation of the SLI partition function, which is an $O(N^3)$ operation for scattered data. For large data sets the $O(N^3)$ complexity is a computational bottleneck. Parameter inference by optimization of a cross validation metric is computationally more efficient, since it is at worst an $O(N^2)$ operation as we show below. The memory requirements for storing the precision matrix are $O(N^2)$ but can be signifi-

9

cantly reduced by using sparse matrix structures. Let $\boldsymbol{\theta}_{-\lambda} = (\alpha_1, \alpha_2, \mu, \mu_X)^T$ represent the parameter vector excluding λ. We use the following *cross validation cost functional*

$$\Phi(\mathbf{x}_S; \boldsymbol{\theta}_{-\lambda}) = \sum_{i=1}^{N} |\hat{x}_i(\boldsymbol{\theta}_{-\lambda}) - x_i|, \tag{14}$$

where $\hat{x}_i(\boldsymbol{\theta}_{-\lambda})$ is the SLI prediction at $\mathbf{s}_i$ based on the reduced sampling set $S_N - \{\mathbf{s}_i\}$ using the parameter vector $\boldsymbol{\theta}_{-\lambda}$ which applies to all $i = 1, \ldots, N$. The prediction is based on the interpolation equation (22) below and does not involve λ (see discussion in Section V B).

The optimal parameter vector excluding λ, i.e., $\boldsymbol{\theta}_{-\lambda}$, is determined by minimizing the cost functional (14):

$$\boldsymbol{\theta}^*_{-\lambda} = \arg\min_{\boldsymbol{\theta}_{-\lambda}} \Phi(\mathbf{x}_S; \boldsymbol{\theta}_{-\lambda}). \tag{15}$$

If $\tilde{H}(\mathbf{x}_S; \boldsymbol{\theta}_{-\lambda})$ is the energy estimated from (8) and (9) by setting $\lambda = 1$, the optimal value λ^* is obtained by minimizing the negative log-likelihood with respect to λ leading to the following solution (see A)

$$\lambda^* = \frac{2\tilde{H}(\mathbf{x}_S; \boldsymbol{\theta}_{-\lambda})}{N}. \tag{16}$$

We determine the minimum of the cross validation cost functional (14) using the MATLAB constrained optimization function `fmincon` with the *interior-point* algorithm [43]. This function determines the local optimum nearest to the initial parameter vector. We use initial guesses for the parameters α_1, α_2, μ, and we assume that the parameters are constrained between the lower bounds $[0.5, 0.5, 0.5]$ and the upper bounds $[300, 300, 15]$. We investigated different initial guesses for the parameters which led to different local optima. We found, however, that the value of the cross validation function is not very sensitive on the local optimum. In the $4D$ case study presented in Section IV B, we also estimate for comparison purposes the global optimum using MATLAB's global optimization tools.

E.   Predictive SLI model

Let us now assume that the prediction point $\mathbf{s}_p$ is added to the sampling points. To predict the unknown value of the field at $\mathbf{s}_p$, we insert this point in the energy functional (8), which is then given by Eq. (19) below. Then, we determine the mode of the joint pdf (1) with

the prediction point inserted in the energy functional. Thus, we obtain a *mode prediction equation* for $\hat{x}_p$ given by (22) below.

1. Modification of kernel weights

Upon inclusion of $\mathbf{s}_p$, the weights (10b) of the network matrices (10a) are modified as follows

$$u_{i,j}(h_{q;i}) = \frac{K\left(\frac{\mathbf{s}_i - \mathbf{s}_j}{h_{q,i}}\right)}{\sum_{i,j} K\left(\frac{\mathbf{s}_i - \mathbf{s}_j}{h_{q,i}}\right) + \sum_i K\left(\frac{\mathbf{s}_i - \mathbf{s}_p}{h_{q,i}}\right) + \sum_i K\left(\frac{\mathbf{s}_i - \mathbf{s}_p}{h_{q,p}}\right)}, \tag{17}$$

where $\sum_{i,j} := \sum_{i=1}^N \sum_{j=1}^N$. The first term in the denominator concerns interactions between sampling points. The second term involves local interactions between the sampling points and the prediction point which result from inserting the prediction point in the local neighborhoods of the sampling points, which control the bandwidths. Finally, the third term also involves interactions between the prediction point and the sampling points, but in this case the bandwidth is controlled by the former. Fig. 1 below illustrates the difference between the second and third term in the context of the entire precision matrix. The index q is used to distinguish between the weights linked to the gradient ($q = 1$) and the three weights ($q = 2, 3, 4$) linked to the curvature terms. The only difference between weights with different q is the bandwidth. In the case of compactly supported kernels, different bandwidths imply that different numbers of pairs are involved in the summations, since a pair separated by a distance that exceeds the bandwidth does not contribute. Calculation of the predictand contributions in the denominator of (17) is an operation with computational complexity $O(N)$ compared to $O(N^2)$ for the interactions between sampling points. The latter term, however, is calculated once and used for all the prediction points.

In addition to the weights that correspond to pairs of sampling points, there are weights for combinations of sampling and prediction points, i.e.,

$$u_{p,j}(h_{q;p}) = \frac{K\left(\frac{\mathbf{s}_p - \mathbf{s}_j}{h_{q,p}}\right)}{\sum_{i,j} K\left(\frac{\mathbf{s}_i - \mathbf{s}_j}{h_{q,i}}\right) + \sum_i K\left(\frac{\mathbf{s}_i - \mathbf{s}_p}{h_{q,i}}\right) + \sum_i K\left(\frac{\mathbf{s}_i - \mathbf{s}_p}{h_{q,p}}\right)}, \tag{18}$$

where $p = 1, \ldots, P$, $j = 1, \ldots, N$. The denominator of (18) is identical to that of (17).

2. SLI mode predictor

Using the precision matrix formulation, the energy functional including the predictand is given by

$$\hat{H}_X(\mathbf{x}_S, x_p; \boldsymbol{\theta}^*) = H_X(\mathbf{x}_S; \boldsymbol{\theta}^*) + J_{p,p}(\boldsymbol{\theta}^*)(x_p - \mu_X)^2 + \sum_{i=1}^{N}(x_i - \mu_X) J_{i,p}(\boldsymbol{\theta}^*)(x_p - \mu_X)$$

$$+ \sum_{i=1}^{N}(x_p - \mu_X) J_{p,i}(\boldsymbol{\theta}^*)(x_i - \mu_X). \tag{19}$$

The elements of the precision matrix that involve the prediction point are

$$[\mathbf{J}_q(\mathbf{h}_q)]_{p,p} = \sum_{i=1}^{N} [u_{i,p}(h_{q;i}) + u_{p,i}(h_{q;p})], \tag{20a}$$

$$[\mathbf{J}_q(\mathbf{h}_q)]_{i,p} = - [u_{i,p}(h_{q;i}) + u_{p,i}(h_{q;p})], \ i \neq p. \tag{20b}$$

Based on (20a) the symmetry property $J_{p,i}(\boldsymbol{\theta}^*) = J_{i,p}(\boldsymbol{\theta}^*)$ follows. The coefficients $u_{i,p}(h_{q;i})$ and $u_{p,i}(h_{q;p})$ differ due to the different bandwidths used (in the former, the bandwidth is determined by the neighborhood of the sampling point $\mathbf{s}_i$, whereas in the latter by the neighborhood of $\mathbf{s}_p$.) A schematic illustration of terms in (19) that involve the predictand is given in Fig. 1. The left diagram corresponds to terms "rooted" at $\mathbf{s}_p$ (i.e., with coefficient $u_{p,i}(h_{q;p})$ that involves the bandwidth h_p), whereas the right hand side diagram corresponds to terms rooted at the sampling points, i.e., with coefficients $u_{i,p}(h_{q;i})$.

