Computational Statistics and Data Analysis 130 (2019) 80–93

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

# Latent Gaussian random field mixture models

David Bolin [a], Jonas Wallin [b,*], Finn Lindgren [c]

[a] Chalmers University of Technology and the University of Gothenburg, Gothenburg, Sweden
[b] Lund University, Tycho Brahes väg 1, 220 07 Lund, Sweden
[c] University of Edinburgh, Edinburgh, UK

ARTICLE INFO

*Article history:*
Received 9 November 2017
Received in revised form 5 August 2018
Accepted 10 August 2018
Available online 5 September 2018

*Keywords:*
Random field
Spatial statistics
Gaussian mixture
Stochastic gradient
Geostatistics
Gaussian process

ABSTRACT

For many problems in geostatistics, land cover classification, and brain imaging the classical Gaussian process models are unsuitable due to sudden, discontinuous, changes in the data. To handle data of this type, we introduce a new model class that combines discrete Markov random fields (MRFs) with Gaussian Markov random fields. The model is defined as a mixture of several, possibly multivariate, Gaussian Markov random fields. For each spatial location, the discrete MRF determines which of the Gaussian fields in the mixture that is observed. This allows for the desired discontinuous changes of the latent processes, and also gives a probabilistic representation of where the changes occur spatially. By combining stochastic gradient minimization with sparse matrix techniques we obtain computationally efficient methods for both likelihood-based parameter estimation and spatial interpolation. The model is compared to Gaussian models and standard MRF models using simulated data and in application to upscaling of soil permeability data.



## 1. Introduction

In spatial statistics, data are often linked to spatially varying discrete covariates such as land cover categories in vegetation models (Bolin et al., 2009), geology in soil permeability models (Kim et al., 2005), or brain tissue type in brain imaging applications (Hildeman et al., 2017b). These covariates cause discontinuities in the data that easily can be accounted for *if* one has access to the covariates. Unfortunately, these covariates are often unknown. In this scenario a standard Gaussian random field model is not suitable, due to its inability of handling discontinuities. Here we introduce a class of models, based on a combination of mixture models and Gaussian random fields, to handle this type of data.

One way to analyze data with missing discrete covariates is to first classify the data into the different distinct spatial regions and then model each region separately. In many applications ranging from video surveillance to speaker identification and image analysis (Reynolds and Rose, 1995; Stauffer and Grimson, 1999), Gaussian mixture models (GMMs) are used for the classification problem. A GMM assumes independence between the observations and that the distribution of each observation is $\pi(\mathbf{y}) = \sum_{k=1}^{K} w_k \pi_k(\mathbf{y})$, where $K$ is the number of classes, $w_k$ is the probability of class $k$, and $\pi_k$ a multivariate normal density. The assumption of independence between the observations is a clear drawback with GMM-based classification for spatial data. A strategy to account for spatial dependency is to allow for dependency in the allocation variables, which can be done in several ways. One way is to model the class probabilities using a logistic regression model based on Gaussian fields (Fernández and Green, 2002). Another way is to note that a random variable $\mathbf{Y}$ with a GMM distribution can be written as $\mathbf{Y} = \sum_{k=1}^{K} z_k \mathbf{X}_k$. Here $\mathbf{X}_k$ is a Gaussian random variable with density $\pi_k$, and $z_k = \mathbb{I}(\tilde{z} = k)$ where $\tilde{z}$ is a discrete random variable with $\mathrm{P}(\tilde{z} = k) = w_k$. Spatial dependency can be introduced by modeling the collection

* Corresponding author.
*E-mail address:* jonas.wallin@stat.lu.se (J. Wallin).

https://doi.org/10.1016/j.csda.2018.08.007
0167-9473/

of the random variables $\tilde{z}$ for all observations as a discrete Markov random field (MRF) (see e.g. Held et al., 1997; Zhang et al., 2001; Van Leemput et al., 1999), which we refer to as an MRF mixture model.

Allowing for spatial dependency in the mixture weights often improves the classification for spatial problems. Yet, it is often not sufficient since it cannot capture the dependence between observations within each class. To account for this, we replace the independent Gaussian variables $\mathbf{X}_k$ for each class by a spatially dependent Gaussian random field (see e.g. Cressie, 1991; Cressie and Wikle, 2011). This allows us to use the model for classification, but also for noise reduction and spatial interpolation in cases where the data consist of noisy partial observations of fields with discontinuities. We refer to models of this type, which are introduced in more detail in Section 2, as latent Gaussian random field mixture (LGFM) models.

The proposed model could be viewed as a non-stationary Gaussian random field, with a specific prior on spatially varying parameters. There is an extensive literature on non-stationary Gaussian fields, see for example Paciorek and Schervish (2006), Fuglstad et al. (2015), Higdon (2001) and Bolin and Lindgren (2011). A non-stationary Gaussian field that resembles the LGFM model is that of Fuentes and Smith (2001), where a process is created as a spatially varying average of stationary Gaussian processes. Other similar modeling approaches are those of Kim et al. (2005), where a tessellation of the spatial domain is used to define a mixture process, and the Bayesian treed Gaussian process models by Gramacy and Lee (2008). However, all these methods either lack the sharp and flexible discontinuities, or the computational efficiency, of the LGFM model.

Since spatial problems often have massive amounts of data, a computationally efficient estimation method is needed in order to fit the LGFM model to data. Further, likelihood estimation for discrete MRFs is problematic due intractable normalizing constants. Two common methods for dealing with this issue are gradient-based minimization and pseudo-likelihood methods (Guyon, 1995; Hildeman et al., 2017b). Recently, gradient-based methods have also been developed for large-scale Gaussian random field models (Anitescu et al., 2012; Stein et al., 2013). We combine these two approaches into a computationally efficient estimation method for LGFM models. The method is a stochastic version of the EM gradient method (Lange, 1995), and is introduced further in Section 3. The model is tested on two simulated data sets in Section 4, and on an application to upscaling soil permeability data in Section 5. Finally, Section 6 contains a discussion of possible extensions and further work. The code used to obtain the results in the article is available at https://bitbucket.org/davidbolin/lgfm/.

## 2. Latent Gaussian random field mixture models

Let $\mathbf{Y}_1, \ldots, \mathbf{Y}_M$ be $d$-dimensional observations at locations $\mathbf{s}_1, \ldots, \mathbf{s}_M$ on a regular lattice with $n$ nodes. The structure of a LGFM model for this data is

$$\begin{aligned} \mathbf{X}_k(\mathbf{s}) &= \mathbf{B}_k(\mathbf{s})\boldsymbol{\beta}_k + \boldsymbol{\xi}_k(\mathbf{s}), \quad k = 1, \ldots, K, \\ \mathbf{X}(\mathbf{s}) &= \sum_{k=1}^{K} z_k(\mathbf{s})\mathbf{X}_k(\mathbf{s}), \\ \mathbf{Y}_m &= \mathbf{X}(\mathbf{s}_m) + \boldsymbol{\varepsilon}_m, \quad m = 1, \ldots, M. \end{aligned} \tag{1}$$

Here $\boldsymbol{\varepsilon}_i$ are independent $\mathsf{N}(\mathbf{0}, \boldsymbol{\Sigma}_\varepsilon)$ random variables representing measurement noise for each dimension, with $\boldsymbol{\Sigma}_\varepsilon = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$, and the latent process $\mathbf{X}(\mathbf{s})$ is modeled as a mixture of $K$ independent Gaussian random fields $\mathbf{X}_k(\mathbf{s})$. These Gaussian fields are specified using the mean-zero Gaussian fields $\boldsymbol{\xi}_k(\mathbf{s})$ as well as regressions $\mathbf{B}_k(\mathbf{s})\boldsymbol{\beta}_k = \sum_{p=1}^{P}\mathbf{B}_{kp}(\mathbf{s})\beta_{kp}$ on fixed-effects $\mathbf{B}_{kp}(\mathbf{s})$ for the mean values. Finally, $z_k(\mathbf{s}) = \mathbb{I}(\tilde{z}(\mathbf{s}) = k)$ where $\tilde{z}(\mathbf{s})$ is a discrete MRF. In the following two sections, we introduce the statistical models for the discrete MRF and the Gaussian fields $\boldsymbol{\xi}_k(\mathbf{s})$ in more detail, and then discuss properties of the model.

### 2.1. A model for the Gaussian fields $\boldsymbol{\xi}_k(\mathbf{s})$

In the case of multivariate data, we assume that the Gaussian fields $\boldsymbol{\xi}_k(\mathbf{s})$ have proportional correlation models (Chiles and Delfiner, 1999), which means that their covariance functions can be written as $C(\boldsymbol{\xi}_k(\mathbf{s}_1), \boldsymbol{\xi}_k(\mathbf{s}_2)) = \boldsymbol{\Sigma}_k\rho_k(\|\mathbf{s}_1 - \mathbf{s}_2\|)$, where $\boldsymbol{\Sigma}_k$ is a $d \times d$ covariance matrix and $\rho_k(\cdot)$ is a spatial correlation function. The reason for this particular choice is that it makes the model a natural extension of the regular Gaussian mixture models, which have covariances $C(\boldsymbol{\xi}_k(\mathbf{s}_1), \boldsymbol{\xi}_k(\mathbf{s}_2)) = \boldsymbol{\Sigma}_k\delta_{\mathbf{0}}(\mathbf{s}_1 - \mathbf{s}_2)$, where $\delta_{\mathbf{0}}$ is a regular Dirac distribution.

What remains is to decide on a model for the spatial correlation function. A popular choice is the Matérn correlation function, $\rho(\mathbf{h}) = 2^{1-\nu}\Gamma(\nu)^{-1}(\kappa\|\mathbf{h}\|)^\nu K_\nu(\kappa\|\mathbf{h}\|)$, where $\Gamma$ is the gamma function and $K_\nu$ is a modified Bessel function of the second kind. The positive parameters $\kappa$ and $\nu$ determine the practical correlation range and the differentiability of the process, respectively. An advantage with this covariance function is that one then can use the stochastic partial differential equation (SPDE) connection (Lindgren et al., 2011) between Gaussian Matérn fields and Gaussian Markov random field models (Besag, 1974) to construct a model for $\boldsymbol{\xi}_k(\mathbf{s})$ that has important computational advantages.

