Multilevel discretized random field models with ‘spin’ correlations for the simulation of environmental spatial data

Milan Žukovič and Dionissios T Hristopulos
Technical University of Crete, Geostatistics Research Unit, Chania 73100, Greece
E-mail: mzukovic@mred.tuc.gr and dionisi@mred.tuc.gr

Received 24 September 2008
Accepted 26 November 2008
Published 9 February 2009

Abstract. A current problem of practical significance is how to analyze large, spatially distributed, environmental data sets. The problem is more challenging for variables that follow non-Gaussian distributions. We show by means of numerical simulations that the spatial correlations between variables can be captured by interactions between ‘spins’. The spins represent multilevel discretizations of environmental variables with respect to a number of pre-defined thresholds. The spatial dependence between the ‘spins’ is imposed by means of short-range interactions. We present two approaches, inspired by the Ising and Potts models, that generate conditional simulations of spatially distributed variables from samples with missing data. Currently, the sampling and simulation points are assumed to be at the nodes of a regular grid. The conditional simulations of the ‘spin system’ are forced to respect locally the sample values and the system statistics globally. The second constraint is enforced by minimizing a cost function representing the deviation between normalized correlation energies of the simulated and the sample distributions. In the approach based on the \( N_c \)-state Potts model, each point is assigned to one of \( N_c \) classes. The interactions involve all the points simultaneously. In the Ising model approach, a sequential simulation scheme is used: the discretization at each simulation level is binomial (i.e., \( \pm 1 \)). Information propagates from lower to higher levels as the simulation proceeds. We compare the two approaches in terms of their ability to reproduce the target statistics (e.g., the histogram and the variogram of the sample distribution), to predict data at unsampled locations, as well as in terms of their computational complexity. The comparison is based on a non-Gaussian data set (derived from a digital elevation model of the Walker Lake area,
We discuss the impact of relevant simulation parameters, such as the domain size, the number of discretization levels, and the initial conditions.

**Keywords:** new applications of statistical mechanics

**ArXiv ePrint:** 0809.3918

## Contents

1. **Introduction** 2
2. **Spatial prediction and classification** 4
   2.1. $k$-nearest-neighbor ($k$-NN) classifier 5
3. **Spatial classification based on ‘spin’ correlation models** 5
   3.1. Ising model 5
   3.2. Potts model 6
   3.3. General assumptions for NN–MLC classifiers 6
   3.4. Ising-based NN–MLC model (I-NN–MLC) 7
   3.5. Potts-based NN–MLC model (P-NN–MLC) 7
4. **Simulations of missing data on regular grids** 8
   4.1. Greedy Monte Carlo 8
   4.2. Simulation algorithms 9
   4.3. Initial state selection 10
5. **Case study: missing data reconstruction** 11
   5.1. Computational details 11
   5.2. Analysis of missing data reconstruction results 12
   5.3. Computational performance of classification methods 15
   5.4. Reconstruction of the synthetic Gaussian random field 16
6. **Conclusions and future research** 19

**Acknowledgments** 20

**References** 20

### 1. Introduction

Spatially distributed data are common in the physical sciences. They represent various environmental variables, such as contaminant concentrations in the atmosphere, soil permeability and dispersivity, etc. To account for the spatial variation and the...
measurement uncertainty of such quantities, the mathematical model of spatial random
fields is commonly used. Huge numbers of large spatial data sets are gathered from
around the globe on a daily basis. The efficient storage, analysis, harmonization, and
integration of such data is of great importance in various scientific areas such as image
processing, pattern recognition, remote sensing, and environmental monitoring. Often
the data coverage is incomplete for various reasons, such as meteorological conditions
(measurements hindered by clouds, aerosols, or heavy precipitation) or equipment
limitations (values below detection level or resolution threshold). Hence, in order to use
standard tools for the data analysis and visualization, one needs to deal with the problem
of incomplete (missing) data. In addition, data sampled at different resolutions may need
to be combined (e.g., in data fusion methods). This requires downscaling (refining) of the
data with the coarser resolution.

These tasks can be performed by means of well established interpolation and
classification techniques [1]. While deterministic methods (e.g., nearest-neighbor or
inverse distance interpolation) can be used, stochastic methods are often preferred
because they are more flexible in incorporating spatial correlations and they provide
estimates of prediction uncertainty. However, considering the ever-increasing size of
spatial data, stochastic methods such as kriging [2] can be impractical due to their
high computational complexity, requiring the use of high performance computational
technologies [3]. Furthermore, kriging is founded on the assumption of a jointly Gaussian
distribution, which is the data often fail to obey. Practical application of kriging
involves considerable human (subjective) input, regarding the selection of the correlation
(variogram) model and the kriging neighborhood selection [4].

The classical geostatistical approach relies on the structure function (variogram) for
modeling the spatial correlations. However, the correlations can also be considered as
the outcome of ’interactions’ between the field values at different points [5] that generate
short-range spatial order. In the case of static disorder, which is typical in the case
of ’quenched’ geological disorder or measurements representing a single time slice of
a dynamical process, such interactions represent ’effective constraints’ imposed by the
underlying process. For example, in the case of a digital elevation model, these constraints
are imposed by the topography of the area. While the nature of the constraints may differ
significantly between processes, we believe that universal aspects of spatial correlations
can be captured by means of relatively simple effective interactions. By incorporating the
interactions in an energy functional a Gibbs probability measure can be defined as in [5].
The realization probability of each spatial configuration is then governed by the relative
weight of the ’many-body’ Gibbs probability density function. In the Gaussian case it is
straightforward to define the interactions between the random field values at neighboring
locations. On a regular lattice this approach defines Gauss–Markov random fields.