The *SLI mode predictor* is defined by the following equation

$$\hat{x}_p = \arg\min_{x_p} \hat{H}_X(\mathbf{x}_S, x_p; \boldsymbol{\theta}^*), \tag{21}$$

where $\hat{H}_X(\mathbf{x}_S, x_p; \boldsymbol{\theta}^*)$ is given by (19). Minimization of the energy with respect to x_p leads to the following mode estimator
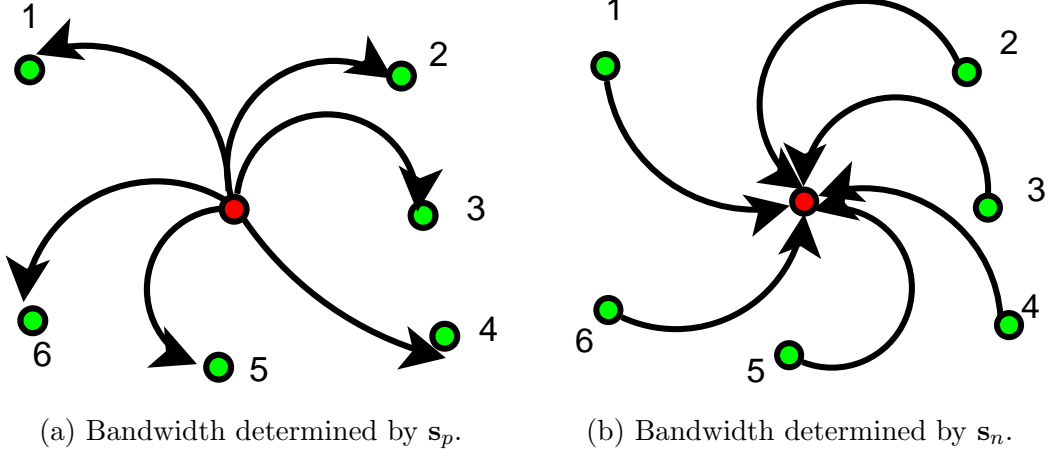
(a) Bandwidth determined by $\mathbf{s}_p$. (b) Bandwidth determined by $\mathbf{s}_n$.

FIG. 1: Schematic diagrams of terms contributing to (19) that include the prediction point $\mathbf{s}_p$ (center point) and six sampling points $\mathbf{s}_n$ ($n = 1, \ldots, 6$). The diagram on the left (a) represents terms $J_{p,i}$ whereas the diagram on the right (b) represents terms $J_{i,p}$. The point at the "root" of each arrow determines the bandwidth for the weight that involves the two points connected by the arrow.

$$\hat{x}_p = \mu_{\mathrm{X}} - \frac{\sum_{i=1}^{N} \left[J_{i,p}(\boldsymbol{\theta}^*) + J_{p,i}(\boldsymbol{\theta}^*) \right] (x_i - \mu_{\mathrm{X}})}{2 \, J_{p,p}(\boldsymbol{\theta}^*)}$$

$$= \mu_{\mathrm{X}} - \frac{\sum_{i=1}^{N} J_{p,i}(\boldsymbol{\theta}^*) (x_i - \mu_{\mathrm{X}})}{J_{p,p}(\boldsymbol{\theta}^*)}, \tag{22}$$

where the precision matrix elements are given by (10a) using the modified kernel weights (17) and (18).

The SLI mode predictor can be generalized to P prediction points as follows

$$\hat{\mathbf{x}}_p = \boldsymbol{\mu}_{\mathrm{X}} - \tilde{\mathbf{J}}_{P,S}(\boldsymbol{\theta}^*) (\mathbf{x} - \boldsymbol{\mu}_{\mathrm{X}}), \tag{23a}$$

where $\tilde{\mathbf{J}}_{P,S}(\boldsymbol{\theta}^*)$ is a $P \times N$ matrix given by

$$[\tilde{\mathbf{J}}_{P,S}(\boldsymbol{\theta}^*)]_{p,i} = J_{p,i}(\boldsymbol{\theta}^*)/J_{p,p}(\boldsymbol{\theta}^*). \tag{23b}$$

3. Properties of SLI predictor

The SLI prediction (23) is *unbiased* in view of the vanishing row sum property (13a) satisfied by the network matrices and the precision matrix. The SLI prediction (23) is independent of the parameter λ which sets the amplitude of the fluctuations, because the transfer matrix $\tilde{\mathbf{J}}_{P,S}(\boldsymbol{\theta}^*)$ is given by the ratio of precision matrix elements. This property is analogous to the independence of the kriging predictor from the random field variance. Hence, leave-one-out cross validation does not determine the optimal value of λ, which is obtained from (16).

The SLI predictor is not necessarily an exact interpolator. In particular, let us consider a point $\mathbf{s}_k$, $k \in \{1, \ldots, N\}$, which is very close to $\mathbf{s}_p$. Based on (20) and (22), $\hat{x}_p \to x_k$ as $\mathbf{s}_p \to \mathbf{s}_k$ only if (i) $|u_{k,p}(h_p)| \gg |u_{i,p}(h_p)|$ and (ii) $|u_{k,p}(h_k)| \gg |u_{i,p}(h_i)|$ for all $i \neq k$. Condition (i) materializes only for compactly supported kernels if $h_p \to 0$ which requires that the bandwidth be determined by the nearest neighbor distance. Condition (ii), on the other hand, requires that $\|\mathbf{s}_k - \mathbf{s}_p\|/h_k \ll \|\mathbf{s}_i - \mathbf{s}_p\|/h_i$ for $i \neq k$. This condition holds approximately at best if the sample is sparse around $\mathbf{s}_p$.

The computational complexity of the SLI predictor is $O(N^2 + P N)$. The $O(N^2)$ term is due to the double summation over the sampling points in (17), which needs to be calculated only once. The remaining operations per each prediction point scale linearly with the sample size, hence the $O(P N)$ dependence. Based on the above, the dominant term (for fixed P) in the computational time scales as $O(N^2)$. In future work we will investigate approximating the double summation in the denominator of (10b) and (18) with analytically evaluated double integrals over the kernel functions to increase the computational efficiency.

IV.   CASE STUDIES

We first consider two synthetic data sets, the first consisting of a time series and the second of a four-dimensional test function. We then investigate a set of scattered real data in two spatial dimensions.

A. Time series with Matérn covariance function

We generate a time series of length $N = 300$ from a random process with Matérn covariance $C(\tau) = \sigma^2 \, 2^{1-\nu} K_\nu(\tau/\xi)(\tau/\xi)^\nu / \Gamma(\nu)$, where $K_\nu(\cdot)$ is the modified Bessel function of order ν, $\Gamma(\cdot)$ is the gamma function, $\sigma = 10$, $\nu = 3.5$ is the smoothness index, and $\xi = 10$ is the correlation time. We use 60 randomly selected points as the training set (corresponding to an 80% degree of thinning) and the remaining 240 points as the validation set. The SLI optimal parameters using a quadratic kernel and $k = 2$ are given by $\alpha_1 \approx 29.30, \alpha_2 \approx 191.02, \mu \approx 1.11, \lambda \approx 297.84$. The sparseness of the precision matrix is evident in Fig. 2a. The darkest areas correspond to negative infinity and reflect distances for which the precision matrix vanishes.

The prediction performance is illustrated in the scatter plot of the SLI predictions versus the respective validation set values shown in Fig. 2b. The Pearson correlation coefficient between the validation values and the predictions is 0.89. The splitting of the time series into training and validation sets is shown in Fig. 3 along with the SLI predictions and associated error bars. The SLI predictions capture well general features of the time series. However, in areas of low sampling density the SLI predictions smoothes excessively the fluctuations in the original series. The SLI performance is excellent for the same degree of thinning, if the length of the initial time series increases to $3\,000$. On the other hand, the prediction accuracy deteriorates for rougher random processes, such as a non-differentiable Matérn process with $\nu = 0.8$.