Since we assume that the data are on a lattice, we do not need the full generality of the SPDE approach. We can instead use that a conditional autoregressive model of order $p \in \mathbb{N}$, a CAR(p) model, on a lattice in $\mathbb{R}^2$ can be viewed as an approximation of a Gaussian field with a Matérn covariance function with $\nu = p - 1$. The CAR(1) model (which could be viewed as an

approximation of the Matérn model for the limiting case $\nu \to 0$) has a sparse precision matrix (inverse covariance matrix) $\mathbf{Q} = \mathbf{G} + \kappa^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix and $\mathbf{G}$ is the Laplacian matrix for the lattice, with elements (for the interior nodes of the lattice)

$$G_{ij} = \begin{cases} 4 & \text{if } i = j, \\ -1 & \text{if } i \text{ is adjacent to } j \text{ in the lattice,} \\ 0 & \text{otherwise.} \end{cases}$$

Here $j$ is adjacent to $i$ if it represents a node in the lattice that is directly left, right, above, or below the $i$th node. Precision matrices for higher-order CAR models are obtained by multiplying the matrix $\mathbf{G} + \kappa^2\mathbf{I}$ with itself $p$ times (Lindgren et al., 2011). From now on, we let $\boldsymbol{\xi}_k$ denote the vector of $\xi_k(\mathbf{s})$ evaluated at the lattice locations, which thus has a $\mathsf{N}(\mathbf{0}, \boldsymbol{\Sigma}_k \otimes \mathbf{Q}_{\kappa_k}^{-1})$ distribution, where $\mathbf{Q}_{\kappa_k}$ is the precision matrix for a CAR(p) model with parameter $\kappa_k$.

### 2.2. A model for the discrete Markov random field $\tilde{z}(\mathbf{s})$

To obtain a model that can be viewed as a direct extension of the MRF mixture models, we let $\tilde{z}(\mathbf{s})$ be a discrete MRF defined on the lattice, taking values in $\{1, \ldots, K\}$. The joint distribution of $\tilde{\mathbf{z}} = (\tilde{z}_1, \ldots, \tilde{z}_n)^T$ is $p(\tilde{\mathbf{z}}) = Z^{-1}\exp(-W(\tilde{\mathbf{z}}))$ where $W(\tilde{\mathbf{z}}) = \sum_{\mathcal{C}} V_C(\tilde{\mathbf{z}})$ is the sum of the potential for all cliques generated by the neighborhood structures of the lattice and $Z$ is a normalizing constant (Winkler, 2003). In the applications later, we use first-order neighborhoods such that a node in the lattice has the four closest other nodes as neighbors (the same neighborhood as was used when defining the Laplacian matrix $\mathbf{G}$). There are then only first and second-order cliques, and we use the potentials $V_i(\tilde{\mathbf{z}}) = \alpha_k$ when $\tilde{z}_i = k$, and $V_{\{i,j\}}(\tilde{\mathbf{z}}) = \gamma$ when $\tilde{z}_i = \tilde{z}_j$ and $V_{\{i,j\}}(\tilde{\mathbf{z}}) = 0$ otherwise. Hence, the model has parameters $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ and $\gamma$ where $\boldsymbol{\alpha}$ determines the prior probabilities of the classes and $\gamma$ governs the strength of the spatial dependency.

The model can easily be extended by considering more complicated neighborhood structures or more complicated models for the interaction potentials, or simplified by assuming that all $\boldsymbol{\alpha} = \mathbf{0}$ (which results in a model that is equivalent to the Ising model for $K = 2$).

### 2.3. Model properties

The LGFM model contains GMMs, MRF mixture models, and latent Gaussian models as special cases. The MRF mixture model is obtained by letting the correlation range of $\rho_k(\mathbf{h})$ go to zero (or equivalently letting $\kappa$ go to infinity for the CAR model). Further, setting $\gamma = 0$ results in a standard GMM, and setting $K = 1$ results in a latent Gaussian model.

Let $\boldsymbol{\Psi}$ denote the set of all parameters. The expectation of the latent process is $\mathsf{E}[\mathbf{X}(\mathbf{s})|\boldsymbol{\Psi}] = \sum_{k=1}^{K}\mathsf{P}[z_k(\mathbf{s}) = 1]\mathbf{B}_k(\mathbf{s})\boldsymbol{\beta}_k$, and the covariance between $\mathbf{X}(\mathbf{s}_1)$ and $\mathbf{X}(\mathbf{s}_2)$ is

$$\mathsf{C}[\mathbf{X}(\mathbf{s}_1), \mathbf{X}(\mathbf{s}_2)|\boldsymbol{\Psi}] = \sum_{k=1}^{K}\mathsf{P}[z_k(\mathbf{s}_1) = 1, z_k(\mathbf{s}_2) = 1]\boldsymbol{\Sigma}_k\rho_k(\|\mathbf{s}_1 - \mathbf{s}_2\|).$$

For most applications, one is also interested in posterior characteristics of the field given the data, $\mathbf{Y} = (\mathbf{Y}_1^\top, \ldots, \mathbf{Y}_M^\top)^\top$. For a given location $\mathbf{s}_0$, the posterior probabilities $\mathsf{E}(z_k(\mathbf{s}_0)|\mathbf{Y}, \boldsymbol{\Psi})$ can be used for classification and the posterior expectation $\mathsf{E}(\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \boldsymbol{\Psi})$ for prediction. The uncertainty of the prediction can be characterized by the posterior variance $\mathsf{V}(\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \boldsymbol{\Psi})$. To calculate these quantities, we use that

$$\begin{aligned} \mathsf{E}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \boldsymbol{\Psi}] &= \mathsf{E}[\mathsf{E}(\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi})|\mathbf{Y}, \boldsymbol{\Psi}], \\ \mathsf{V}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \boldsymbol{\Psi}] &= \mathsf{E}[\mathsf{V}(\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi})|\mathbf{Y}, \boldsymbol{\Psi}] + \mathsf{V}[\mathsf{E}(\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi})|\mathbf{Y}, \boldsymbol{\Psi}]. \end{aligned} \tag{2}$$

Since $\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi}$ is Gaussian, the expectations and variances with respect to $\mathbf{X}$ in (2) are known analytically. To derive these expressions we let $\mathcal{I}_k$ denote the set of indices (of the lattice nodes) for which there are observations and where the entries of $\tilde{\mathbf{z}}$ equals $k$, and define the corresponding matrix $\mathbf{A}_k = \mathbf{I}_{\mathcal{I}_k, 1:n}$ (an identity matrix where rows have been removed). We then have

$$\begin{aligned} \mathsf{E}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi}] &= \sum_{k=1}^{K} z_k(\mathbf{s}_0)\mathbf{B}_k(\mathbf{s}_0)\boldsymbol{\beta}_k + (\mathbf{I}_d \otimes \tilde{\mathbf{a}}_{\mathbf{s}_0})\hat{\mathbf{Q}}_k^{-1}(\boldsymbol{\Sigma}_\varepsilon^{-1} \otimes \mathbf{A}_k^\top)(\mathbf{Y} - \mathbf{B}_k\boldsymbol{\beta}_k), \\ \mathsf{V}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi}] &= \sum_{k=1}^{K} z_k(\mathbf{s}_0)(\mathbf{I}_d \otimes \tilde{\mathbf{a}}_{\mathbf{s}_0})\hat{\mathbf{Q}}_k^{-1}(\mathbf{I}_d \otimes \tilde{\mathbf{a}}_{\mathbf{s}_0})^\top, \end{aligned}$$

where $\hat{\mathbf{Q}}_k = \boldsymbol{\Sigma}_k^{-1} \otimes \mathbf{Q}_{\kappa_k} + \boldsymbol{\Sigma}_\varepsilon^{-1} \otimes \mathbf{A}_k^\top\mathbf{A}_k$ is the posterior precision matrix for $\boldsymbol{\xi}_k$, $\tilde{\mathbf{a}}_{\mathbf{s}_0}$ is a vector with a one at the position corresponding to the location $\mathbf{s}_0$ in the lattice and zero elsewhere, and $\mathbf{B}_k$ is a matrix with the covariates $\mathbf{B}_k(\mathbf{s})$ evaluated at the observation locations.

There are no closed form expressions for the expectations and variances with respect to $\tilde{\mathbf{z}}$ in (2), but these can be estimated using Monte Carlo integration as

$$\mathsf{E}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \Psi] \approx \frac{1}{J}\sum_{j=1}^{J} \mathsf{E}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}^{(j)}, \Psi] := \hat{\mathbf{X}},$$

$$\mathsf{V}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \Psi] \approx \frac{1}{J}\sum_{j=1}^{J} \mathsf{V}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}^{(j)}, \Psi] + \frac{1}{J}\sum_{j=1}^{J}(\mathsf{E}[\mathbf{X}(\mathbf{s}_0)|\mathbf{Y}, \tilde{\mathbf{z}}^{(j)}, \Psi] - \hat{\mathbf{X}})^2,$$

where $\tilde{\mathbf{z}}^{(j)}$ are draws from $\pi(\tilde{\mathbf{z}}|\mathbf{Y}, \Psi)$.

For regular MRF mixture models, where $\mathbf{Y}|\tilde{\mathbf{z}}, \Psi$ is a vector of independent variables, samples of $\tilde{\mathbf{z}}|\mathbf{Y}, \Psi$ are often obtained by a Gibbs sampler. The procedure is based on dividing the pixels into sets of conditionally independent nodes (in our case with first order neighborhoods two sets, $\tilde{\mathbf{z}}_w$ and $\tilde{\mathbf{z}}_b$, obtained as a checker pattern on the lattice), and then iteratively sampling from each of the sets conditioning on all others. See Winkler (2003) for details. A sampler for $\tilde{\mathbf{z}}|\mathbf{Y}, \Psi$ for the LGFM model can be constructed by introducing an additional step in this Gibbs sampler as follows. First initiate $\tilde{\mathbf{z}}^{(0)}$ and repeat the following three steps for $j = 1, \ldots, J$.

1. Sample the Gaussian fields $\{\boldsymbol{\xi}_k\}^{(j)}$ from their respective distributions $\pi(\boldsymbol{\xi}_k|\mathbf{Y}, \tilde{\mathbf{z}}^{(j-1)}, \Psi)$.
2. Sample $\tilde{\mathbf{z}}_w^{(j)}$ from $\pi(\tilde{\mathbf{z}}_w|\mathbf{Y}, \tilde{\mathbf{z}}_b^{(j-1)}, \{\boldsymbol{\xi}_k\}^{(j)}, \Psi)$.
3. Sample $\tilde{\mathbf{z}}_b^{(j)}$ from $\pi(\tilde{\mathbf{z}}_b|\mathbf{Y}, \tilde{\mathbf{z}}_w^{(j)}, \{\boldsymbol{\xi}_k\}^{(j)}, \Psi)$ and set $\tilde{\mathbf{z}}^{(j)} = (\tilde{\mathbf{z}}_w^{(j)}, \tilde{\mathbf{z}}_b^{(j)})$.

Since $\mathbf{Y}|\{\boldsymbol{\xi}_k\}^{(j)}, \Psi$ is a vector of independent variables, the last two steps are performed in the same way as for the standard MRF mixture model.