In the non-Gaussian case, it is convenient to discretize the continuous values of the
random field using a set of discrete ’spins’. We arbitrarily define higher spin values to
represent higher values of the field. In the spirit of the approach described above, one can
then consider interactions between different ’spins’. In this framework, the problems of
spatial interpolation and simulation are mapped into a spatial classification problem, in
which each ’prediction’ location is assigned to a specific spin value (class). The concept
of classical spins is a suitable choice for modeling the multilevel discretization of the
continuous field. Spin models from statistical physics, e.g., the Ising and Potts models,
have already been applied in various problems in economy and finance [6]–[8], materials science [9]–[11], and biology [12]. However, these studies focus mainly on long-range correlations. In the framework of Gibbs–Markov random fields, the short-range correlation properties of Potts and Ising models have been widely applied in image analysis (see e.g. [13]–[17]). The Potts model in the superparamagnetic regime was also proposed as a data clustering model [18]. We introduce here two non-parametric models for spatial classification that are loosely based on the Ising and Potts spin models. The first model employs a sequential classification approach while the second has a simultaneous (parallel) classification of all levels.

The rest of the paper is organized as follows. In section 2, we present the problem of spatial classification/prediction, and we review the commonly used classification algorithm of the k nearest neighbors. Section 3 briefly reviews the Ising and Potts models and introduces the ‘spin’ nearest-neighbor correlation models that we propose for spatial classification. In section 4 we investigate technical aspects of the simulations, and we describe computational details. Section 5 focuses on the case study: it presents computational details as well as the analysis of the simulation results. Finally, in section 6, we summarize the relevant results and point out some future directions.

2. Spatial prediction and classification

Let us consider a set of sampling points $G_s = \{\vec{r}_i\}$, where $\vec{r}_i = (x_i, y_i) \in \mathbb{R}^2$ and $i = 1, \ldots, N$. These points are assumed to be scattered on a rectangular grid $\hat{G}$ of size $N_G = L_x L_y > N$, where $L_x$ and $L_y$ are respectively the number of nodes in the orthogonal directions (in terms of the unit length). If $z_i$ is a value attributed to the point $\vec{r}_i$, the set $Z\{G_s\} = \{z_i \in \mathbb{R}\}$ represents the sample of the process. Here we assume that $Z\{G_s\}$ represents a sample from a realization of a continuous random field $Z(\vec{r}; \omega)$, where $\omega$ is the state index.

Let $G_p = \{\vec{r}_p\}$ be the set of prediction points where $p = 1, \ldots, P$, such that $\hat{G} = G_s \cup G_p$. We discretize the continuous distribution of $Z$ using a number of classes, $C, q = 1, \ldots, N_c$. The classes are defined with respect to a set of threshold levels $t_k, k = 1, \ldots, N_c + 1$. If $z_{\text{min}} = \min(z_1, \ldots, z_N), z_{\text{max}} = \max(z_1, \ldots, z_N), \delta t = (z_{\text{max}} - z_{\text{min}})/N_c$, the thresholds are defined as $t_1 = -\infty, t_2 = z_{\text{min}} + \delta t, t_i = t_{i-1} + \delta t$ ($i = 3, \ldots, N_c$) and $t_{N_c+1} = \infty$. Each class $C_q$ corresponds to the interval $C_q = (t_q, t_{q+1}]$ for $q = 1, \ldots, N_c$. This means that all the classes, except the first and the last, have uniform width. The classes $C_1$ and $C_{N_c}$ extend to infinity (negative and positive respectively), to allow for values at the prediction points that lie outside the observed interval $[z_{\text{min}}, z_{\text{max}}]$. We define the class indicator field (phase field) $I_Z(\vec{r})$ by means of

$$I_Z(\vec{r}_i) = \sum_{q=1}^{N_c} q [\theta(z_i - t_q) - \theta(z_i - t_{q+1})], \quad \forall i = 1, \ldots, N,$$

where $\theta(x) = 1$ for $x > 0$ and $\theta(x) = 0$ for $x \leq 0$ is the unit step function. Prediction of the field values $Z\{G_p\}$ is mapped onto a classification problem, i.e., into estimating $\hat{I}_Z\{G_p\}$. If the number of levels is high, the continuum limit is approached.

\texttt{doi:10.1088/1742-5468/2009/02/P02023}
2.1. \( k \)-nearest-neighbor (\( k \)-NN) classifier

The \( k \)-NN method is a supervised learning algorithm, which assigns to each prediction point the class represented by the majority of its \( k \) nearest neighbors from the sample [19]. The distance metric used is typically Euclidean. The optimal value of the parameter \( k \) depends on the data and requires tuning for different applications. The optimization of \( k \) usually involves heuristic techniques (e.g., cross-validation). The accuracy of the \( k \)-NN classifier is affected by noise or by selecting a neighborhood that does not include the pertinent spatial information. The \( k \)-NN algorithm has been widely applied in the fields of data mining, statistical pattern recognition, image processing and many others. In general, it has been found to outperform many other flexible non-linear methods, particularly in high dimensional spaces [20]. Furthermore, it is relatively easy to implement and has desirable consistency properties, e.g., favorable asymptotic classification error dependence.

We use the \( k \)-NN classifier as a benchmark for the ‘spin-based’ classification methods that we propose. To eliminate the effect of \( k \) on the classification results, for each simulated realization we perform the classification for a wide range of values of \( k = 1, \ldots, k_{\text{max}} \), and select the value \( k_{\text{opt}} \) that minimizes the misclassification rate.

3. Spatial classification based on ‘spin’ correlation models

We propose two non-parametric, nearest-neighbor (NN), multilevel correlation (MLC) models that are loosely based on the Ising and Potts models. The NN–MLC models are based on matching suitably normalized correlation energy functions calculated from the samples with those estimated over the entire prediction grid. Similar ideas of correlation energy matching were also applied in the reconstruction of digitized random media from limited morphological information [21, 22]. In contrast with those studies, in the NN–MLC models our study also uses local spatial information (i.e., the values at the sampling points). The prediction of the field values (more precisely, class) at unsampled locations is achieved by means of conditional simulations that respect locally the sample values and the correlation energy globally.