B. Four-dimensional deterministic test function

We consider the function $x(\mathbf{s})$

$$x(\mathbf{s}) = A \, e^{-2\|\mathbf{s}-\mathbf{a}\|} \prod_{i=1}^{4} s_i \, (1 - s_i), \tag{24}$$

where $A = 500$ and $\mathbf{a} = (0.3, 0.3, 0.3, 0.3)$, defined over the four-dimensional cube with unit length edges, i.e., for $\mathbf{s} \in [0, 1]^4$. We sample the function at $N = 1\,000$ randomly selected points over the unit cube, and we generate a validation set of $N = 1\,000$ points also by random selection. The SLI optimal parameters for the quadratic kernel with $k = 2$ are given by $\alpha_1 \approx 10.12, \alpha_2 \approx 25.04, \mu \approx 1.64, \lambda \approx 0.0193$ starting with initial values

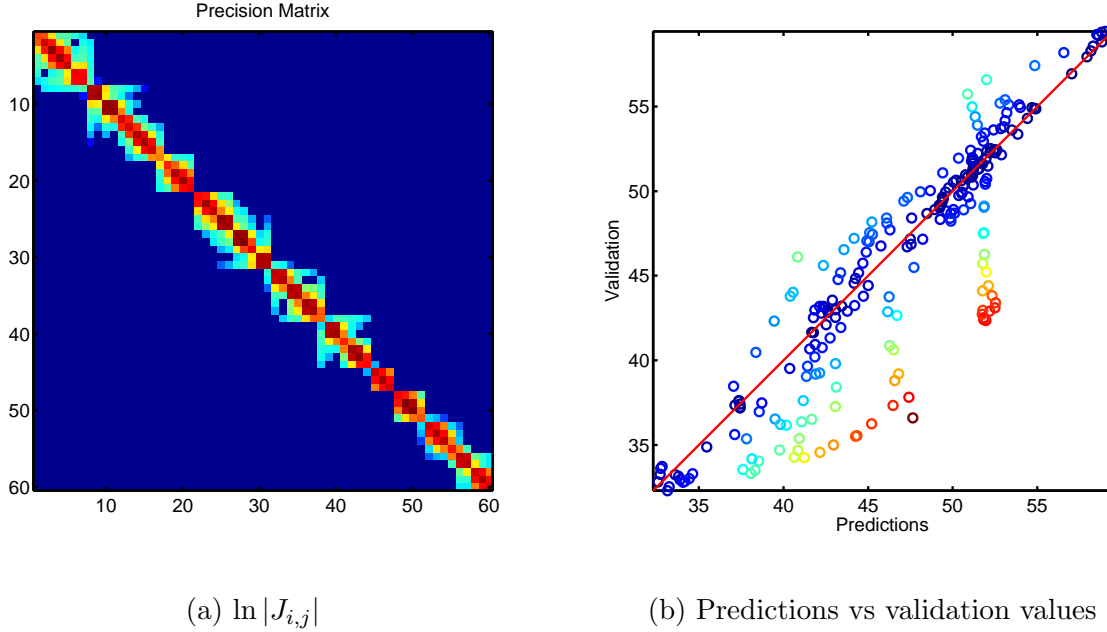(a) $\ln|J_{i,j}|$ (b) Predictions vs validation values

FIG. 2: Analysis of time series with Matérn correlations ($\sigma = 10$, $\nu = 3.5$, $\xi = 10$). (a) Logarithm of absolute value of the precision matrix; dark areas (blue online) correspond to low values whereas lighter areas (green to red online) correspond to higher values. (b) Scatter plot of SLI predictions versus the respective values of the validation set.

$\alpha_1 = 10, \alpha_2 = 25, \mu = 3$. Similar results in terms of cross validation performance are also obtained with different initial conditions that lead to different local optima. The cross validation measures for the parameters above are given by ME= 0.0046, MAE= 0.0320, RMSE= 0.0459, r= 0.96. The sparse structure of the precision matrix is illustrated in Fig. 4a which displays the logarithm of the absolute value. The scatter plot of the validation values versus the respective SLI predictions is shown in Fig. 4b and demonstrates very good agreement at most points.

We repeat the experiment by adding Gaussian noise to the sample. The standard deviation of the noise is set to $\approx 10\%$ of the maximum sampled value $x_{\max}$ (in the simulations that we ran $x_{\max} \approx 1$). While the coefficients α_1 and α_2 remain practically unchanged, μ changes to ≈ 1.83. The sparsity of the precision matrix is $\approx 76\%$. The respective cross validation measures are given by ME= 0.012, MAE= 0.047, RMSE= 0.061, and r= 0.93.

We also used the MATLAB global optimizer `GlobalSearch` with the same initial parameter vector as above to determine the SLI model parameters. `GlobalSearch` uses a scatter-search algorithm to generate starting points (initial parameter guesses). The minimization is conducted using `fmincon` to determine the local minimum close to the current starting point.
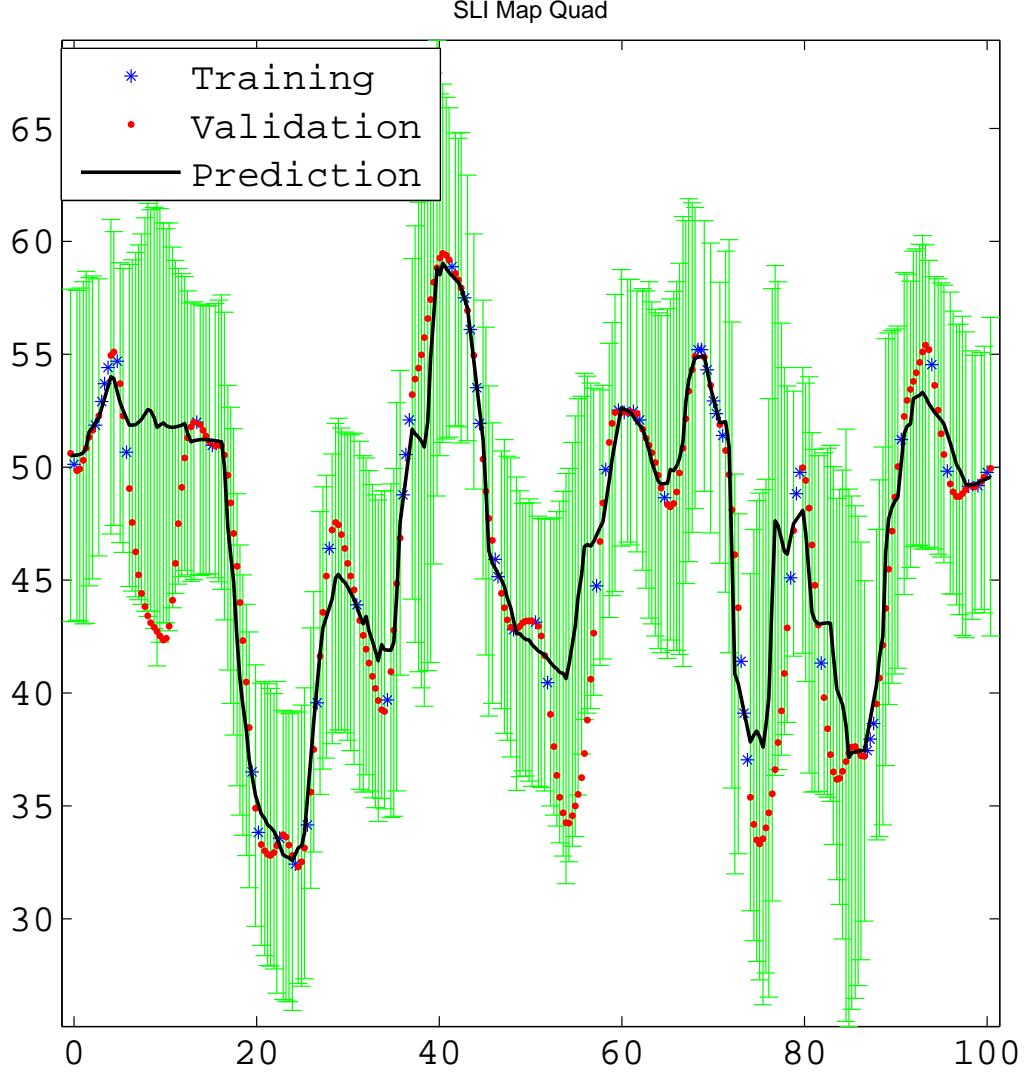
FIG. 3: SLI predictions at 240 validation points of time series with Matérn correlations ($\sigma = 10$, $\nu = 3.5$, $\xi = 10$). The 60 training points are marked by stars (blue online), validation points are marked by dots (red online), and SLI predictions are marked by the continuous line (black online). Error bars are based on three times the conditional standard deviation (green online).