For small problems, simulation from $\pi(\boldsymbol{\xi}_k|\mathbf{Y}, \tilde{\mathbf{z}}^{(j-1)}, \Psi)$ can be done using sparse Cholesky factorization of $\hat{\mathbf{Q}}_k = \boldsymbol{\Sigma}_k^{-1} \otimes \mathbf{Q}_{\kappa_k} + \boldsymbol{\Sigma}_\varepsilon^{-1} \otimes \mathbf{A}_k^\top \mathbf{A}_k$, where $\mathbf{A}_k$ is based on the $j$th sample of $\tilde{\mathbf{z}}$. A more computationally efficient method, that avoids computing the Cholesky factor of $\hat{\mathbf{Q}}_k$ is presented in Appendix A.2.

## 3. Parameter estimation

Parameter estimation for MRF mixture models is difficult, and allowing for spatial dependency within each class introduces further complications. Furthermore, we want these models to be applicable for massive multivariate problems, which are common in areas such as brain imaging, making computational efficiency of the estimation procedure paramount.

The MRF mixture models are typically either estimated with some modified version of the EM algorithm (Dempster et al., 1977) or through Markov chain Monte Carlo (MCMC) methods. Both of these procedures are too computationally demanding to be useful for the LGFM models. Instead, we base our estimation on the EM gradient (EMG) algorithm (Lange, 1995). The method is built on Fisher's identity

$$\nabla_\Psi \log L(\Psi; \mathbf{Y}) = \mathsf{E}_{\mathbf{X}}\left[\nabla_\Psi \log \pi(\mathbf{Y}, \mathbf{X}|\Psi)|\mathbf{Y}, \Psi\right],$$

and the fact that the gradient of the complete likelihood, $\nabla_\Psi \log \pi(\mathbf{Y}, \mathbf{X}|\Psi)$, often is explicit even when the likelihood, $L(\Psi; \mathbf{Y})$, is analytically intractable. Computing the expectation above is viewed as the E-step of the algorithm, and the following M-step then updates the parameters given the gradient in a gradient descent step. The $p$th iteration of the EMG algorithm goes as follows:

E-step: compute $\mathsf{E}_{\mathbf{X}}\left[\nabla_\Psi \log \pi(\mathbf{Y}, \mathbf{X}|\Psi^{(p)})|\mathbf{Y}, \Psi^{(p)}\right]$.
M-step: Set $\Psi^{(p+1)} = \Psi^{(p)} + \gamma^{(p)}\mathbf{S}\mathsf{E}_{\mathbf{X}}\left[\nabla_\Psi \log \pi(\mathbf{Y}, \mathbf{X}|\Psi^{(p)})|\mathbf{Y}, \Psi^{(p)}\right]$.

Here $\gamma^{(p)}$ is a non-negative step-size parameter. In order to ensure convergence to a local minimum one needs $\sum_{p=1}^{\infty}\gamma^{(p)} = \infty$ and $\sum_{p=1}^{\infty}\left(\gamma^{(p)}\right)^2 < \infty$, see Asmussen and Glynn (2007). Further, $\mathbf{S}$ is a scaling matrix. Taking $\mathbf{S}$ as an identity matrix results in an ordinary gradient descent method which has linear convergence. Ideally, we would like to take $\mathbf{S}$ as the inverse of the Hessian matrix $\mathbf{H}$ of the likelihood to obtain a Newton method with quadratic convergence. If it is not possible to compute the true Hessian of the log-likelihood, one can instead use

$$\mathbf{S}_{ij}^{-1} = -\mathsf{E}_{\mathbf{X}}\left[\frac{\partial^2}{\partial \Psi_i \, \partial \Psi_j} \log \pi(\mathbf{Y}, \mathbf{X}|\Psi)|\mathbf{Y}, \Psi\right]. \tag{3}$$

(Lange, 1995) showed that if $\mathbf{S}$ in (3) is used, then the convergence of the EMG algorithm is equivalent to that of the EM-algorithm (with latent variable $\mathbf{X}$).

For the LGFM models, we are not able to evaluate the expectation in the E-step analytically, and instead approximate it by the Monte Carlo estimate

$$\nabla_\Psi \log L(\Psi; \mathbf{Y}) = \mathsf{E}_{\mathbf{X}}\left[\nabla_\Psi \log \pi(\mathbf{Y}, \mathbf{X}|\Psi)|\mathbf{Y}, \Psi\right] \approx \frac{1}{J}\sum_{j=1}^{J} \nabla_\Psi \log \pi(\mathbf{Y}, \mathbf{x}^{(j)}|\Psi),$$

where $\mathbf{x}^{(j)}$ are draws from $\pi(\mathbf{X}|\mathbf{Y}, \Psi)$. We refer to this estimation procedure as the MCEMG algorithm.

Results of Andrieu et al. (2005) can be used to show that the algorithm converges to a stationary point under suitable constraints, when $\mathbf{x}^{(j)}$ are draws from a Monte Carlo algorithm or an MCMC chain with geometric ergodicity.

### 3.1. Estimation of the LGFM model

For the LGFM model, the parameters of interest are $\Psi = \{\Psi_z, \Psi_\xi\}$, here $\Psi_z = \{\boldsymbol{\alpha}, \gamma\}$ are the parameters for $\tilde{\mathbf{z}}$ and $\Psi_\xi = \{\sigma_1, \ldots, \sigma_d, \boldsymbol{\beta}_k, \boldsymbol{\Sigma}_k, \kappa_k, k = 1, \ldots, K\}$ all other parameters. To estimate the parameters using the MCEMG algorithm, we use both $\tilde{\mathbf{z}}$ and the Gaussian random fields $\boldsymbol{\xi} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_K\}$ as latent variables. By Fisher's identity and the law of total expectation it follows that

$$\nabla_\Psi \log L(\Psi; \mathbf{Y}) = \mathsf{E}_{\tilde{\mathbf{z}}}\Big\{\nabla_\Psi \log \pi(\tilde{\mathbf{z}}|\Psi_z) + \mathsf{E}_{\boldsymbol{\xi}}\big[\nabla_\Psi \log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}, \Psi_\xi)\big|\mathbf{Y}, \tilde{\mathbf{z}}, \Psi_\xi\big]\Big|\mathbf{Y}, \Psi\Big\}.$$

Thus the gradients for $\Psi_z$ and $\Psi_\xi$ can be updated separately, since

$$\nabla_{\Psi_z} \log L(\Psi; \mathbf{Y}) = \mathsf{E}_{\tilde{\mathbf{z}}}\Big[\nabla_{\Psi_z} \log \pi(\tilde{\mathbf{z}}|\Psi_z)\Big|\mathbf{Y}, \Psi\Big],$$
$$\nabla_{\Psi_\xi} \log L(\Psi; \mathbf{Y}) = \mathsf{E}_{\tilde{\mathbf{z}}}\Big\{\mathsf{E}_{\boldsymbol{\xi}}\big[\nabla_{\Psi_\xi} \log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}, \Psi_\xi)\big|\mathbf{Y}, \tilde{\mathbf{z}}, \Psi_\xi\big]\Big|\mathbf{Y}, \Psi\Big\}.$$

It is not possible to compute the gradient $\nabla_{\Psi_z} \log \pi(\tilde{\mathbf{z}}|\Psi_z)$ due to the normalizing constant of the density of $\tilde{\mathbf{z}}$. A standard method for handling this problem is to replace $\pi(\tilde{\mathbf{z}}|\Psi_z)$ by a pseudo-likelihood. Our approach for doing this approximation is identical to that of, for example, Hildeman et al. (2017b), and details are provided in the Appendix. It should be noted that convergence results for this modification are not established, to the authors knowledge, for the MRF parameters. However, the method has worked well for all applications we have tested it on and we later show that the parameters converge to reasonable values for simulated data.

To calculate $\nabla_{\Psi_\xi} \log L(\Psi; \mathbf{Y})$ we note that the expectation with respect to $\tilde{\mathbf{z}}$ is not explicit and must be approximated using MC sampling. Since the expectation with respect to $\boldsymbol{\xi}$ is known analytically, see the Appendix, we can use Rao-Blackwellization to estimate the gradient as

$$\nabla_{\Psi_\xi} \log L(\Psi; \mathbf{Y}) \approx \frac{1}{J}\sum_{j=1}^{J} \mathsf{E}_{\boldsymbol{\xi}}\left[\nabla_{\Psi_\xi} \log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}^{(j)}, \Psi_\xi)\big|\mathbf{Y}, \tilde{\mathbf{z}}^{(j)}, \Psi\right],$$

where $\tilde{\mathbf{z}}^{(j)}$ are samples from $\pi(\tilde{\mathbf{z}}|\mathbf{Y}, \Psi)$ obtained using the Gibbs sampler from Section 2.3. How to compute $\nabla_{\Psi_\xi} \log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}^{(j)}, \Psi_\xi)$ analytically in a computationally efficient way is presented in the Appendix. A typical convergence plot for the resulting algorithm can be seen in Fig. 9 in the Appendix.

## 4. Estimation and prediction on simulated data

In this section, we compare the predictive ability of the LGFM model against that of a standard latent Gaussian model on two simulated data sets. All results are obtained using Matlab a implementation of the method on a Macbook Pro computer with a 2.6 GHz Intel Core i7 processor.

### 4.1. Simulated LGFM data

The first simulated data set is used to test how well spatial prediction with the LGFM works when the data are from the correct model but with unknown parameters. We generate data from a univariate LGFM model with $K = 3$ and CAR(2) models for the spatial covariances. We assume that the Gaussian field for each class $k = 1, 2, 3$ has a constant mean $\beta_k = 2k$, a precision parameter $Q_k = \Sigma_k^{-1} = 2k$, and $\kappa = 0.1$. For the discrete MRF, we use $\gamma = 1$ and $\boldsymbol{\alpha} = \mathbf{0}$. We simulate the field on a $60 \times 100$ regular lattice (the same size as will be used for the application later), shown in Fig. 1. The observed data are constructed by randomly selecting 33% of the grid points and observing the process at these locations with Gaussian measurement noise with variance $\sigma^2 = 0.05$.

To test the model's sensitivity against the choice of $K$, we fit models with values of $K$ ranging from 1 (a standard Gaussian model) to 5 to the data. The parameters of the model are estimated based on the data using the proposed estimation method, using 1000 iterations (and $J = 5$ iterations for the Gibbs sampler). For each model we then compute the posterior mean of the latent field using the method from Section 2.3 with 1000 MC samples. Parameter tracks for the estimation with $K = 3$ can be seen in Fig. 9.

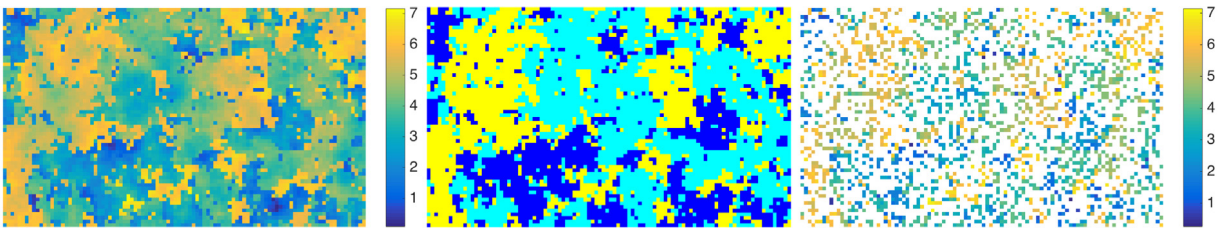

**Fig. 1.** A simulated LGFM model with $K = 3$ (left), its corresponding classification field (middle), and data generated by observing the field under Gaussian measurement noise at 33% of the locations (right).