3.1. Ising model

The Ising model (e.g. [23]) involves discrete variables \( s_i \) (spins) placed on a sampling grid. Each spin can take two values, \( \pm 1 \), and the spins interact in pairs. Assuming only nearest-neighbor interactions, the energy of the system can be expressed by the following Hamiltonian:

\[
H_1 = - \sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i. \tag{2}
\]

The coupling strength \( J_{ij} \) controls the type (ferromagnetic for \( J_{ij} > 0 \), antiferromagnetic for \( J_{ij} < 0 \)) and strength of the interactions. The second term introduces a symmetry-breaking bias caused by the presence of a site dependent external field \( h_i \). The latter controls the mean spin value (the magnetization). The model is usually defined on a regular grid; the interactions are considered uniform and their range limited to nearest neighbors. However, the model can be generalized to include also irregular grids and longer-range interactions.
3.2. Potts model

The Potts model is a generalization of the Ising model (e.g. [24]). Instead of ±1, each spin is assigned an integer value \( s_i \in \{1, \ldots, N_c\} \), where \( N_c \) represents the total number of states or classes. The Hamiltonian of the Potts model is given by

\[
H_P = -\sum_{i,j} J_{ij} \delta(s_i, s_j) - \sum_i h_i s_i,
\]

where \( \delta \) is the Kronecker delta. Hence, only pairs of spins in the same state give a non-zero contribution to the correlation energy. For \( N_c = 2 \), the Potts model is equivalent to the 2D Ising model.

3.3. General assumptions for NN–MLC classifiers

The states (configurations) of both the Ising and Potts models are determined by the Gibbs probability density function

\[
f = Z^{-1} \exp(-H/k_B T),
\]

where \( H \) is the respective Hamiltonian, \( k_B \) is the Boltzmann’s constant, and \( T \) is the temperature. The partition function \( Z \) is obtained by summing \( \exp(-H/k_B T) \) over all possible spin configurations. Essentially, the Hamiltonian involves two parameters, i.e. the normalized interaction couplings \( \tilde{J}_{ij} = J_{ij}/k_B T \) and \( \tilde{h}_i = h_i/k_B T \). In the case of spatial classification, the parameters are not known a priori and need to be determined from the sample. The standard procedure for addressing this inverse problem entails using the maximum likelihood method. Assuming that the parameters can be inferred, the spin values at unsampled locations are predicted by maximizing the conditional probability density function \( f(I_Z\{G_p\}|I_Z\{G_s\}; \tilde{J}_{ij}, \tilde{h}_i) \). However, optimizing the likelihood of the models with respect to the parameters \( \tilde{J}_{ij}, \tilde{h}_i \) is a computationally intensive task, since there are no generally valid closed-form expressions for the partition function. In order to overcome this problem we opted for a non-parametric approach.

Our NN–MLC models retain only the interaction energies of the Ising and Potts spin Hamiltonians. The sample values \( Z\{G_s\} \) are mapped into spin values by suitable discretization (as shown below). The main idea in both methods is to match the sample correlation energy with the correlation energy of all the spins (i.e. including the simulated spins at prediction points). This relies on the ergodic assumption that the sample spin correlation energies accurately describe those of the entire system. The matching of the correlation energies is performed by means of a numerical Monte Carlo approach. During the process, the spins at the sample sites remain fixed. Hence, the procedure followed is conditional simulation, as opposed to deterministic prediction. Focusing on regular grids and assuming isotropic and nearest-neighbor ‘spin interactions’, it is reasonable to set \( J_{ij} = J_0 \) if the points \( \vec{r}_i, \vec{r}_j \) are lattice neighbors and \( J_{ij} = 0 \) otherwise. Furthermore, we set \( h_i = 0 \) \((i = 1, \ldots, N)\). This choice prevents explicit control of the mean spin. Nevertheless, as shown by the simulation results, judicious choice of the initial state allows the distribution of the predicted classes to accurately recover the class distribution of the sample. This is due to the fact that both the correlation energies and the local spin values are restricted by the sample.
3.4. Ising-based NN–MLC model (I-NN–MLC)

We propose a sequential scheme in which the sample, \( G^q_s \), and prediction, \( G^q_p \), grids are sequentially updated. Each simulation level \( q \) corresponds to a class index. The number of simulation levels coincides with that of discretization levels. For the lowest level \( G^1_s = G_s, G^1_p = G_p \). For \( q = 1, \ldots, N_c - 1 \) it holds that \( G^q_s \subseteq G^{q+1}_s \), \( G^q_p \supseteq G^{q+1}_p \) and \( \tilde{G} = G^q_s \cup G^q_p \). Binary-valued spins are used at each level \( q \) (\( q = 1, \ldots, N_c \)). Sites (either from the sample or simulated) having \( I_Z \leq q \) are assigned a spin value of \(-1\), while sites having \( I_Z > q \) are assigned a spin value of \( 1 \). All the sites that are assigned a spin \(-1\) value at level \( q \) retain this value at higher levels. This means that the areas of low values are classified first. Once a site \( \tilde{r}_i \) is assigned a spin \(-1\) value at level \( q < N_c \), it is also assigned a class value \( I_Z(\tilde{r}_i) = q \). At the same time, the set \( G^{q+1}_p \) acquires all the \(-1\) points while the set \( G^{q+1}_p \) is accordingly reduced. In contrast, for sites that are assigned a spin \( 1 \) value, \( I_Z(\tilde{r}_i) > q \), and the precise class value \( I_Z(\tilde{r}_i) \) is determined at a higher level. At level \( q = N_c \) all remaining sites are assigned to the \( N_c \)th class.

Let \( S^q_q = \{ s^q_i; \forall i \text{ s.t. } \tilde{r}_i \in G^q_s \} \), \( \forall q = 1, \ldots, N_c \), be the set of spin values included in the ‘sample’ at level \( q \). These values are \( \pm 1 \) (depending on \( z_i \)) if \( \tilde{r}_i \in G_s \) and \(-1\) if \( \tilde{r}_i \) is an already classified prediction point. The unknown values at level \( q \) are the spins at the remaining locations, denoted by \( S^q_p \).