We use the lower and upper bounds defined in Section III D to constrain the space of the starting points. `GlobalSearch` investigates a set of 66 starting points, and convergence to a local minimum is achieved for all of them. The globally optimal SLI parameters are estimated as $\alpha_1 \approx 1.50$, $\alpha_2 \approx 224.62$, $\mu \approx 2.01$, $\lambda \approx 0.748$. The respective validation measures are given by ME= 0.0055, MAE= 0.045, RMSE= 0.063, and $r = 0.93$. These measures do not differ

(a) $\ln|J_{i,j}|$
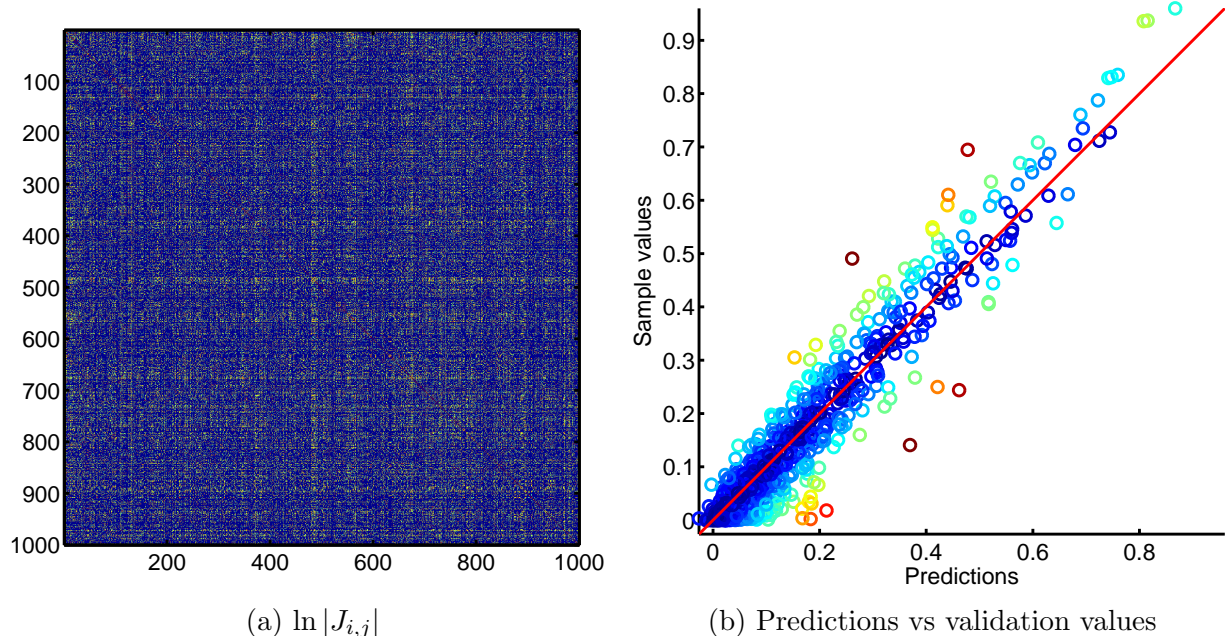
(b) Predictions vs validation values

FIG. 4: Analysis of the truncated exponential function (24). (a) Logarithm of the absolute value of the precision matrix; dark areas (blue online) correspond to low values whereas lighter areas (green to red online) correspond to higher values. (b) Scatter plot of SLI predictions versus the respective values of the validation set.

significantly from those obtained with the locally optimum solution. The MAE value is lower for the global optimum, which is expected since MAE reflects the value of the cost function (14). On the other hand, the RMSE obtained with the global optimum is slightly higher than that of the local optimum. This result indicates that quite different parameter vectors can lead to similar cross validation results. This behavior has also been observed with covariance models whose parameter vector involves more than the variance and the correlation length [24, 46].

C. Radioactivity data in two dimensions

This example focuses on daily means of radioactivity gamma dose rates over part of the Federal Republic of Germany. The data were provided by the German automatic radioactivity monitoring network for the Spatial Interpolation Comparison Exercise 2004 (SIC 2004) [12]. This data set is well studied and thus allows easy comparisons with other methods [13]. The 1 008 stations are partitioned into a *training set* of 200 randomly selected locations and a *validation set* of 808 locations where predictions are compared with the

18

observations. Two different scenarios are investigated: A *normal* data set corresponding to typical background radioactivity measurements, and an *emergency* data set, in which a local release of radioactivity in the southwest corner of the monitored area was simulated using a dispersion process to obtain a few values with magnitudes around 10 times higher than the background. The rates are measured in nanoSievert per hour (nSv/h). The normal training set follows the Gaussian distribution with the minimum around 58 nSv/h and the maximum around 153 nSv/h. In the emergency training set there are two values $> 1\,000$ nSv/h, with the maximum at $1\,499$ nSv/h. We compare the prediction performance of the SLI model against the 808 values of the validation set.

1. *Normal data*

For normal data, the optimal SLI parameters based on the training set with a quadratic kernel and $k = 2$ are given by $\alpha_1 \approx 143, \alpha_2 \approx 47.56, \mu \approx 2.64, \lambda \approx 3.24 \times 10^3$. Figure 5a illustrates the relative values of the bandwidths used. Higher values correspond to more isolated points in areas of low sampling density and along the boundaries of the convex hull of the domain. Figure 5b presents the natural logarithm of the absolute value of the precision matrix. Overall, about 32% (i.e., $12\,718$) of the total number of pairs yield nonzero precision values, implying that the sparsity of the precision matrix is $\approx 68\%$.

The cross-validation results are tabulated in Table II. The cross validation measures (based on the validation set) obtained in a recent study by means of *Ordinary Kriging* are: ME$= -1.36$, MAE$= 9.29$, RMSE$= 12.59$, $r= 0.78$ [13]. These values are in close agreement with the SLI results in Table II.

Various geostatistical and machine learning methods have been applied to the SIC 2004 data (neural networks, geostatistics, and splines). Excluding the results of some poor performers, the cross validation measures obtained are in the following ranges [12]: ME $\in [-1.39, -0.04]$ and $\in [0.20, 1.60]$, MAE $\in [9.05, 12.10]$, RMSE $\in [12.43, 15.90]$, and r $\in [0.64, 0.79]$. Hence, the SLI cross validation results are close to the best performers. Fig. 6 presents a map of the radioactivity pattern generated by SLI and contrasts it with the map generated by bilinear interpolation. The SLI spatial pattern is smoother and thus appears more realistic. Its smoothing effect near the sample values, however, is more pronounced than that caused by bilinear interpolation.