**Table 1**
MAE and RMSE for the estimated kriging predictor compared to the true latent field, and the QIGN score, for the different models in the first simulated example. For each model, the computation times (in seconds) for obtaining a suitable starting value for the parameter estimation (Time init), the parameter estimation (Time estimation), and kriging (Time kriging) are also shown.

| | MAE | RMSE | QIGN | Time init | Time estimation | Time kriging |
|---|---|---|---|---|---|---|
| $K = 1$ | 0.792 | 1.115 | 0.541 | 0 | 73 | 0.6 |
| $K = 2$ | 0.571 | 0.947 | −0.246 | 90 | 1704 | 568 |
| $K = 3$ | 0.552 | 0.938 | −0.365 | 132 | 2459 | 835 |
| $K = 4$ | 0.547 | 0.940 | −0.374 | 196 | 3627 | 1108 |
| $K = 5$ | 0.548 | 0.941 | −0.342 | 258 | 5672 | 1912 |

To measure the accuracy of the predictions, we consider the mean absolute error (MAE) and root-mean squared error (RMSE) between the estimated kriging predictor (the posterior mean) and the true latent field. To assess the accuracy of the predicted uncertainty, we also use the mean quasi ignorance score (Gneiting and Raftery, 2007, Section 4.4), which is a proper scoring rule defined by

$$QIGN_i = \frac{(x_i - \hat{x}_i)^2}{2\hat{\sigma}_i} + \log(\hat{\sigma}_i),$$

where $x_i$ denotes the true field at lattice point $i$, $\hat{x}_i$ the kriging prediction, and $\hat{\sigma}_i$ the kriging standard deviation. The resulting values can be seen in Table 1. One can note that the model is quite insensitive to the choice of $K$ as long as $K > 1$, and that the Gaussian model ($K = 1$) performs poorly compared to the other models.

The table also shows the required computation times for the analysis, which are divided into three steps: Time init is the time required to obtain reasonable starting values for the optimization. For $K = 1$ the starting values are manually set based on the size of the domain and the variance of the data, whereas for $K > 1$ an initial classification of the domain is performed by estimating a standard MRF mixture model. Given this classification, Gaussian models are then separately estimated for each class to obtain starting values for the Gaussian parameters. Time kriging shows the computation time required for obtaining the kriging prediction and its variance. The majority of the time in this step is spent on calculating the kriging variance, since this requires computing the diagonal of the inverse of the posterior precision matrix. To perform this step, we used the method based on the Takahashi equations (Takahashi et al., 1973). To reduce the computation time of this step, one alternative would be to use an iterative method such as that of Sidén et al. (2018).

The resulting kriging predictions for the Gaussian model and the LGFM model with $K = 3$ are shown in Fig. 2. One can see that the Gaussian model handles the discontinuities in the field poorly. Not surprisingly, the LGFM model performs much better, and also has a much more accurate estimate of the uncertainty.

### *4.2. Simulated Gaussian field with discontinuous mean*

The second data set is created to study the effect of missing an important covariate for a Gaussian model. We generate data from a bivariate model $\mathbf{Y}_i = \boldsymbol{\mu}(\mathbf{s}_i) + \boldsymbol{\xi}(\mathbf{s}_i) + \boldsymbol{\epsilon}_i$ where $\boldsymbol{\xi}(\mathbf{s})$ is a bivariate Gaussian field as in Section 2.1, with a spatial CAR(1) model and multivariate dependence structure given by

$$\boldsymbol{\Sigma} = \frac{1}{3}\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

The model is defined on the same $60 \times 100$ lattice as in the previous example and has a discontinuous mean value function $\boldsymbol{\mu}(\mathbf{s}) = (\mu_1(\mathbf{s}), \mu_2(\mathbf{s}))^\top$ where $\mu_d(\mathbf{s}) = 2(\mathbb{I}(s_1 > 32) + \mathbb{I}(s_1 > 66))$ for $d = 1,2$.

Data are generated by observing the field at 20% of the locations under Gaussian measurement noise with variance $\sigma^2 = 0.05$. The simulated field and the observations are shown in Fig. 3. Based on these data, we estimate LGFM models with constant mean values for each class, using $K = 1, \ldots, 5$ to test the sensitivity of the model against the choice of $K$. As for the previous example, we run the estimation algorithm for 1000 iterations and use 5 iterations for the Gibbs sampler.

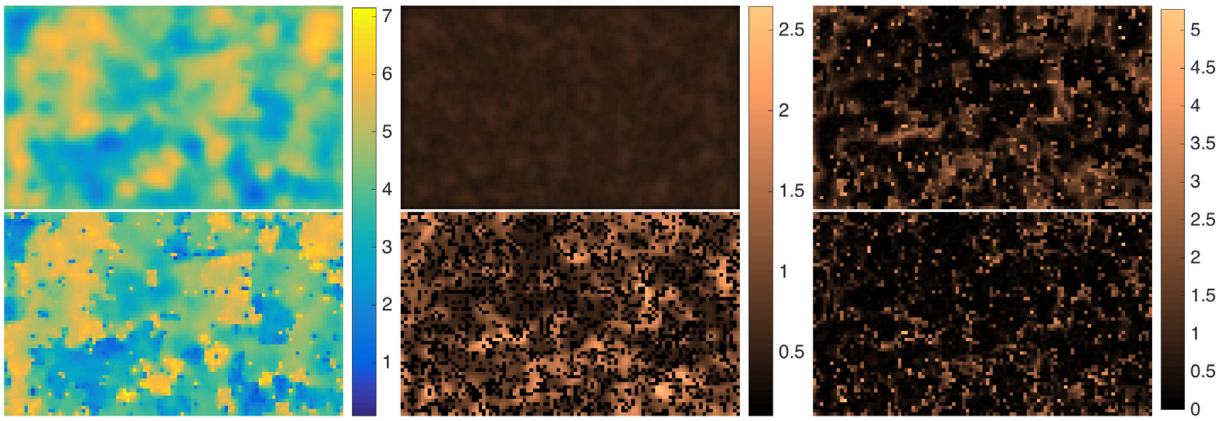


**Fig. 2.** Results for the Gaussian model (top) and the LGFM model with $K = 3$ (bottom). The panels show the kriging prediction (left) in the color scale used for Fig. 1(left), kriging standard deviation (middle), and the absolute difference between the kriging estimate and the true latent field (right).

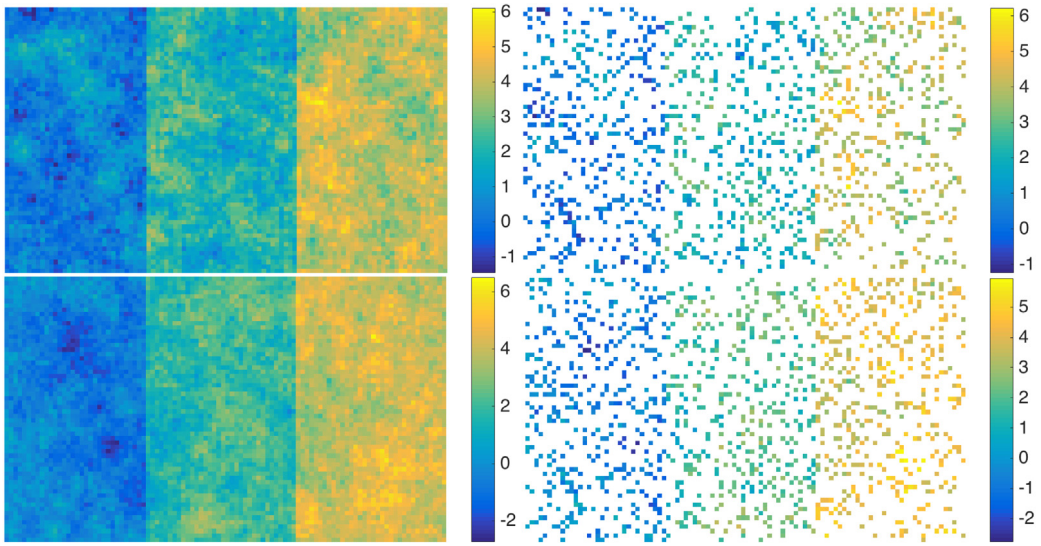


**Fig. 3.** A simulated bivariate Gaussian field with a discontinuous mean, where the first dimension is shown in the top left panel, and the second in the bottom left panel. The observations of the two dimensions are shown in the right panels.

**Table 2**
MAE and RMSE for the estimated kriging predictor compared to the true latent field, and the QIGN score, for the different models in the second simulated example. For each model, the computation times (in seconds) for obtaining a suitable starting value for the parameter estimation (Time init), the parameter estimation (Time estimation), and kriging (Time kriging) are also shown.

| | MAE | RMSE | QIGN | Time init | Time estimation | Time kriging |
|---|---|---|---|---|---|---|
| $K = 1$ | 0.421 | 0.552 | −0.2349 | 0 | 351 | 0.26 |
| $K = 2$ | 0.361 | 0.471 | −0.4673 | 58 | 1141 | 591 |
| $K = 3$ | 0.345 | 0.446 | −0.5265 | 95 | 1847 | 904 |
| $K = 4$ | 0.345 | 0.445 | −0.5544 | 115 | 2284 | 1191 |
| $K = 5$ | 0.386 | 0.497 | −0.5754 | 156 | 2390 | 1314 |

We compute the posterior mean of the latent field for each model and compare against the true simulated field using MAE, RMSE, and the QIGN score. The results can be seen in Table 2.