Our non-parametric approach utilizes a cost function, \( U_1(S^q_p|S^q_q) \), that describes the deviation between the normalized correlation energies of the simulated spin configuration, \( \tilde{C}^q_1 \), and its sample counterpart, \( C^q_{1,s} \), estimated from the spins in \( G^q_s \):

\[
U_1(S^q_p|S^q_q) = \begin{cases} 
[1 - \tilde{C}^q_1(S^q_p,S^q_q)/C^q_{1,s}(S^q_q)]^2, & \text{for } C^q_{1,s} \neq 0, \\
\tilde{C}^q_1(S^q_p,S^q_q)^2, & \text{for } C^q_{1,s} = 0,
\end{cases}
\]

where \( C^q_{1,s}(S^q_q) = \langle s^q_i s^q_j \rangle_{G^q_s} \) is the spatially averaged (normalized by the number of nearest-neighbor pairs in \( G^q_s \)) spin pair correlation of the \( q \)-level sample and \( \tilde{C}^q_1(S^q_p,S^q_q) = \langle s^q_i s^q_j \rangle_{\tilde{G}} \) is the spatially averaged spin pair correlation over the entire grid. Given the above, the estimation of \( S^q_p \) is equivalent to finding the optimal configuration \( \tilde{S}^q_p \) that corresponds to the minimum of the cost function \( U_1 \) at a fixed temperature \( T \), i.e.,

\[
\tilde{S}^q_p = \arg \min_{S^q_p} U_1(S^q_p|S^q_q), \quad \text{for } q = 1, \ldots, N_c.
\]

3.5. Potts-based NN–MLC model (P-NN–MLC)

In the P-NN–MLC model the classification is performed for all classes simultaneously. Hence, there is only a single simulation level irrespective of the number of discretization levels. The grid spins \( S = \{ s_i \}, i = 1, \ldots, N, \) take values in the set \( s_i \in \{ 1, \ldots, N_c \} \). The cost function is given by

\[
U_p(S_p|S_s) = \begin{cases} 
[1 - \tilde{C}_p(S_p,S_s)/C_{P,s}(S_s)]^2, & \text{for } C_{P,s} \neq 0, \\
\tilde{C}_p(S_p,S_s)^2, & \text{for } C_{P,s} = 0.
\end{cases}
\]

\[\text{doi:10.1088/1742-5468/2009/02/P02023}\]
where $C_{P;S}(S) = \langle \delta(s_i, s_j) \rangle_{G_s}$ is the spatially averaged (normalized by the number of nearest-neighbor pairs in $G_s$) spin pair correlation of the sample and $\tilde{C}_P(S_p, S_s) = \langle \delta(s_i, s_j) \rangle_{\tilde{G}}$ is the average spin pair correlation over $\tilde{G}$. The estimation of $S_p$ is equivalent to finding the minimum of the cost function (6), i.e.,

$$\hat{S}_p = \operatorname{arg\,min}_{S_p} U_P(S_p|S_s).$$

(7)

4. Simulations of missing data on regular grids

We focus on samples with missing data on regular grids. On such grids, it is straightforward to determine the nearest neighbors and calculate the correlation energies. Both the I-NN–MLC and P-NN–MLC methods return a class indicator field (see equation (1)) $\hat{I}_Z = I_Z(G_s) \cup I_Z(G_p)$, which consists of the original sample classification and the class estimates at $G_p$. The indicator values at the sampling sites are exactly reproduced. Below we refer to $I_Z(G_s)$ as the training set. Optimizations of the cost functions (4) and (6) are performed using the Monte Carlo approach. The generation of new ‘trial’ spin states is realized using the Metropolis algorithm at zero temperature.

We use the rejection ratio defined by $\rho = (#\text{rejected states})/N(G^q_p)$, where $N(G^q_p)$ is the number of prediction points at the $q$ level, to determine the stopping criterion. More specifically, our simulations terminate if $\rho = 1$, i.e., if one complete sweep through the entire grid $G^q_p$ does not produce a single successful update. Reaching the termination criterion may require several sweeps through the lattice, depending on the initial state.

4.1. Greedy Monte Carlo

The $T = 0$ assumption implies that the stochastic selection of an energetically unfavorable spin configuration in the Metropolis step is not possible. Hence, the candidate ‘spin’ for the update is flipped unconditionally only if it lowers the cost function. This approach is called the ‘greedy’ Monte Carlo algorithm [25] and leads to very fast convergence. In contrast, in simulated annealing $T$ is slowly lowered starting from an initial high temperature state. This approach is much slower computationally. However, the configuration resulting from simulated annealing is less sensitive to the initial state. The sensitivity of the greedy algorithm is known to be especially pronounced in high dimensional spaces with non-convex energies. In such cases, the greedy algorithm is likely to be trapped in local minima, instead of converging to the global one. However, this is not a concern in the current problem. In fact, targeting the global minimum of $U_I$ and $U_P$ strongly emphasizes the sample correlation energy per ‘spin’ pair. However, the latter is influenced by sample-to-sample fluctuations.

On rectangular or square grids, further increase in computational efficiency is gained by taking advantage of the geometry and the nearest-neighbor interactions. This is achieved by splitting the grid into two interpenetrating subgrids, which allows vectorizing the algorithm. Hence, one sweep through the entire grid is performed in just two steps by simultaneously updating all the sites on one of the subgrids in each step.
4.2. Simulation algorithms

On the basis of the above, the Monte Carlo (MC) algorithms for the I-NN–MLC and P-NN–MLC models consist of the following steps:

Algorithm for I-NN–MLC model
(1) Initialize the indicator field on the entire grid by means of $\hat{I}_Z(\hat{G}) = \text{NaN}$
(2) Set the simulation level (class index) to $q = 1$
(3) While [loop 1] $q \leq N_c - 1$ discretize $Z\{G_s\}$ with respect to $t_{q+1}$ to obtain $S^q_s$
   (3.1) Given the data $S^q_s$, calculate the sample correlation energy $C^q_{I,s}$
   (3.2) Assign initial values to the spins at $G^q_p$, i.e., generate $\hat{S}^q_p(0)$
   (3.3) Calculate the initial values of the simulated correlation $\hat{C}^q_p(0)$ and the cost function $U^q_p(0)$; initialize the MC index $i = 1$
   (3.4) Initialize the rejection ratio $\rho \rightarrow 0$; and the rejected states index $i_r = 0$
(3.5) While [loop 2] $\rho < 1$ repeat the following updating steps:
   (3.5.1) Generate a new state $\hat{S}^{(i+1)}_p$ by perturbing $\hat{S}^{(i)}_p$
   (3.5.2) Calculate $\hat{C}^{(i+1)}_p$ and $U^{(i+1)}_p$
   (3.5.3) If $U^{(i+1)}_p < U^{(i)}_p$ accept the new state; $i_r \rightarrow 0$.
   else keep the “old” state; $i_r \rightarrow i_r + 1$; endif
(3.6) Assign the $-1$ “spins” to the $q$ level, i.e., $\hat{I}_Z(\{\bar{r}_i\}) = q$
   If $\hat{S}^{(i_{\text{max}})}(\{\bar{r}_i\}) = -1$, $\{\bar{r}_i\} \in \hat{G}$
(3.7) Increase simulation level, $q \rightarrow q + 1$, return to step (3)
end [loop 2]
(4) For $q = N_c$, $\forall \bar{r}_i$ ($i = 1, \ldots, N_G$) such that $\hat{I}_Z(\{\bar{r}_i\}) = \text{NaN}$, set $\hat{I}_Z(\{\bar{r}_i\}) = N_c$.

In the above, the symbol NaN is used to denote non-numeric values.

Algorithm for P-NN–MLC model
(1) Discretize $Z\{G_s\}$ with respect to $t_k$, $k = 1, \ldots, N_c + 1$ to obtain $S_s$
(2) Given the data $S_s$, calculate the sample correlation energy $C_{P,s}$
(3) Assign initial values to the spins at $G_p$, i.e., generate $\hat{S}^{(0)}_p$
(4) Calculate the initial values of the simulated correlation $\hat{C}^{(0)}_p$ and the cost function $U^{(0)}_p$; initialize the MC index $i = 1$
(5) Initialize the rejection ratio $\rho \rightarrow 0$; and the rejected states index $i_r = 0$
(6) While $\rho < 1$ repeat the following updating steps:
   (6.1) Generate a new state $\hat{S}^{(i+1)}_p$ by perturbing $\hat{S}^{(i)}_p$
   (6.2) Calculate $\hat{C}^{(i+1)}_p$ and $U^{(i+1)}_p$
   (6.3) If $U^{(i+1)}_p < U^{(i)}_p$ accept the new state; $i_r \rightarrow 0$.
   else keep the ‘old’ state; $i_r \rightarrow i_r + 1$; endif
(6.4) $\rho \rightarrow i_r/N(G_p)$; $i \rightarrow i + 1$;
end
(7) Assign $\hat{I}_Z(\{\bar{r}_i\}) = \hat{S}^{(i_{\text{max}})}(\{\bar{r}_i\})$, $\{\bar{r}_i\} \in \hat{G}$.
Figure 1. Schematic demonstrating the stencil selection. Arrows pointing up correspond to +1 spins, arrows pointing down to −1 spins, and red circles to the prediction location. In the first plot (a) a square stencil of size $m_1 = 3$ is used. Within the neighborhood marked by the dash-line square there are equal numbers of +1 and −1 sites. In the second plot (b) a stencil of size $m_2 = 5$ is used to break the tie.

4.3. Initial state selection

The initial configuration of class indices can be selected in a number of ways. Since the proposed models aim to provide fast and automatic classification mechanisms, the initial configuration (assigned in steps (3.2) and (3) in the I-NN–MLC and P-NN–MLC algorithms) should minimize the relaxation path in state space to the equilibrium. It should also be selected with little or no user intervention. Since a degree of spatial continuity is common in geospatial data sets, it makes sense that the initial state of the individual prediction points is determined on the basis of the sample states in their immediate neighborhood. For square grids, we determine the neighborhood of $\vec{r}_p$ using an $m \times m$ stencil (where $m = 2l + 1$) centered at $\vec{r}_p$. Then, the initial value at a prediction point is assigned by majority rule, on the basis of the prevailing value of its sample neighbors inside the stencil. If we considered a circular stencil, this method would correspond to the $k$-NN classification algorithm, $k$ being the number of sampling points inside the stencil.

We set the stencil size adaptively to the smallest size that contains a clear majority of sample spin values, as shown schematically in figure 1. In practice, it makes sense to impose an upper limit on the stencil size $m_{\text{max}}$. If no majority is established up to the maximum stencil size $m_{\text{max}} \times m_{\text{max}}$, the initial value at $\vec{r}_p$ is assigned randomly from the equally represented class indices with the highest frequency (in the I-NN–MLC this means ±1). If a majority is not reached due to absence of sampling points inside the maximum stencil, the initial value is drawn randomly from the entire range of the labels $1, \ldots, N_c$. There are sensible reasons for imposing a maximum stencil size. First, considering too large neighborhoods in the $k$-NN classification generally generates oversmoothing at larger scales that cannot be justified as an effect of local continuity. Second, large neighborhoods increase the computational demands (both for memory and CPU time) disproportionally to expected benefits. Finally, assigning a portion of the prediction points initial values at random introduces a degree of randomness, which can be used to assess uncertainty by performing multiple runs on the same sample set. The choice of $m_{\text{max}}$ is arbitrary to an extent. Intuitively, for sparser sampling patterns larger $m_{\text{max}}$ should be considered. In
our investigations, relatively small sizes (up to $m_{\text{max}} = 7$) were sufficient for establishing good statistical performance at relatively small computational cost.