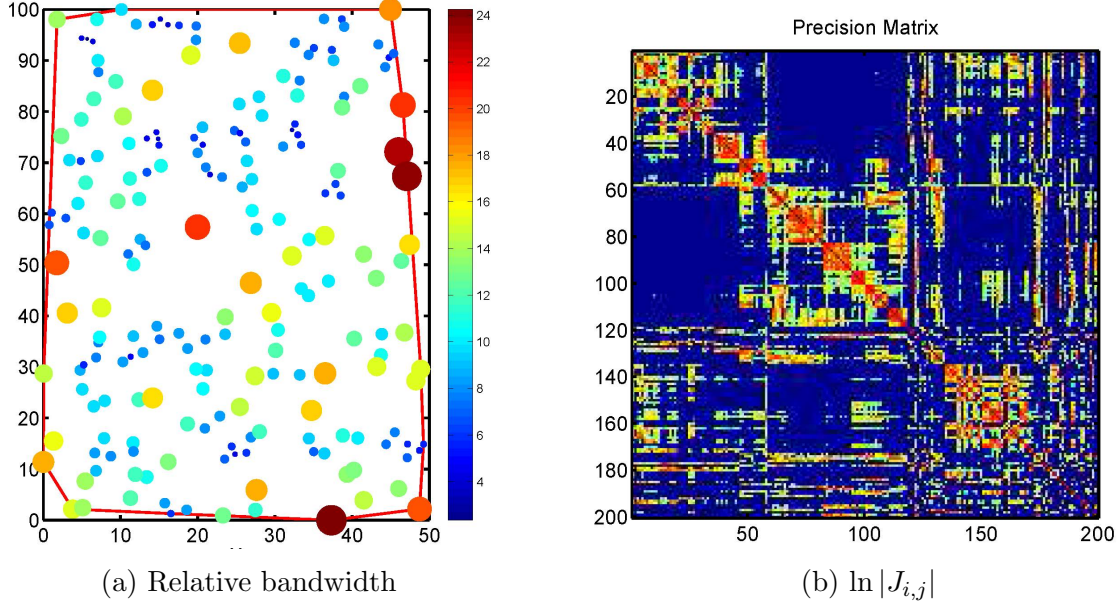| | (a) Relative bandwidth | (b) $\ln|J_{i,j}|$ |

FIG. 5: Analysis of SIC 2004 normal data set using normalized coordinates; the longest side is set to 100 and the aspect ratio is maintained. (a) Bubble plot of the relative size of local bandwidths: larger circles correspond to bigger bandwidths. The continuous line along the domain boundary marks the convex hull of the sampling set. (b) Logarithm of the absolute values of the precision matrix elements. Darker areas (blue online) correspond to lower values, whereas lighter areas (red online) correspond to higher values.

TABLE II: Cross validation performance measures for SIC 2004 normal data. The second row presents the performance of the SLI predictor at the 808 validation set points. ME: Mean error (bias); MAE: Mean absolute error; MARE: Mean absolute relative error; RMSE: Root mean square error; r: Pearson correlation coefficient.

SLI	ME	MAE	MARE	RMSE	r
Validation set	−1.30	9.30	0.09	12.62	0.78

We also conduct a stability analysis by removing one sampling point at a time and determining the optimal SLI model using leave-one-out cross validation with that point removed. The variation of the SLI parameters is shown in Fig. 7; α_1, α_2 and μ are quite stable, whereas λ shows more variability. The spikes in the plots of Fig. 7 are exaggerated by using a narrow vertical range to better illustrate the parameter variability. For α_1, α_2 the maximum relative variation (with respect to the mean) ranges from a fraction of a thousandth (for α_1) to few thousandths (for α_2); μ shows stronger variations, whereas the strongest variation is exhibited by λ, since the latter is a scaling factor that determines the overall energy of the ensemble of points and compensates for variations in the other parameters. We believe that the parameter variations exhibited in Fig. 7 are, at least partially due to extremely
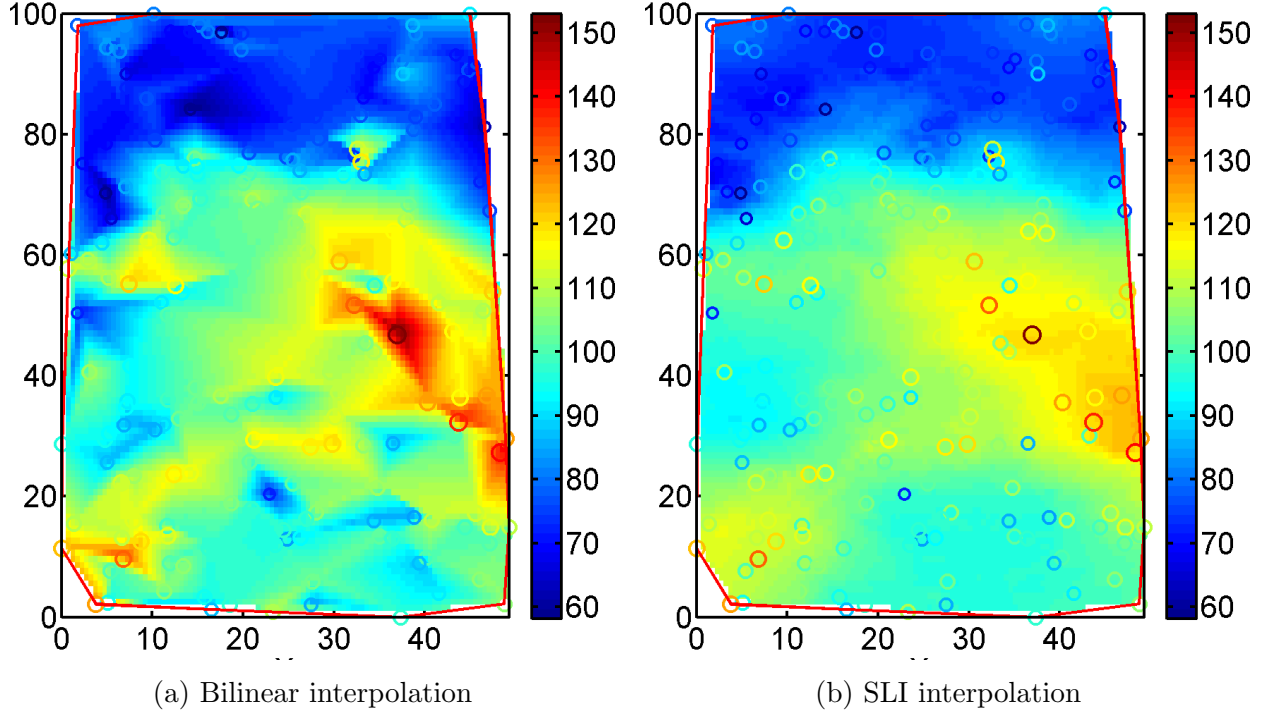
20

(a) Bilinear interpolation

(b) SLI interpolation

FIG. 6: Map of background radioactivity rates in Germany (based on 200 training data) using a mapping grid with 100 nodes per side. (a) Bilinear interpolation using the *griddata* command of MATLAB. (b) SLI interpolation map using the optimal parameters reported in the text.

slow variation of the cost function over a region of the parameter space, a condition also observed in maximum likelihood estimation of spatial models with Matérn covariance [46]. This slow variation implies *quasi-degeneracy* of the parameter vector; the quasi-degeneracy implies that vectors which are very far in parameter space may lead to very similar cost function values. More recently, the difficulties involved in nonlinear fits of multi-parametric models to data have been investigated in [37].