As for the previous example, the table also shows the computation times. One can note that the computation times are smaller for this example even though it is a bivariate problem. The main reason for this is that CAR(1) models are used instead of CAR(2) models, which results in sparser precision matrices for the Gaussian fields. Another reason for why increasing the dimension of the fields does not affect the estimation time as much as one would expect is that the proportional correlation

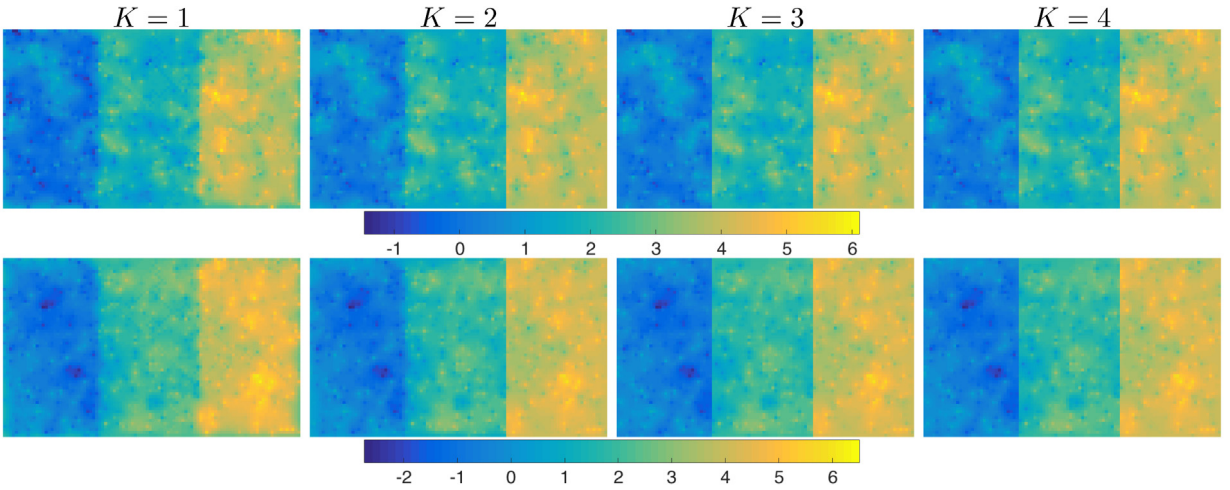


**Fig. 4.** Kriging predictions for the LGFM models for different values of $K$ for the second simulated example. The predictions of the first dimension are shown in the top row, and the predictions of the second dimension are shown in the bottom row. Note that for $K = 1$, both lines of discontinuity are smoothed. The model correctly finds the right-hand line for $K = 2$, and both lines for $K = 3$ and $K = 4$.

structure of the latent models can be used to efficiently calculate the required gradients. Details of these computations is given in the Appendix.

Fig. 4 shows the kriging predictions for the models with $K = 1, \ldots, 4$. As can be seen in the figure, the model correctly finds the points of discontinuity if $K \geq 3$, but is forced to disregard one of the discontinuities when $K = 2$. The reason for the very similar results for the models with $K \geq 3$ in Table 2 is that the model basically does not use the extra classes in these cases, which can be seen in Fig. 4 for $K = 4$, where no additional points of discontinuities are introduced. Thus, also in this case, the model is fairly stable against the choice of $K$.

Even though this example was constructed to show the strengths of the method, it should be noted that the example did not use data simulated from the LGFM model that we estimated, but a standard latent Gaussian model with a discontinuous mean value function. Thus, the method may be a good alternative also for cases where a latent Gaussian model is appropriate but where one suspects that the mean value has points of discontinuity which cannot be explained by available covariates.

## 5. Upscaling of soil permeability data

In this section we present an application where the LGFM model is compared with standard latent Gaussian models and MRF mixture models. The data comes from the Tenth SPE Comparative Solution Project (http://www.spe.org/web/csp/), and was originally used to compare different upscaling approaches to predict the performance of a waterflood in a black oil reservoir described by a fine-scale Cartesian geological model. The data set consists of three-dimensional porosity and (horizontal and vertical) permeability data from a geological model. Fig. 5 shows the (log) horizontal permeability and porosity data for one horizontal slice of the three-dimensional structure.

We start by considering the problem of upscaling of the permeability data. We use the horizontal slice shown in Fig. 5 to create a lower-resolution data set by removing every second measurement. The aim is to use these data to compare how well different models can predict the removed values.

We first use the LGFM model with $K = 1, \ldots, 10$, where the latent Gaussian fields $\xi_k(\mathbf{s})$ are modeled as CAR(1) processes and the discrete MRF has $\boldsymbol{\alpha} = \mathbf{0}$. For $K = 1$, the model thus is a latent Gaussian CAR(1) model. As benchmarks, we also fit corresponding MRF mixture models with $K = 2, \ldots, 10$. Recall that these models can be obtained from the LGFM models by letting the correlation spatial range go to zero. The reason for not using $K = 1$ for the MRF model is that this would result in a model without any spatial dependence, which therefore is inappropriate for spatial prediction.

The parameters of the different models are estimated using the stochastic gradient method from Section 3, with the same number of iterations as for the simulated data examples in Section 4. For each model, we then predict the values of the removed pixels using the posterior mean given the observed pixels, and compute the posterior variances as measures of uncertainty.

To assess the accuracy of the models, we use MAE, RMSE, and the QIGN score as for the simulated examples. The results are shown in Fig. 6. One can note that the LGFM models outperform both the MRF mixture models and the Gaussian model, and that the results are quite stable against the choice of $K$. Fig. 7 shows the reconstructed images together with their uncertainty estimates as well as the absolute errors for the Gaussian model, the LGFM model with $K = 7$, and the MRF model with $K = 7$. One can note that LGFM reconstruction is more similar to the true field compared with the other models, and that it has a much more accurate estimate of the prediction variance compared to the Gaussian model. It should be mentioned that we

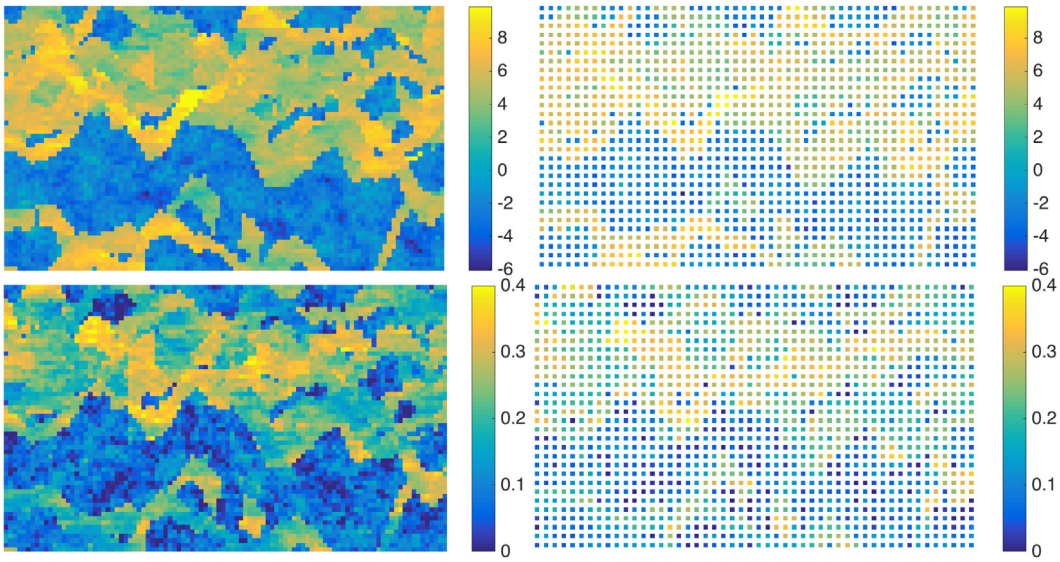


**Fig. 5.** Log horizontal permeability (top left) and porosity (bottom left), as well as observations obtained by taking every second value for permeability (top right) and porosity (bottom right).

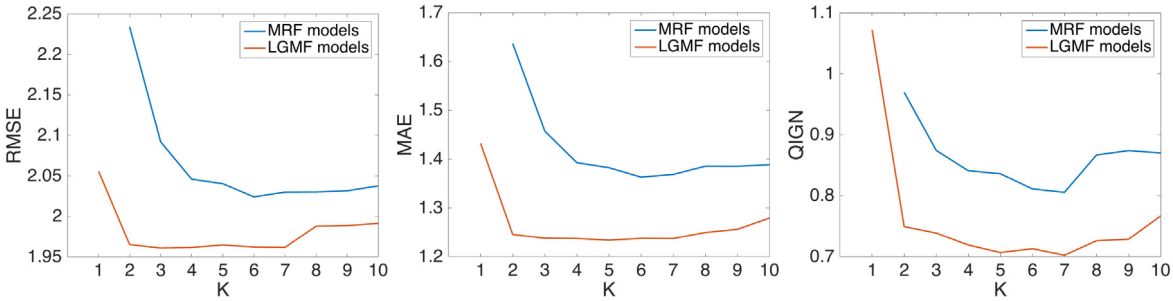


**Fig. 6.** RMSE (left), MAE (middle), and QIGN scores (right) of the different models for the permeability data.

also tested using CAR(2) models in the LGFM (and Gaussian) models, but that this did not improve the results. We therefore do not include these results. Since the size of this data set is the same as for the second simulated example, the computation times are similar to those in Table 2.

To investigate the multivariate aspects of the LGFM models, we now consider the bivariate data set of both permeability and porosity. We use the LGFM model with $K = 1, \ldots, 10$, where the latent Gaussian fields $\boldsymbol{\xi}_k$ have proportional correlation models as described in Section 2.1, where the spatial covariance is given by a CAR(1) model, and the discrete MRF has $\boldsymbol{\alpha} = \mathbf{0}$. Also for these data, we fit the corresponding MRF mixture models with $K = 1, \ldots, 10$ as benchmarks.

As for the univariate data, we predict the values of the removed pixels using the posterior mean given the observed pixels, and use MAE, RMSE, and the QIGN score to assess the accuracy. The results are shown in Fig. 8. Also for the bivariate data, one can note that the LGFM models outperform both the MRF mixture models and the Gaussian model, and that the results are quite stable against the choice of $K$.

As seen in Fig. 8 there is no major improvement in the porosity prediction when the multivariate data set is used. Two things should be noted about this: First, model parameters are not optimized to predict the permeability only, and thus some improvement could be achieved by using this as target for the parameter estimation. Second, the spatial dependency introduced for the multivariate model is somewhat simplistic. In fact, if there were no measurement error the porosity observations would only contribute to the classification (the prediction of $\tilde{\mathbf{z}}$) but not to the prediction of the permeability values given $\tilde{\mathbf{z}}$.