The algorithms thus require only that two parameters be set: $m_{\text{max}}$ and $N_c$. The latter depends on the study’s objectives: if the goal is to determine exceedance levels of a pollutant concentration with respect to a regulatory threshold, a binary classification is adequate. For environmental monitoring and decision making purposes a moderate number of classes (e.g., eight) is often sufficient.

5. Case study: missing data reconstruction

To test the classification models we use a synthetic pollutant concentration data set derived from a digital elevation model of the Walker Lake area in Nevada [26]. A two-dimensional projection of the pollution field is shown in figure 2(a). The units used for the $Z$ values are arbitrarily set to parts per million (ppm). Some summary statistics are as follows: number $N_{\tilde{G}} = 78,000$ on a $260 \times 300$ rectangular grid, $z_{\min} = 0$, $z_{\max} = 1631.2$, $\bar{z} = 277.9$, $z_{0.50} = 221.3$, $\sigma_z = 249.9$; the skewness coefficient is 1.02, and the kurtosis coefficient 3.78. As is evident from the above statistics and the histogram in figure 2(b), the distribution is clearly neither Gaussian nor log-normal.

5.1. Computational details

From the complete data we draw a sample $Z\{G_s\}$ (training set) of size $N = (1 - p)N_{\tilde{G}}$ by randomly removing $P = pN_{\tilde{G}}$ values (validation set), which are later used for prediction validation. For three degrees of thinning, $p = [0.33, 0.50, 0.66]$, we generate 100 different configurations of the training and validation sets. The values at the prediction points
are predicted (classified) using the I-NN–MLC and P-NN–MLC classification algorithms. The class indicator values at the prediction points $I_Z(G_p)$ are then compared with the classification estimates $\hat{I}_Z(G_p)$. To evaluate the classification performance, we calculate the misclassification rate as a fraction of misclassified pixels:

$$F = \frac{1}{P} \sum_{p=1}^{P} [1 - \delta(I_Z(\vec{r}_p), \hat{I}_Z(\vec{r}_p))],$$

where $I_Z(\vec{r}_p)$ is the true value at the validation points, $\hat{I}_Z(\vec{r}_p)$ is the classification estimate and $\delta(I, I') = 1$ if $I = I'$, $\delta(I, I') = 0$ if $I \neq I'$. The standard deviation of the quantity $F$ is evaluated using the values obtained from all the configurations, as

$$\text{STD}_F = \sqrt{\sum_{i=1}^{100} (F_i^* - F_i)^2 / 99}.$$  

We also compare the class index variograms of the original and reconstructed patterns in the orthogonal lattice directions, defined by

$$\hat{\gamma}_\iota(h) = \frac{1}{2|N_\iota(h)|} \sum_{i,j \in N_\iota(h)} |I_Z(\vec{r}_i) - I_Z(\vec{r}_j)|^2,$$

where $N_\iota(h)$ denotes the set of pairs of points in $\iota$ orthogonal lattice direction separated by distance $h$ (lag), and $|N_\iota(h)|$ denotes the number of pairs in the set.

Furthermore, we record the CPU time, the number of MC sweeps needed to reach equilibrium, and the residual values of the cost functions at termination. The procedure is repeated for $N_c = 8$ and $N_c = 16$. The computations are performed in the Matlab® programming environment on a desktop computer with 1.93 GB of RAM and an Intel® Core™2 CPU 6320 processor at 1.86 GHz.

### 5.2. Analysis of missing data reconstruction results

In figures 3–7 the classification performance of the two spin-based models is demonstrated for two limiting cases: one with a low number of levels and low degree of thinning ($N_c = 8$ and $p = 0.33$), and the other one with a high number of levels and high degree of thinning ($N_c = 16$ and $p = 0.66$). The plots in these figures include: in figure 3, a 2D projection of the reconstructed isolevel map (shown for the first realization), in figure 4, the spatial distribution of the class index standard deviations based on the 100 realizations, in figure 5, the histograms of the original data as well as the best (lowest $F^*$) and worst (highest $F^*$) reconstructions, and in figures 6 and 7, the variograms (along the directions of coordinate axes) of the original data versus those of the best and worst reconstructions. In all cases, there is good visual agreement between the spatial patterns of the original data and the reconstructions. However, at higher values of $p = 0.66$ and $N_c = 16$, closer inspection of the maps can reveal some degree of smoothing and small speckles of misclassified pixels. The higher misclassification in the $p = 0.66$ and $N_c = 16$ case is also manifested in the remaining quantities, in particular, poorer matching of the histograms, and larger deviations of the reconstructions’ variogram curves with respect to the original.

The histograms are displayed on a log–linear scale in order to better visualize also the small frequency classes (in the tail). The natural logarithm of the class frequencies, $N_i (i = 1, \ldots, N_c)$ is used. On the other hand, the logarithmic scale somewhat visually suppresses the differences in the high frequency classes. Nevertheless, in the I-NN–MLC model we can observe a systematic underestimation of the highest frequency (first)
Figure 3. The reconstructed maps corresponding to the first realization obtained from (a) I-NN–MLC with $p = 0.33$, (b) I-NN–MLC with $p = 0.66$, (c) P-NN–MLC with $p = 0.33$, and (d) P-NN–MLC with $p = 0.66$.

class and the low frequency classes (especially at larger $p$). Their underestimation (and overestimation of the classes closer to the mean $I_{30}^{16} \approx 3.3$) is reflected in the noticeable decrease in the class index variance of the reconstructed maps (as shown by the variogram plots). On the other hand, for the P-NN–MLC model, the frequencies of the most represented classes are reconstructed reasonably well, and the classes in the tail only represent a small portion of the total data (e.g., for $N_c = 16$ the classes larger than 13 represent less than 0.1% of $N_G$) and, therefore, the variations in their frequencies have relatively little impact on the variograms.

doi:10.1088/1742-5468/2009/02/P02023
Figure 4. The standard deviations calculated on the basis of the classification results based on 100 realizations obtained by means of (a) I-NN–MLC with $p = 0.33$, (b) I-NN–MLC with $p = 0.66$, (c) P-NN–MLC with $p = 0.33$, and (d) P-NN–MLC with $p = 0.66$.