2. Emergency data

For the SIC 2004 emergency data, cross validation results with different kernel functions (tricubic, exponential, and quadratic) are shown in Table III. The SLI parameters are initialized using the optimal values for the normal data. The last row of the table is based on estimation with a quadratic kernel function after removing the three highest values. The best results in Table III are obtained with the quadratic kernel including all the data. The optimal SLI parameters are $\alpha_1 \approx 143, \alpha_2 \approx 47.56, \mu \approx 2.69, \lambda \approx 4.32 \times 10^5$. The parameters,
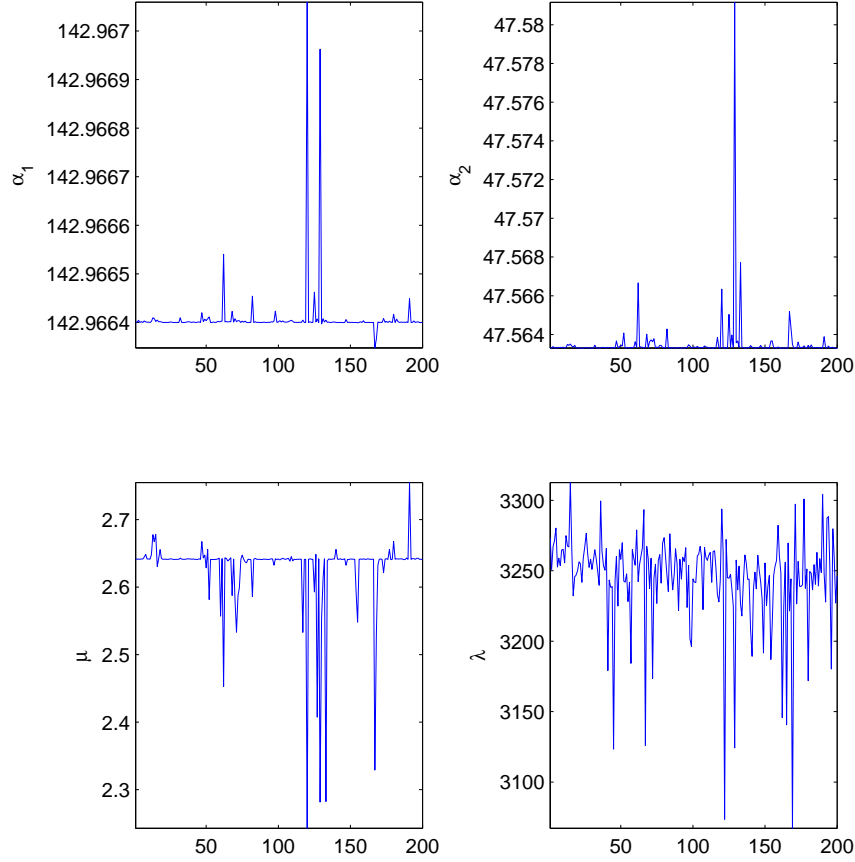
21

FIG. 7: Variation of SLI parameters estimated by removing one value at a time from the 200 training locations of the SIC 2004 normal data.

except for λ, are close to their normal case counterparts. The difference in λ is due to the much higher variance of the emergency data set.

The variation of the SLI parameters in leave-one-out cross validation exhibits similar patterns as for the normal data, except that more pronounced variations of λ are observed when the extreme values are removed. The precision matrix has 23 232 non-zero elements, implying a sparsity of $\approx 42\%$, in contrast with 12 718 (sparsity $\approx 32\%$) in the normal case. This difference clearly illustrates the dependence of the precision matrix on the sample values in addition to the sampling pattern. SLI does not rely on estimating the variogram function, and thus it is not hindered by the presence of extreme values. On the other hand, geostatistical methods rely on the variogram function, which may not be reliably estimated in such cases [19]. The Pearson correlation coefficient is significantly lower than in the normal set due to underestimation of the extreme values, while the Spearman rank correlation coefficient is comparable to the normal case. The cross validation measures obtained in SIC

2004 are in the following intervals [12]: $ME \in [-11.10, -0.12]$ and $\in [0.41, 19.71]$, MAE $\in [14.85, 146.36]$, RMSE $\in [45.46, 212.10]$, and $r \in [0.02, 0.86]$. The best performance in terms of both MAE and RMSE was obtained by means of a Generalized Regression Neural Network [36]. Looking at the scatter plot of MAE versus RMSE —Fig. 6 in [11]— the SLI performance is closer to the geostatistical and spline methods.

TABLE III: SLI cross validation performance measures for SIC 2004 emergency data. The second row presents the performance of the SLI predictor at the 808 validation set points. The first five cross validation measures are as described in the caption of Table II, and r_S is the Spearman correlation coefficient.

SLI Kernel function	ME	MAE	MARE	RMSE	r	r_S
Tricubic: $(1 - u^3)^3$	5.78	24.22	0.20	81.33	0.25	0.57
Exponential: e^{-u}	6.06	23.84	0.19	79.78	0.34	0.63
Quadratic	3.04	23.16	0.17	75.63	0.43	0.77
Quadratic (outliers removed)	-8.28	16.46	0.10	81.41	0.27	0.77

V. DISCUSSION

A. Connections with machine learning

The SLI model is similar to k-Nearest Neighbors (KNN), since both methods employ an optimal neighborhood range. In the case of KNN a uniform optimal number of nearest neighbors is determined, and the estimate at an unmeasured point is simply the mean of its k nearest neighbors. In SLI, a locally optimal neighborhood size is determined implying that the number of neighbors used in prediction varies locally. In addition, the estimate is a weighted mean of the neighbor values, in which the weights are determined by the kernel function and the bandwidths. In this respect, SLI is similar to the Nadaraya-Watson kernel regression method [29, 41] and to the Support Vector Machine algorithm [39]. SLI can also be viewed as a particular type of Gaussian process with a sparse inverse covariance kernel, which could be used as an alternative to the sparse Gaussian process framework to improve the computational efficiency of predictions [9].

In this study we formulated the SLI model using the spatial locations S_N as inputs and the respective values of the scalar field values as outputs. This framework is appropriate for scattered spatial data. It is possible, however, to use more general input variables instead

23

of the spatial locations, so long as a suitable measure of distance can be defined.

B.   Notes on implementation

We presented a "plain vanilla" version of the SLI model. Modifications that can increase the flexibility but also the complexity of the model are possible. The local kernel bandwidths are determined by fixing the neighbor order k and using a uniform scaling parameter μ. Alternatively, one can consider estimating k from the data and using a locally varying μ. With respect to the latter, potential gains should be weighted against the loss of computational efficiency that will result from the significant increase of the parameter vector size. While our estimate of μ_X is based on the sample mean, it is possible to estimate μ_X by means of the leave-one-out cross validation procedure. It is also possible to replace μ_X with a space-dependent trend function.

The present version of the SLI model does not involve anisotropy. Nevertheless, anisotropy is important in cases such as the radioactivity emergency data [33]: the best performing method in SIC 2004 for this set was a general regression neural network with an anisotropic Gaussian kernel function. Similarly, in SLI it is possible to use weighted Euclidean distances or Minkowski metrics instead of the classical Euclidean distance [4]. SLI can also be extended to spherical surfaces, a case which is relevant for global geospatial data. In addition, the SLI model can capture correlations in higher-dimensional, abstract feature spaces equipped with a suitable distance.