## 6. Discussion

This work has introduced the class of LGFM models as well as a computationally efficient stochastic gradient parameter estimation method for the model class. The model is a mix of latent Gaussian processes and a MRF mixture model, and

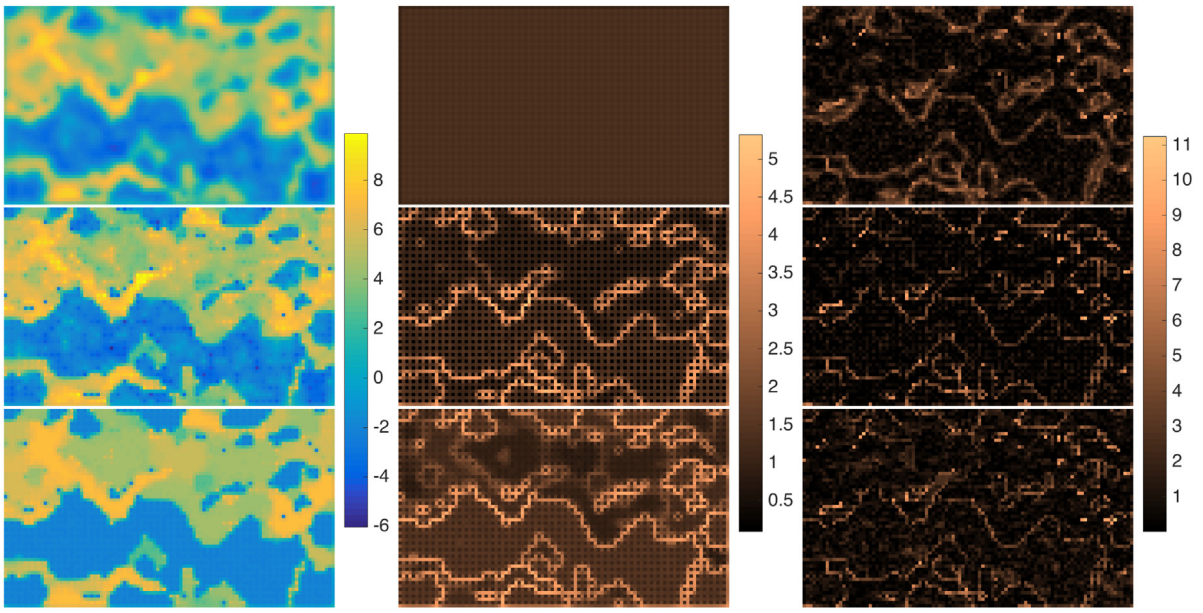


**Fig. 7.** Results for the Gaussian (top row), LGMF (middle row), and MRF (bottom row) models. The left column shows the kriging estimates in the color scale used for the data in Fig. 5. The middle column shows the prediction standard deviations, and the right column shows the absolute differences between the predictions and the true permeability data.

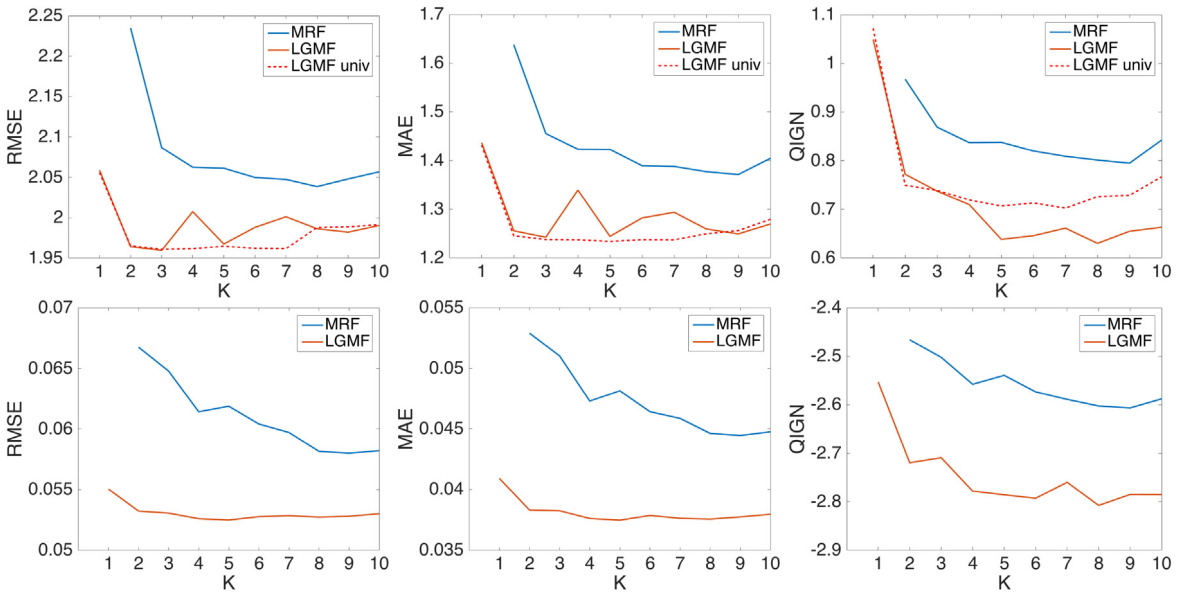


**Fig. 8.** Results for the bivariate data where permeability is shown in the top row and porosity in the bottom row. The RMSE of the kriging predictions is shown to the left, the MAE in the middle, and the QIGN scores to the left. The dashed lines show the results for the univariate data from Fig. 6.

therefore has the ability to model non-stationary data with both sharp discontinuities and spatially varying covariance parameters. These features were found to be useful for the application to upscaling of soil permeability data. The mixture aspect of the model allowed for more accurate uncertainty quantification of the upscaling compared with a standard latent Gaussian model, and the spatial dependency within each region enabled the model to outperform standard MRF mixture models for prediction.

The LGFM model preforms well when there are sharp spatial variations in the parameters of the latent field. A major difference between the model and other non-stationary models, like Fuglstad et al. (2015), is that the LGFM model can allow for sudden changes in the parameters without any prior knowledge of these changes. Another difference is that, conditionally

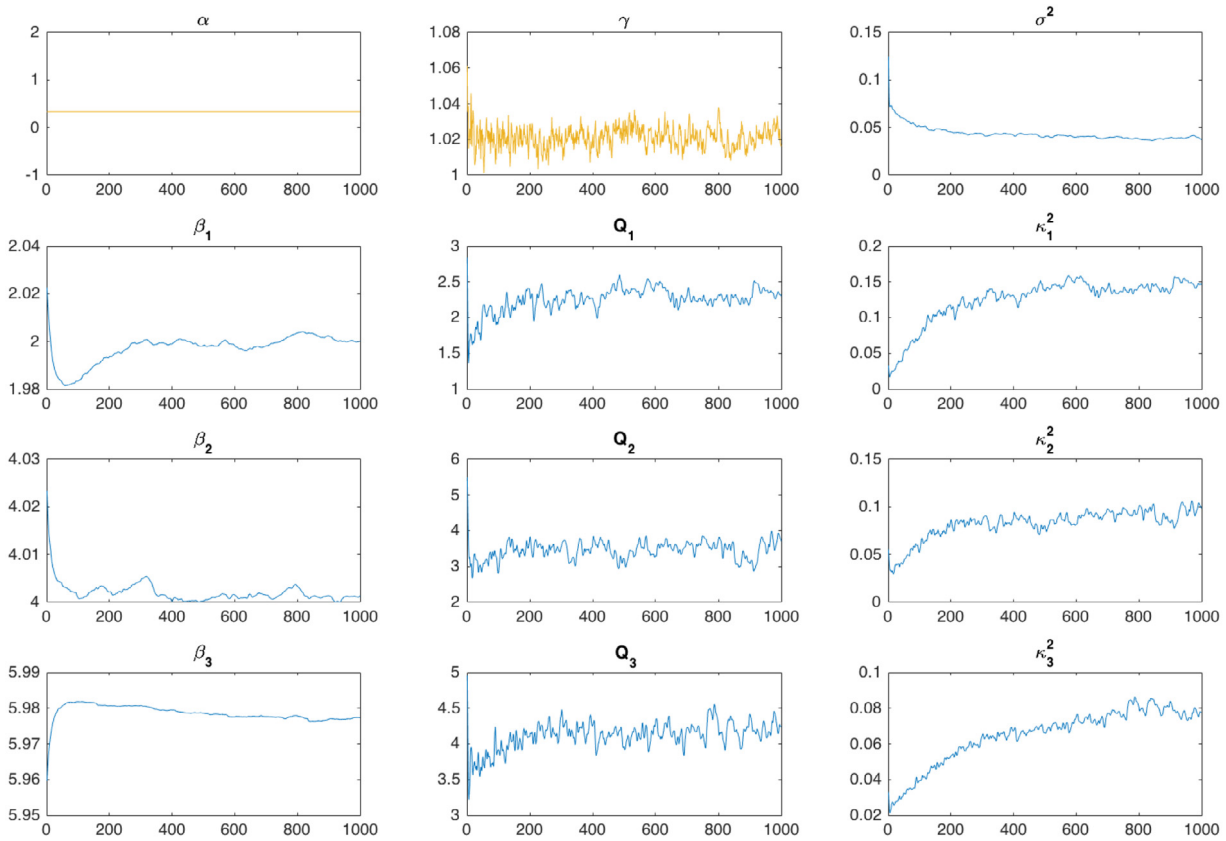


**Fig. 9.** Trace plot of the parameter estimates for the LGFM model with $K = 3$ for the simulated data in Section 4.1.

on the MRF part of the model, the Gaussian fields in the different regions are assumed to be independent. This is easy to relax in theory, but the computational efficiency of the proposed estimation method is highly dependent on this assumption. However, the model may be a good alternative also for situations where the independence assumption is not satisfied. An example of this was shown in one of the examples using simulated data, where data from a stationary latent Gaussian model with a missing discontinuous covariate were analyzed.

As a next step, it may be of interest to compare the performance of the model to other similar models, such as that of Kim et al. (2005). Since the model is specified using a discrete MRF for the class indicators, the model is most suitable for applications on discrete domains. For more traditional geostatistical applications on continuous domains, one could use a high-resolution lattice over the domain to define the MRF. An alternative approach, which we would like to investigate further in future research, is to replace the MRF with a truncated Gaussian field, similar to what was developed by Dunlop et al. (2017) and Hildeman et al. (2017a). In this case, replacing the CAR models for the Gaussian random fields by Gaussian Matérn fields on the continuous domain will not increase the computational cost as long as they are specified using the SPDE approach by Lindgren et al. (2011).

## Acknowledgments

This research has been supported by the Knut and Alice Wallenberg foundation, Sweden (KAW 20012.0067) and the Swedish Research Council (grant 2016-04187). We would like to thank the two anonymous reviewers for their valuable comments which greatly improved the article.

## Appendix. Details about estimation procedure

In this appendix, we provide details for how to compute the gradients needed for the parameter estimation procedure.