The misclassification rates and their standard deviations for the two algorithms are compared in table 1. In terms of the misclassification rate, the I-NN–MLC model performs uniformly better than the P-NN–MLC model. For the current set the differences are not large, but they can be significant for different data (see section 5.4 below). Comparing the proposed spin models with the $k$-NN classifier, both models gave uniformly smaller misclassification rates than the best achievable by the $k$-NN algorithm.

doi:10.1088/1742-5468/2009/02/P02023
5.3. Computational performance of classification methods

The computational performance of the proposed spin classifiers is compared to the $k$-NN classifier also in table 1. Due to binary values of the Ising spins, the I-NN–MLC model requires a very small number of Monte Carlo sweeps over the grid to reach equilibrium. In the most ‘difficult’ case ($N_c = 16$ and $p = 0.66$), it takes less than 20 lattice sweeps. This implies short optimization CPU times at each level. A substantial fraction of the total CPU time is spent on the initial state assignments at each level. For the P-NN–MLC model the initial state is determined once. On the other hand, due to the significantly larger configuration space, the relaxation is much slower than for the I-NN–MLC model. Nevertheless, it is accomplished within maximum 32 (fastest) and 85 (slowest) MC sweeps in less than 2 (fastest) and 8 (slowest) seconds of CPU time. Optimizing the $k$-NN algorithm involved time-consuming multiple runs for each realization to test a wide
range of $k$ values, leading to considerably higher CPU times (not reported). In practical applications, the optimal value of $k$ is often selected by heuristic techniques (e.g., cross-validation), which also require user input and computational resources. Overall, the I-NN–MLC and P-NN–MLC models can provide better classification accuracy more efficiently and without user intervention.

5.4. Reconstruction of the synthetic Gaussian random field

To further investigate differences in the classification performance between the I-NN–MLC and P-NN–MLC models, we generated smooth synthetic data on a $50 \times 50$ grid. The data represent a realization (see figure 8) from a Gaussian random field $Z \sim N(m = 50, \sigma = 10)$ with Whittle–Matérn correlations [27].

Figure 6. The $x$-axis direction variograms of the original, the best ($F_{\text{min}}^*$), and the worst ($F_{\text{max}}^*$) reconstructed data, obtained by means of (a) I-NN–MLC with $p = 0.33$, (b) I-NN–MLC with $p = 0.66$, (c) P-NN–MLC with $p = 0.33$, and (d) P-NN–MLC with $p = 0.66$. 
Table 1. The mean values of the misclassification rate \( \langle F^* \rangle \) and the standard deviations STD\( F^* \) obtained from the I-NN–MLC and P-NN–MLC models are compared with the best results obtained from the \( k \)-NN classification (\( \langle F^*_{knn} \rangle \) and STD\( F^*_{knn} \)). The additional statistics for the I-NN–MLC and P-NN–MLC models include: the mean numbers of Monte Carlo sweeps \( \langle N_{MC} \rangle \), the mean values of the CPU time \( \langle T_{cpu} \rangle \), and the mean values of the cost function at termination \( \langle U^* \rangle \). The mean values are calculated over an ensemble of 100 realizations.

| \# of classes | 8 classes | 16 classes |
|---------------|-----------|------------|
| \( p \)       | 0.33      | 0.5        | 0.66       | 0.33 | 0.5 | 0.66 | 0.33 | 0.5 | 0.66 |
| Model         | k nearest neighbors |
| \( \langle F^*_{knn} \rangle \) (%) | 22.6 | 24.0 | 25.9 | 22.6 | 24.0 | 25.9 | 39.7 | 41.4 | 43.4 | 43.4 |
| STD\( F^*_{knn} \) | 0.24 | 0.20 | 0.16 | 0.24 | 0.20 | 0.16 | 0.31 | 0.22 | 0.18 | 0.31 |
| \( \langle F^* \rangle \) (%) | 21.4 | 22.6 | 24.2 | 21.7 | 22.9 | 24.5 | 37.1 | 39.0 | 41.7 | 38.1 |
| STD\( F^* \) | 0.23 | 0.19 | 0.17 | 0.23 | 0.20 | 0.17 | 0.28 | 0.22 | 0.21 | 0.29 |
| \( \langle N_{MC} \rangle \) | 5.9 | 7.5 | 9.1 | 31.8 | 41.4 | 41.4 | 13.8 | 17.4 | 19.4 | 65.8 |
| \( \langle T_{cpu} \rangle \) (s) | 2.62 | 3.21 | 3.87 | 1.97 | 3.04 | 4.28 | 5.31 | 6.29 | 7.15 | 3.37 |
| \( \langle U^* \rangle \) | \( 5 \times 10^{-4} \) | \( 1 \times 10^{-3} \) | \( 2 \times 10^{-3} \) | \( 1 \times 10^{-3} \) | \( 2 \times 10^{-3} \) | \( 2 \times 10^{-3} \) | \( 4 \times 10^{-4} \) | \( 8 \times 10^{-4} \) | \( 1 \times 10^{-3} \) | \( 3 \times 10^{-3} \) | \( 6 \times 10^{-3} \) | \( 6 \times 10^{-3} \) |
Figure 7. The y-axis direction variograms of the original, the best ($F^*_\text{min}$), and the worst ($F^*_\text{max}$) reconstructed data, obtained by means of (a) I-NN–MLC with $p = 0.33$, (b) I-NN–MLC with $p = 0.66$, (c) P-NN–MLC with $p = 0.33$, and (d) P-NN–MLC with $p = 0.66$.