At this point there is no rigorous physical interpretation of the coefficients α_1, α_2 and λ. In general, higher values of α_1 (α_2) imply higher cost for gradient (curvature), whereas λ controls the overall "energy". In the continuum case (i.e., for Spartan random fields) coefficients α_1 and α_2 are related to a rigidity coefficient and a characteristic length [21, 24]. A similar correspondence can also be established for data distributed on rectangular grids. In contrast, such relations are not available for scattered data. Even in the continuum and grid cases, however, statistical measures such as the variance and the correlation length have a nonlinear dependence on the SSRF model parameters [21, 24]. A reasonable initial value for μ is around 2–3, to allow even compactly supported kernel functions to build local neighborhoods containing at least a few data points. For α_1 and α_2, we have used positive values between the arbitrary bounds of 0.5 and 300. Exploratory runs with different initial

conditions can help to locate a reasonable starting point. Alternatively, a global optimization approach can be used as in Section IV B.

We have opted for a cross-validation cost functional which is based on the mean absolute error. It is possible to use different cost functionals that involve a linear combination of validation measures such as the mean absolute error and the root mean square error. Most results for the case studies investigated above were obtained using an interior-point optimization method that searches for local minima of the cost function. In all of the cases that we have investigated (including data not presented herein), the local optimization led to reasonable cross validation measures which were comparable to those obtained with other methods. As we have shown in the case of $4D$ synthetic data, searching for global optima does not necessarily lead to significant performance improvement. The investigation of global optimization methods with different data sets, however, deserves further attention.

VI.   CONCLUSIONS

The SLI model presented above provides a bridge between geostatistics and machine learning. It is based on an exponential joint density which involves an energy functional with an explicit precision (inverse covariance) matrix. The latter is constructed by superimposing network sub-matrices that implement local interactions between neighboring field values in terms of kernel functions. The algorithmic complexity of SLI missing value estimation scales linearly with the sample size except for a global $O(N^2)$ term which is, however, computed once for all the prediction points. Hence, the leave-one-out cross-validation approach can be used to efficiently infer the SLI model parameters.

For missing data on rectangular grids (ongoing research) the computational complexity of the SLI method can be simplified to linear scaling with N, because $\mathcal{S}_1$ and $\mathcal{S}_2$ can be calculated without kernel functions [47, 48]. In addition, calculating and storing the large $N \times N$ distance matrix is not necessary in this case. In conclusion, the SLI model is a promising tool for the analysis of big spatial data. In future research we will investigate the extension of the model to space-time data. Finally, the MATLAB code used for the case studies in Section IV is available at the web address of the Geostatistics laboratory: `http://www.geostatistics.tuc.gr/4940.html`.

ACKNOWLEDGMENT

REFERENCES

[1] Addair, T. G., D. A. Dodge, W. R. Walter, and S. D. Ruppert (2014). Large-scale seismic signal analysis with hadoop. *Computers & Geosciences 66*, 145–154.

[2] Adler, R. J. (1981). *The Geometry of Random Fields*. New York: Wiley.

[3] Ahrens, J., B. Hendrickson, G. Long, S. Miller, R. Ross, and D. Williams (2011). Data-intensive science in the US DOE: Case studies and future challenges. *Computing in Science & Engineering 13*(6), 14–24.

[4] Atkenson, S. G., A. W. Moore, and S. Schaal (1997). Locally weighted learning. *Artificial Intelligence Review 11*(1-5), 11–73.

[5] Barndorff-Nielsen, O. (2014). *Information and Exponential Families in Statistical Theory*. West Sussex, UK: John Wiley & Sons.

[6] Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological) 36*, 192–236.

[7] Chilès, J. P. and P. Delfiner (2012). *Geostatistics: Modeling Spatial Uncertainty* (2nd ed.). New York: Wiley.

[8] Cressie, N. and G. Johannesson (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(1), 209–226.

[9] Csató, L. and M. Opper (2002). Sparse on-line gaussian processes. *Neural Computation 14*(3), 641–668.

[10] Du, J., H. Zhang, and V. S. Mandrekar (2009). Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *The Annals of Statistics 37*(6A), 3330–3361.

[11] Dubois, G. (1998). Spatial interpolation comparison 97: Foreword and introduction. *Journal of Geographic Information and Decision Analysis 2*(2), 1–10.

[12] Dubois, G. and S. Galmarini (2006). Spatial interpolation comparison (SIC) 2004: introduction to the exercise and overview of results. In G. Dubois (Ed.), *Automatic Mapping Algorithms for Routine and Emergency Monitoring*, Volume EUR 21595 EN, pp. 7–18. Luxembourg, European Communities: Office for Official Publications of the European Communities.

[13] Elogne, S. N., D. Hristopulos, and E. Varouchakis (2008). An application of Spartan spatial random fields in environmental mapping: focus on automatic mapping capabilities. *Stochastic Environmental Research and Risk Assessment 22*(5), 633–646.

[14] Elogne, S. N., C. Thomas, and O. Perrin (2008). Nonparametric estimation of smooth stationary covariance functions by interpolation methods. *Statistical Inference and Stochastic Processes 11*(2), 177–205.

[15] Farmer, C. L. (2007). Bayesian field theory applied to scattered data interpolation and inverse problems. In A. Iske and J. Levesley (Eds.), *Algorithms for Approximation*, pp. 147–166. Heidelberg: Springer-Verlag.

[16] Furrer, R., M. G. Genton, and D. Nychka (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics 15*(3), 502–523.

[17] García-Soidán, P. H., M. Febrero-Bande, and W. González-Manteiga (2004). Nonparametric kernel estimation of an isotropic semivariogram. *Journal of Statistical Planning and Inference 121*(1), 65–92.

[18] Geman, S. and D. Geman (1984, Nov). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*(6), 721–741.

[19] Giraldi, N. and S. Bengio (2006). Machine learning for automatic environmental mapping: when and how? In G. Dubois (Ed.), *Automatic Mapping Algorithms for Routine and Emergency Monitoring*, Volume EUR 21595 EN, pp. 123–138. Luxembourg, European Communities: Office for Official Publications of the European Communities.

[20] Hall, P., N. Fisher, and B. Hoffman (1994). Properties of nonparametric estimators of autocovariance for stationary random fields. *Annals of Statistics 22*(4), 2115–2134.

[21] Hristopulos, D. (2003). Spartan Gibbs random field models for geostatistical applications. *SIAM Journal of Scientific Computing 24*(6), 2125–2162.

[22] Hristopulos, D. T. and S. Elogne (2007). Analytic properties and covariance functions of a new class of generalized Gibbs random fields. *IEEE Transactions on Information Theory 53*(12), 4667–4679.

[23] Hristopulos, D. T. and S. N. Elogne (2009). Computationally efficient spatial interpolators based on Spartan spatial random fields. *IEEE Transactions on Signal Processing 57*(9), 3475–3487.

[24] Hristopulos, D. T. and M. Žukovič (2011). Relationships between correlation lengths and integral scales for covariance models with more than two parameters. *Stochastic Environmental Research and Risk Assessment 25*(1), 11–19.

[25] Kanevski, M. and M. Maignan (2004). *Analysis and Modelling of Spatial Environmental Data*. Lausanne, Switzerland: EPFL Press.

[26] Kaufman, C. G., M. J. Schervish, and D. W. Nychka (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association 103*(484), 1545–1555.

[27] Lemm, J. C. (2005). *Bayesian Field Theory*. Baltimore: Johns Hopkins University Press.

[28] Lindgren, F. and H. Rue (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The SPDE approach. *Journal of the Royal Statistical Society, Series B 73*(4), 423498.

[29] Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability and its Applications 9*(1), 141–142.

[30] Nychka, D., S. Bandyopadhyay, D. Hammerling, F. Lindgren, and S. Sain (2014). A multi-resolution gaussian process model for the analysis of large spatial data sets. *Journal of Computational and Graphical Statistics* (just-accepted), 00–00.

[31] Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. Boston, MA: MIT Press.

[32] Rue, H. and L. Held (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC.