### *A.1. The gradient for the MRF parameters*

Since we cannot evaluate $\pi(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z)$ due to its intractable normalizing constant, we use the common solution to replace it with a pseudo-likelihood, $\pi_p(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z)$, which is a product of the full conditionals of $\tilde{\mathbf{z}}$. Recall that $\mathcal{N}_i$ are the neighbors of

the $i$th node in the lattice and let $f_{ik} = \sum_{j\in\mathcal{N}_i} z_{jk}$. The conditional class probabilities for the $i$th node can then be written as $\mathsf{P}(\tilde{z}_i = k|f_{ik}, \boldsymbol{\Psi}_z) = \mathsf{E}(z_{ik}|f_{ik}, \boldsymbol{\Psi}_z) \propto \exp(\alpha_k + \gamma f_{ik})$, and the pseudo-likelihood is

$$\pi_p(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z) = \prod_i \pi(\tilde{z}_i|\tilde{z}_j, j \in \mathcal{N}_i, \boldsymbol{\Psi}_z) = \prod_i \frac{\exp(\sum_k \alpha_k z_{ik} + \gamma \sum_k z_{ik} f_{ik})}{\sum_k \exp(\alpha_k + \gamma f_{ik})}.$$

The derivatives needed for the parameter estimation are

$$\frac{\partial \log\left(\pi_p(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z)\right)}{\partial\, \alpha_k} = \sum_i \left( z_{ik} - \frac{\exp(\alpha_k + \gamma f_{ik})}{\sum_l \exp(\alpha_l + \gamma f_{il})} \right),$$
$$\frac{\partial \log\left(\pi_p(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z)\right)}{\partial\, \gamma} = \sum_i \sum_k \left( z_{ik} f_{ik} - \frac{\exp(\alpha_k + \gamma f_{ik}) f_{ik}}{\sum_l \exp(\alpha_l + \gamma f_{il})} \right).$$

The derivatives needed to evaluate the scaling matrix $\mathbf{S}$ are

$$\frac{\partial^2 \log\left(\pi_p(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z)\right)}{\partial\, \alpha_{k_1}\, \partial\, \alpha_{k_2}} = \sum_i \left( -\mathbb{I}_{k_1=k_2} \frac{\exp(\alpha_{k_1} + \gamma f_{ik_1})}{\sum_j \exp(\alpha_j + \gamma f_{ij})} + \frac{\exp(\alpha_{k_1} + \gamma f_{ik_1}) \exp(\alpha_{k_2} + \gamma f_{ik_2})}{(\sum_j \exp(\alpha_j + \gamma f_{ij}))^2} \right),$$
$$\frac{\partial^2 \log\left(\pi_p(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z)\right)}{\partial\, \gamma^2} = \sum_i \left( -\left( \sum_k \frac{\exp(\alpha_k + \gamma f_{ik}) f_{ik}^2}{\sum_j \exp(\alpha_j + \gamma f_{ij})} \right) + \frac{(\sum_k \exp(\alpha_k + \gamma f_{ik}) f_{ik})^2}{(\sum_j \exp(\alpha_j + \gamma f_{ij}))^2} \right),$$
$$\frac{\partial^2 \log\left(\pi_p(\tilde{\mathbf{z}}|\boldsymbol{\Psi}_z)\right)}{\partial\, \alpha_k\, \partial\, \gamma} = \sum_i \left( -\frac{\exp(\alpha_k + \gamma f_{ik}) f_{ik}}{\sum_j \exp(\alpha_j + \gamma f_{ij})} + \frac{\exp(\alpha_k + \gamma f_{ik}) \sum_j \exp(\alpha_j + \gamma f_{ij}) f_{ij}}{(\sum_j \exp(\alpha_j + \gamma f_{ij}))^2} \right).$$

### *A.2. The gradient for the parameters of the Gaussian fields*

We also need the gradient of the log-likelihood with respect to $\boldsymbol{\Psi}_\xi$. In order to derive an estimate of the gradient we first compute $\log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}, \boldsymbol{\Psi}_\xi)$ using that the density $\pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}, \boldsymbol{\Psi}_\xi)$ is Gaussian, and that the fields $\boldsymbol{\xi}_k$ are independent given $\tilde{\mathbf{z}}$. To state the expression for $\log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}, \boldsymbol{\Psi}_\xi)$, we let $\mathcal{I}$ denote the set of indices of the lattice nodes for which there are observations. We further let $\mathcal{I}_k^{(j)}$ be the indices where also $\tilde{\mathbf{z}}^{(j)} = k$. Define the matrices $\mathbf{A}_{obs} = \mathbf{I}_{\mathcal{I},1:n}$ and $\mathbf{A}_k = \mathbf{I}_{\mathcal{I}_k^{(j)},1:n}$, the complete log-likelihood then is

$$\begin{aligned}\log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}, \boldsymbol{\Psi}_\xi) = C + \sum_{i=1}^{d} -|\mathcal{I}| \log(\sigma_i) + \sum_{k=1}^{K} \Bigg\{ \frac{1}{2} \log|\boldsymbol{\Sigma}_k^{-1} \otimes \mathbf{Q}_{\kappa_k}| - \frac{1}{2} \boldsymbol{\xi}_k^\top \boldsymbol{\Sigma}_k^{-1} \otimes \mathbf{Q}_{\kappa_k} \boldsymbol{\xi}_k \\ - \frac{1}{2} \left[\mathbf{Y} - \mathbf{A}_{obs}(\mathbf{B}_k \boldsymbol{\beta}_k + \boldsymbol{\xi}_k)\right]^\top \left(\boldsymbol{\Sigma}_\epsilon^{-1} \otimes \mathbf{A}_k^\top \mathbf{A}_k\right) \left[\mathbf{Y} - \mathbf{A}_{obs}(\mathbf{B}_k \boldsymbol{\beta}_k + \boldsymbol{\xi}_k)\right] \Bigg\},\end{aligned} \tag{4}$$

where $C$ is a constant not depending on the parameters and $\boldsymbol{\Sigma}_\epsilon$ is a diagonal matrix with $[\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2]$ on the diagonal. To simplify computations, we parametrize $\boldsymbol{\Sigma}_k$ as $\boldsymbol{\Sigma}_k = \mathbf{Q}_k^{-1}$, where $\mathbf{Q}_k = \mathbf{R}_k^\top \mathbf{R}_k$ and

$$\mathbf{R}_k = \begin{bmatrix} \exp(\eta_{k,1}) & \eta_{k,2} & \eta_{k,4} & \cdots & \eta_{k,d(d-1)/2+1} \\ 0 & \exp(\eta_{k,3}) & \eta_{k,5} & \cdots & \vdots \\ 0 & 0 & \ddots & & \\ 0 & 0 & 0 & 0 & \exp(\eta_{k,d(d+1)/2}) \end{bmatrix} \tag{5}$$

is the unique Cholesky factor of $\mathbf{Q}_k$ with $d(d+1)/2$ parameters $\boldsymbol{\eta}_k$. We take the exponential on the diagonal in (5) to ensure that the elements are non-negative. We further let $\tilde{\kappa}_k = \log(\kappa_k^2)$ and $\tilde{\sigma}_j = \log(\sigma_j)$ since these parameters are positive.

Let $L = \log \pi(\mathbf{Y}, \boldsymbol{\xi}|\tilde{\mathbf{z}}, \boldsymbol{\Psi}_\xi)$. The derivatives needed to evaluate the gradient are

$$\mathsf{E}\left[ \frac{\partial L}{\partial\, \tilde{\kappa}_k} \middle| \mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi} \right] = d \operatorname{tr}\left( \mathbf{Q}_{\kappa_k}^{-1} \frac{\partial\, \mathbf{Q}_{\kappa_k}}{\partial\, \tilde{\kappa}_k} \right) - \frac{1}{2} \hat{\boldsymbol{\xi}}_k^\top \mathbf{Q}_k \otimes \frac{\partial\, \mathbf{Q}_{\kappa_k}}{\partial\, \tilde{\kappa}_k} \hat{\boldsymbol{\xi}}_k - \frac{1}{2} \operatorname{tr}\left( \mathbf{Q}_k \otimes \frac{\partial\, \mathbf{Q}_{\kappa_k}}{\partial\, \tilde{\kappa}_k} \hat{\mathbf{Q}}_k^{-1} \right),$$
$$\mathsf{E}\left[ \frac{\partial L}{\partial\, \eta_{k,j}} \middle| \mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi} \right] = n \operatorname{tr}\left( \mathbf{Q}_k \frac{\partial\, \mathbf{Q}_k}{\partial\, \eta_{k,j}} \right) - \frac{1}{2} \hat{\boldsymbol{\xi}}_k^\top \frac{\partial\, \mathbf{Q}_k}{\partial\, \eta_{k,j}} \otimes \mathbf{Q}_{\kappa_k} \hat{\boldsymbol{\xi}}_k - \frac{1}{2} \operatorname{tr}\left( \frac{\partial\, \mathbf{Q}_k}{\partial\, \eta_{k,j}} \otimes \mathbf{Q}_{\kappa_k} \hat{\mathbf{Q}}_k^{-1} \right),$$
$$\mathsf{E}\left[ \frac{\partial L}{\partial\, \boldsymbol{\beta}_k} \middle| \mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi} \right] = \mathbf{B}_k^\top \left( \boldsymbol{\Sigma}_\epsilon^{-1} \otimes \mathbf{A}_k \right) \left( \mathbf{Y} - \mathbf{A}_{obs}(\mathbf{B}_k \boldsymbol{\beta}_k + \hat{\boldsymbol{\xi}}_k) \right),$$

$$\mathsf{E}\left[\left.\frac{\partial L}{\partial \tilde{\sigma}_j}\right| \mathbf{Y}, \tilde{\mathbf{z}}, \boldsymbol{\Psi}\right] = -|\mathcal{I}| - \frac{1}{2}\sum_{k=1}^{K}\left\{\operatorname{tr}\left[\left(\frac{\partial\, \boldsymbol{\Sigma}_\epsilon^{-1}}{\partial\, \tilde{\sigma}_j} \otimes \mathbf{A}_k^\top \mathbf{A}_k\right)\hat{\mathbf{Q}}_k^{-1}\right] + \left[\mathbf{Y} - \mathbf{A}_{obs}\left(\mathbf{B}_k\boldsymbol{\beta}_k + \hat{\boldsymbol{\xi}}_k\right)\right]^\top \left(\frac{\partial\, \boldsymbol{\Sigma}_\epsilon^{-1}}{\partial\, \tilde{\sigma}_j} \otimes \mathbf{A}_k^\top \mathbf{A}_k\right)\left[\mathbf{Y} - \mathbf{A}_{obs}\left(\mathbf{B}_k\boldsymbol{\beta}_k + \hat{\boldsymbol{\xi}}_k\right)\right]\right\}.$$

Here $\hat{\mathbf{Q}}_k = \mathbf{Q}_k \otimes \mathbf{Q}_{\kappa_k} + \boldsymbol{\Sigma}_\varepsilon^{-1} \otimes \mathbf{A}_k^\top \mathbf{A}_k$ and $\hat{\boldsymbol{\xi}}_k = \hat{\mathbf{Q}}_k^{-1}\mathbf{A}_{obs}^\top\left(\boldsymbol{\Sigma}_\varepsilon^{-1} \otimes \mathbf{A}_k^\top \mathbf{A}_k\right)(\mathbf{Y} - \mathbf{B}_k\boldsymbol{\beta}_k)$ is the expected value of $\boldsymbol{\xi}_k$ given the current parameter estimates. The derivatives of $\mathbf{Q}_{\kappa_k}$ and $\mathbf{Q}_k$ are easy to evaluate in closed form. Similar computations give closed form expressions for the second derivatives needed to evaluate the scaling matrix $\mathbf{S}$.