The correlation function is $c(r) = \sigma^2 (2^{1-\nu}/\Gamma(\nu))(\kappa r)^\nu K_\nu(\kappa r)$, where $\nu = 2.5$ is the smoothness parameter, $\kappa = 0.2$ is the inverse correlation length, and $K_\nu$ is the modified Bessel function of order $\nu$. The biggest difference in classification accuracy between the two models is obtained for $N_c = 16$ and $p = 0.33$: $F^*_\text{P-NN–MLC} = 35.1\%$, while $F^*_\text{I-NN–MLC} = 21.2\%$. We believe that the superior performance of the I-NN–MLC model results from the sequential strategy, in which points classified as $-1$ at lower levels are included in the sampling set at higher levels. Provided that the classification at lower levels is accurate, which is more likely for rather smooth and noise-free data (like the synthetic ones shown above), the estimates included can significantly improve the model’s performance. The sequential algorithm also reduces potential state degeneracy (i.e., spin configurations with the same energy). This feature is likely to occur in the spin models and it increases ambiguity in the identification of a particular spin configuration from...
the correlation energy. At the same time, the propagation of classification results from lower to higher levels can also be a weakness of the sequential algorithm, since low level classification errors influence the higher levels.

6. Conclusions and future research

We presented non-parametric approaches for spatial classification, inspired from the Ising and the Potts spin models, with a sequential and simultaneous classification strategy, respectively. The concept is based on the idea of matching the normalized correlation energies calculated from discretized data over the sampling grid and the entire area of interest. The matching is performed using Monte Carlo simulations, conditional on the sample values. The main advantage of the models is that they do not have any hyperparameters that need tuning: hence the classification is automatic, objective, competitive (in accuracy) and computationally efficient. The proposed methods are applicable to non-Gaussian distributions. In addition, they can incorporate non-stationary data, because even with a constant coupling strength the spin interactions imply a local impact of the sample values.

The future research includes the extension to scattered sampling patterns. One possible way to proceed is to define the interaction constant $J_{ij}$ in the Hamiltonians (2)–(3) through a kernel function (such as the radial basis function), and the interaction neighborhood (nearest neighbors), for example, as pairs of points whose Voronoi cells have a common boundary. Another way is to first use a simple interpolation method to place the irregularly spaced points on a regular grid with a specified resolution and then proceed as in the current study. The latter approach would allow vectorization and preserve the computational efficiency. Further possible extensions of the current models could also include further-neighbor or ‘higher order’ (e.g., three-point) correlation energy in the respective Hamiltonians. We could expect some elimination of the degeneracy, witnessed in the present models, and more faithful characterization of the nature of the
spatial dependence. Both effects should contribute to the improvement of the classification performance. It would also be interesting to consider data sets with different patterns of missing data and investigate the effect of various gap patterns. Finally, it remains to be seen whether the proposed methods can be used in the case of data sets containing a small number of extreme values, for example, two or three unusually elevated values detected using a monitoring network in the case of a radioactivity release.

Acknowledgments

This research project was supported by a Marie Curie Transfer of Knowledge Fellowship of the European Community’s Sixth Framework Program under contract number MTKD-CT-2004-014135.

References

[1] Atkinson P M and Tate N J (ed), 1999 Advances in Remote Sensing and GIS Analysis (New York: Wiley)
[2] Wackernagel H, 2003 Multivariate Geostatistics (Berlin: Springer)
[3] Hawick K A, 1998 Proc. High Performance Computing and Networks Europe (LNCS vol 1401) (Berlin: Springer)
[4] Diggle P J and Ribeiro P J, 2007 Model-based Geostatistics (Springer Series in Statistics) (Berlin: Springer)
[5] Hristopulos D T, 2003 SIAM J. Sci. Comput. 24 2125
[6] Mantegna R N and Stanley H E, 1999 An Introduction to Econophysics: Correlations and Complexity in Finance (Cambridge: Cambridge University Press)
[7] Bouchaud J P and Potters M, 2000 Theory of Financial Risk (Cambridge: Cambridge University Press)
[8] Bouchaud J P, Alstrom P and Lauritsen K B (ed), Application of physics in financial analysis, 2000 Int. J. Theor. Appl. Finance 3 (special issue)
[9] Kardar M, Stella A L, Sartoni G and Derrida B, 1995 Phys. Rev. E 52 R1269
[10] Wouts M, 2008 Stoch. Process. Appl. 118 1929
[11] Zheng Y G, Lu C, Mai Y W, Zhang H W and Chen Z, 2006 Sci. Technol. Adv. Mater. 7 812
[12] Graner F and Glazier J A, 1992 Phys. Rev. Lett. 69 2013
[13] Besag J, 1986 J. R. Stat. Soc. B 48 259
[14] Gimel’farb G, 1999 Pattern Recognit. Lett. 20 1123
[15] Kato Z and Pong T-C, 2006 Image Vis. Comput. 24 1103
[16] Zhao J and Wang X, 2007 ICNC 2007: 3rd Int. Conf. on Natural Computation vol 2, p 742
[17] Tanaka K and Morita T, 1995 Phys. Lett. A 203 122
[18] Blatt M, Wiseman S and Domany E, 1996 Phys. Rev. Lett. 76 3251
[19] Dasarathy B V (ed), 1991 Nearest Neighbor (NN) Norms NN Pattern Classification Techniques (Los Alamitos, CA: IEEE Computer Society Press)
[20] Cherkesky V S and Gehring D, 1996 IEEE Trans. Neural Netw. 7 969
[21] Yeong C L Y and Torquato S, 1998 Phys. Rev. E 57 495
[22] Yeong C L Y and Torquato S, 1998 Phys. Rev. E 58 224
[23] McCoy B M and Wu T T, 1973 The Two-Dimensional Ising Model (Cambridge, MA: Harvard University Press)
[24] Wu F Y, 1982 Rev. Mod. Phys. 54 235
[25] Papadimitriou C H and Steiglitz K, 1982 Combinatorial Optimization (Englewood Cliffs, NJ: Prentice-Hall)
[26] Isaak E and Srivastava R, 1989 An Introduction to Applied Geostatistics (New York: Oxford University Press)
[27] Whittle P, 1954 Biometrika 41 434

doi:10.1088/1742-5468/2009/02/P02023