[33] Spiliopoulos, I., D. T. Hristopulos, M. P. Petrakis, and A. Chorti (2011). A multigrid method for the estimation of geometric anisotropy in environmental data from sensor networks. *Com-*

puters & Geosciences 37(3), 320–330.

[34] Steed, C. A., D. M. Ricciuto, G. Shipman, B. Smith, P. E. Thornton, D. Wang, X. Shi, and D. N. Williams (2013). Big data visual analytics for exploratory earth system simulation analysis. *Computers & Geosciences 61*, 71–82.

[35] Sun, Y., B. Li, and M. G. Genton (2012). Geostatistics for large datasets. In E. Porcu, M. Montero, J, and M. Schlather (Eds.), *Advances and Challenges in Space-time Modelling of Natural Events*, Lecture Notes in Statistics, pp. 55–77. Springer Berlin Heidelberg.

[36] Timonin, V. and E. Savelieva. (2006). Spatial prediction of radioactivity using general regression neural network. In G. Dubois (Ed.), *Automatic Mapping Algorithms for Routine and Emergency Monitoring*, Volume EUR 21595 EN, pp. 53–54. Luxembourg, European Communities: Office for Official Publications of the European Communities.

[37] Transtrum, M. K., B. B. Machta, and J. P. Sethna (2010). Why are nonlinear fits to data so challenging? *Physical Review Letters 104*, 060201.

[38] Van Duijn, M. et al. (2009). A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks 31*(1), 52.

[39] Vapnik, V. N. (2000). *The Nature of Statistical Learning*. New York: Springer Verlag.

[40] Varin, C., N. Reid, and D. Firth (2011). An overview of composite likelihood methods. *Statistica Sinica 21*(1), 5–42.

[41] Watson, G. S. (1964). Smooth regression analysis. *Sankhya Ser. A 26*(1), 359–372.

[42] Winkler, G. (1995). *Image Analysis, Random Fields and Dynamic Monte Carlo Methods: A Mathematical Introduction*. New York: Springer Verlag.

[43] Wright, M. (2005). The interior-point revolution in optimization: history, recent developments, and lasting consequences. *Bulletin of the American Mathematical Society 42*(1), 39–56.

[44] Wu, X., X. Zhu, G.-Q. Wu, and W. Ding (2014). Data mining with big data. *Knowledge and Data Engineering, IEEE Transactions on 26*(1), 97–107.

[45] Yaglom, A. M. (1987). *Correlation Theory of Stationary and Related Random Functions I*. New York: Springer Verlag.

[46] Žukovič, M. and D. Hristopulos (2009). The method of normalized correlations: a fast parameter estimation method for random processes and isotropic random fields that focuses on short-range dependence. *Technometrics 51*(2), 173–185.

[47] Žukovič, M. and D. T. Hristopulos (2013a). A directional gradient-curvature method for gap filling of gridded environmental spatial data with potentially anisotropic correlations. *Atmospheric Environment 77*, 901–909.

[48] Žukovič, M. and D. T. Hristopulos (2013b). Reconstruction of missing data in remote sensing images using conditional stochastic optimization with global geometric constraints. *Stochastic Environmental Research and Risk Assessment 27*(4), 785–806.

Appendix A: Minimization of NLL

For the pdf given by (1), the log-likelihood is given by

$$\mathrm{LL}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}) \doteq \ln L(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}) = -H_{\mathrm{X}}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}) - \ln Z(\boldsymbol{\theta}). \tag{A1}$$

The partition function in (A1) is given by the multiple integral

$$Z(\boldsymbol{\theta}) = \prod_{i=1}^{N} \int_{-\infty}^{\infty} dx_i \, \exp\left(-H_{\mathrm{X}}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta})\right). \tag{A2}$$

The square gradient and square curvature terms do not depend on μ_{X} because they involve differences $x_i - x_j$. Hence, we can express (8) as follows

$$H_{\mathrm{X}}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}) = \frac{1}{2}\mathbf{x}_{\mathrm{S}}^{T} \, \mathbf{J}(\boldsymbol{\theta}) \, \mathbf{x}_{\mathrm{S}} + \frac{\mu_{\mathrm{X}}^{2}}{\lambda} - \frac{2\mu_{\mathrm{X}} \overline{\mu}_{\mathrm{X}}}{\lambda}, \tag{A3}$$

where $\overline{\mu}_{\mathrm{X}}$ is the sample mean. Maximizing the NLL with respect to μ_{X}, using (A3) for the energy functional, yields

$$\mu_{\mathrm{X}} = \overline{\mu}_{\mathrm{X}}.$$

Since this fixes the parameter μ_{X}, we can use expression (8) for the energy functional. We apply the scaling transformation $H_{\mathrm{X}}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}) = \tilde{H}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}_{-\lambda})/\lambda$, where $\boldsymbol{\theta}_{-\lambda}$ is the parameter vector except for λ and $\tilde{H}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}_{-\lambda})$ is λ-independent. The transformation $H(\cdot) \mapsto \tilde{H}(\cdot)$ is equivalent to $x_i \mapsto y_i = (x_i - \overline{\mu}_{\mathrm{X}})/\sqrt{\lambda}$. Let us then define the scaled partition function

$\tilde{Z}(\boldsymbol{\theta}_{-\lambda})$ by means of

$$\tilde{Z}(\boldsymbol{\theta}_{-\lambda}) = \prod_{i=1}^{N} \int_{-\infty}^{\infty} dy_i \, \exp\left(-\tilde{H}(\mathbf{y}; \boldsymbol{\theta}_{-\lambda})\right)$$

$$= \lambda^{-N/2} \prod_{i=1}^{N} \int_{-\infty}^{\infty} dx_i \, \exp\left(-H(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta})\right)$$

$$= \lambda^{-N/2} Z(\boldsymbol{\theta}). \tag{A4}$$

In light of the above transformations, the dependence of NLL on λ takes the following explicit form

$$\mathrm{NLL}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}) = \frac{\tilde{H}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}_{-\lambda})}{\lambda} + \frac{N}{2} \ln \lambda + \ln \tilde{Z}(\boldsymbol{\theta}_{-\lambda}).$$

Hence, by minimizing NLL with respect to λ, i.e., $\frac{d\mathrm{NLL}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta})}{d\lambda} = 0$, we obtain the following expression for the optimal λ:

$$\lambda^* = \frac{2\tilde{H}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}_{-\lambda})}{N}. \tag{A5}$$

From the Gaussian joint pdf (8) it follows that

$$\tilde{Z}(\boldsymbol{\theta}_{-\lambda}) = (2\pi)^{N/2} \left\{ \det\left[\tilde{J}(\boldsymbol{\theta}_{-\lambda})\right] \right\}^{-1/2},$$

where $\tilde{J}(\boldsymbol{\theta}_{-\lambda}) = \lambda J(\boldsymbol{\theta})$. We insert the optimal value λ^* in NLL and use the expression above for the log-partition function which leads to

$$\mathrm{NLL}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}_{-\lambda}) = \frac{N}{2} \ln\left(\frac{2\tilde{H}(\mathbf{x}_{\mathrm{S}}; \boldsymbol{\theta}_{-\lambda})}{N}\right) + \frac{N}{2} \ln(2\pi) - \frac{1}{2} \det\left[\tilde{J}(\boldsymbol{\theta}_{-\lambda})\right]. \tag{A6}$$

The NLL (A6) is minimized numerically using the MATLAB constrained minimization function `fmincon`. Constraints are used to ensure that the parameter values are positive. The log-determinant is calculated numerically using the singular value decomposition of the precision matrix. This is a procedure with numerical complexity $O(N^3)$ for a full rank matrix. For this reason, we use cross validation instead of maximum likelihood for parameter inference.