For regular maximum likelihood inference of Gaussian variables one needs to compute $|\hat{\mathbf{Q}}_k|$, which typically is the main computational challenge when dealing with large data sets. For the gradient method, one instead has to compute the various traces in the expressions above. There are two computational issues that have to be solved for the method to be applicable to large data sets.

The first issue is to compute $\hat{\boldsymbol{\xi}}_k$, or more generally solve $\mathbf{v} = \hat{\mathbf{Q}}_k^{-1}\mathbf{b}$ for a vector $\mathbf{b}$. This can be done using sparse Cholesky factorization and back-substitution. However, in order to reduce the computational complexity we instead use the preconditioned conjugate gradient method (PCG) with a robust incomplete Cholesky preconditioner (Ajiz and Jennings, 1984). This issue is related to the problem of sampling $\pi(\boldsymbol{\xi}_k|\mathbf{Y}, \tilde{\mathbf{z}}^{(j-1)}, \boldsymbol{\Psi})$. Using the method based on Cholesky factorization of $\hat{\mathbf{Q}}_k$ quickly becomes computational infeasible for large multivariate data sets. Fortunately, we can adapt the method by Papandreou and Yuille (2011) (see also Barman and Bolin, 2018) to sample $\boldsymbol{\xi}_k$ much cheaper. For a CAR(2) model, the method can be implemented as follows and does not require any computationally expensive Cholesky factors.

1. Generate $\mathbf{x} = \left(\mathbf{R}_k \otimes (\mathbf{G} + \kappa_k^2\mathbf{I})\right)\mathbf{x}_1 + \mathbf{A}_{obs}^\top\left(\boldsymbol{\Sigma}_\epsilon^{-1/2} \otimes \mathbf{A}_k^\top \mathbf{A}_k\right)\mathbf{x}_2$ where $\mathbf{x}_1$ and $\mathbf{x}_2$ are vectors of independent $\mathsf{N}(0, 1)$ random variables.
2. Solve $\hat{\mathbf{Q}}\boldsymbol{\xi}_k = \mathbf{x} + \mathbf{A}_{obs}^\top\left(\boldsymbol{\Sigma}_\epsilon^{-1} \otimes \mathbf{A}_k^\top \mathbf{A}_k\right)(\mathbf{Y} - \mathbf{B}_k\boldsymbol{\beta}_k)$ using the PCG method.

The method is based on the fact that $\mathbf{G} + \kappa_k^2\mathbf{I}$ is a matrix root of the precision matrix $\mathbf{Q}_{\kappa_k}$ of the CAR(2) model. For CAR(p) models with even $p$, the method can be used with $(\mathbf{G} + \kappa_k^2\mathbf{I})^{p/2}$ as a matrix root. For the CAR(1) model, a matrix root can instead be calculated based on the interpretation of the model as a prior on differences between neighboring voxels (Sidén et al., 2017). This can then be used to construct a matrix root for a CAR(p) model with any odd $p$.

The second issue with evaluating the gradient is to solve the various traces of inverse matrices present in the expressions. Recent work in spatial statistics (Anitescu et al., 2012; Stein et al., 2013) has proposed solving this issue using stochastic programming. The basic idea is to note that $\mathsf{E}[\mathbf{u}^\top\mathbf{Q}\mathbf{u}] = tr(\mathbf{Q})$ for any vector $\mathbf{u}$ of independent random variables with mean zero and variance one (Hutchinson, 1990). Thus, we can rewrite all the traces in the gradient as expectations, which can be approximated using Monte Carlo integration. For example $\operatorname{tr}\left(\mathbf{Q}_\kappa^{-1}\frac{\partial \mathbf{Q}_\kappa}{\partial \kappa_0}\right) = \mathsf{E}\left[\mathbf{u}^\top\frac{\partial \mathbf{Q}_\kappa}{\partial \kappa_0}\mathbf{Q}_\kappa^{-1}\mathbf{u}\right]$ is replaced with $J^{-1}\sum_{j=1}^{J}\left(\mathbf{u}^{(j)}\right)^\top\frac{\partial \mathbf{Q}_\kappa}{\partial \kappa_0}\mathbf{Q}_\kappa^{-1}\mathbf{u}^{(j)}$. The standard choice for $\mathbf{u}^{(j)}$ is a vector with mean-zero Bernoulli random variables, but for spatial problems the variance of the estimator can be reduced by for example using the probing vectors proposed by Aune et al. (2012). The trace approximation adds an additional source of randomness to the stochastic gradient approximation, but does not change the convergence properties. For all applications in this paper, we have used $J = 20$.

## References


Ajiz, M., Jennings, A., 1984. A robust incomplete Choleski-conjugate gradient algorithm. Internat. J. Numer. Methods Engrg. 20 (5), 949–966.
Andrieu, C., Moulines, É., Priouret, P., 2005. Stability of stochastic approximation under verifiable conditions. SIAM J. Control Optim. 44 (1), 283–312.
Anitescu, M., Chen, J., Wang, L., 2012. A matrix-free approach for solving the parametric Gaussian process maximum likelihood problem. SIAM J. Sci. Comput. 34 (1), A240–A262.
Asmussen, S., Glynn, P., 2007. Stochastic Simulation: Algorithms and Analysis. In: Stochastic Modelling and Applied Probability, Springer New York.
Aune, E., Simpson, D.P., Eidsvik, J., 2012. Parameter estimation in high dimensional Gaussian distributions. Stat. Comput. 1–17.
Barman, S., Bolin, D., 2018. A three-dimensional statistical model for imaged microstructures of porous polymer films. J. Microsc. 269 (3), 247–258.
Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems (with discussion). J. Roy. Statist. Soc. Ser. B Stat. Methodol. 36, 192–225.
Bolin, D., Lindgren, F., 2011. Spatial models generated by nested stochastic partial differential equations, with an application to global ozone mapping. Ann. Appl. Stat. 5 (1), 523–550.
Bolin, D., Lindström, J., Eklundh, L., Lindgren, F., 2009. Fast estimation of spatially dependent temporal vegetation trends using Gaussian Markov random fields. Comput. Stat. Data Anal. 53, 2885–2896.
Chiles, J.-P., Delfiner, P., 1999. Geostatistics, Modeling Spatial uncertainty. Wiley Series in Probability and statistics.
Cressie, N., 1991. Statistics for Spatial Data. John Wiley & Sons Ltd, New York, NY, USA.
Cressie, N., Wikle, C., 2011. Statistics for Spatio-Temporal Data. In: Wiley Series in Probability and Statistics, Wiley, Hoboken, New Jersey.
Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM Algorithm. J. Roy. Statist. Soc. Ser. B Stat. Methodol. 39 (1), 1–38.
Dunlop, M.M., Iglesias, M.A., Stuart, A.M., 2017. Hierarchical Bayesian level set inversion. Stat. Comput. 27 (6), 1555–1584.
Fernández, C., Green, P.J., 2002. Modelling spatially correlated data via mixtures: a Bayesian approach. J. Roy. Statist. Soc. Ser. B Stat. Methodol. 64 (4), 805–826.
Fuentes, M., Smith, R.L., 2001. A New Class of Nonstationary Spatial Models. Technical report. North Carolina State University, Raleigh, NC.
Fuglstad, G.A., Lindgren, F., Simpson, D., Rue, H., 2015. Exploring a new class of non-stationary spatial Gaussian random fields with varying local anisotropy. Statist. Sinica 25, 115–133.

Gneiting, T., Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. J. Amer. Statist. Assoc. 102 (477), 359–378.
Gramacy, R.B., Lee, H.K.H., 2008. Bayesian treed Gaussian process models with an application to computer modeling. J. Amer. Statist. Assoc. 103 (483), 1119–1130.
Guyon, X., 1995. Random Fields on a Network: Modeling, Statistics, and Applications. In: Graduate Texts in Mathematics, Springer.
Held, K., Kops, E.R., Krause, B.J., Wells III, W.M., Kikinis, R., Muller-Gartner, H.-W., 1997. Markov random field segmentation of brain MR images. IEEE Trans. Med. Imaging 16 (6), 878–886.
Higdon, D., 2001. Space and Space-time modeling using process convolutions. Technical Report 01–03, Duke University, Durham, NC.
Hildeman, A., Bolin, D., Wallin, J., Illian, J.B., 2017a. Level set Cox processes. Spat. Stat. in press.
Hildeman, A., Bolin, D., Wallin, J., Johansson, A., Nyholm, T., Asklund, T., Yu, J., 2017b. Whole-brain substitute CT generation using Markov random field mixture models. Preprint arXiv:1607.02188.
Hutchinson, M., 1990. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Commun. Stat. Simul. Comput. 19 (2), 433–450.
Kim, H.-M., Mallick, B.K., Holmes, C.C., 2005. Analyzing nonstationary spatial data using piecewise Gaussian processes. J. Amer. Statist. Assoc. 100 (470), 653–668.
Lange, K., 1995. A gradient algorithm locally equivalent to the EM algorithm. J. Roy. Statist. Soc. Ser. B Stat. Methodol. 57 (2), 425–437.
Lindgren, F., Rue, H., Lindström, J., 2011. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach (with discussion). J. Roy. Statist. Soc. Ser. B Stat. Methodol. 73, 423–498.
Paciorek, C.J., Schervish, M.J., 2006. Spatial modelling using a new class of nonstationary covariance functions. Environmetrics 17 (5), 483–506.
Papandreou, G., Yuille, A.L., 2011. Efficient variational inference in large-scale Bayesian compressed sensing. In: IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition. IEEE, pp. 1332–1339.
Reynolds, D.A., Rose, R.C., 1995. Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans. Speech Audio Process. 3 (1), 72–83.
Sidén, P., Eklund, A., Bolin, D., Villani, M., 2017. Fast Bayesian whole-brain fMRI analysis with spatial 3D priors. NeuroImage 146, 211–225.
Sidén, P., Lindgren, F., Bolin, D., Villani, M., 2018. Efficient covariance approximations for large sparse precision matrices. Comput. Graph. Statist. (in press).
Stauffer, C., Grimson, W.E.L., 1999. Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2. IEEE, pp. 246–252.
Stein, M.L., Chen, J., Anitescu, M., 2013. Stochastic approximation of score functions for Gaussian processes. Ann. Appl. Stat. 7 (2), 1162–1191.
Takahashi, K., Fagan, J., Chen, M.S., 1973. Formation of sparse bus impedance matrix and its application to short circuit study. In: IEEE Power Industry Computer Applications Conference, Vol. 8. pp. 63–69.
Van Leemput, K., Maes, F., Vandermeulen, D., Suetens, P., 1999. Automated model-based tissue classification of MR images of the brain. IEEE Trans. Med. Imaging 18 (10), 897–908.
Winkler, G., 2003. Image Analysis, Random Fields and Markov chain Monte Carlo methods: A Mathematical Introduction, second ed.. Springer, Berlin, Heidelberg.
Zhang, Y., Brady, M., Smith, S., 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans. Med. Imaging 20 (1), 45–57.