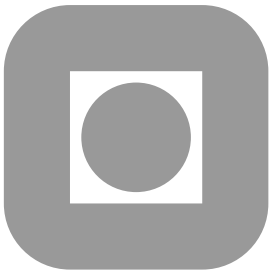
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Fully Bayesian binary Markov random field models: Prior specification and posterior simulation

by

Petter Arnesen and Håkon Tjelmeland

PREPRINT
STATISTICS NO. 7/2013



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL http://www.math.ntnu.no/preprint/statistics/2013/S7-2013.pdf
Petter Arnesen has homepage: http://www.math.ntnu.no/∼petterar
E-mail: petterar@stat.ntnu.no
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Fully Bayesian binary Markov random field models: Prior specification and posterior simulation

Petter Arnesen and Håkon Tjelmeland

We propose a flexible prior model for the parameters of a binary Markov random field (MRF) defined on a rectangular lattice and with $k \times l$ cliques. The prior model allows higher-order interactions to be included in the MRF. We also define a reversible jump Markov chain Monte Carlo (RJMCMC) algorithm to sample from the associated posterior distribution. The number of possible parameters for an MRF with $k \times l$ cliques becomes high even for small values of k and l. To get a flexible model which may adapt to the structure of a particular observed image we do not put any absolute restrictions on the parametrization. Instead we define a parametric form for the MRF where the parameters have interpretation as potentials for the various clique configurations, and limit the effective number of parameters by assigning apriori discrete probabilities for events where groups of parameter values are equal.

To run our RJMCMC algorithm we have to cope with the computationally intractable normalizing constant of MRFs. For this we adopt a previously defined approximation for binary MRFs, but we also briefly discuss other alternatives. We demonstrate the flexibility of our prior formulation in two examples with simulated data and in one real data example.

Key words: Approximate inference; Ising Model; Markov random fields; Reversible jump MCMC.

1. INTRODUCTION

Markov random fields (MRF) are frequently used as prior distributions in spatial statistics. A common situation is that we have an observed or latent field x which we model as an MRF, $p(x|\theta)$, conditioned on a vector of model parameters θ. The most common situation in the literature is to consider θ as fixed, see for instance examples in Besag (1986) and Hurn et al. (2003), but several articles have also considered a fully Bayesian approach by assigning a prior on θ. A fully Bayesian model is computationally simplest when $x|\theta$ is a Gaussian Markov random field (GMRF) and this case is therefore especially well developed. A flexible imple-

mentation of the GMRF case is given in the integrated nested Laplace approximation (INLA) software, see Rue et al. (2009) and Martins et al. (2013). The case when the components of x are discrete variables is computationally much harder and therefore less developed in the literature. However, some articles have considered the fully Bayesian approach also in this case, see in particular the early Heikkinen and Högmander (1994) and Higdon et al. (1997) and the more recent Møller et al. (2006), Friel et al. (2009), Austad (2011), McGrory et al. (2012) and Tjelmeland and Austad (2012).

Discrete MRFs contain a computationally intractable normalizing constant and this makes the fully Bayesian approach problematic. Three classes of approaches have been proposed to circumvent or solve this problem. The first is to replace the MRF likelihood with a computationally tractable approximation. The second alternative is to use an estimate of the normalization constant obtained by some Markov chain Monte Carlo (MCMC) procedure prior to simulating from the posterior for θ, and the third approach is to include an auxiliary variable sampled from the MRF $p(x|\theta)$ in the posterior simulation algorithm. Of the references cited above, Heikkinen and Högmander (1994), Friel et al. (2009), Austad (2011), McGrory et al. (2012) and Tjelmeland and Austad (2012) fall into the first class. The first of these five articles are using the pseudo-likelihood as approximation, Friel et al. (2009) and McGrory et al. (2012) are using a reduced dependency approximation (RDA), while the two remaining papers are using theory for pseudo-Boolean functions to construct approximations for binary MRFs. The strategy adopted in Higdon et al. (1997) falls into the second class defined above. Møller et al. (2006) is the first article using the third approach, and the exchange algorithm in Murray et al. (2006) is another member of this class. The three approaches all have their advantages and disadvantages. First of all, only the third approach is without approximations in the sense that it defines an MCMC algorithm with

limiting distribution exactly equal to the posterior distribution of interest. However, for this approach to be feasible perfect sampling from $p(x|\theta)$ must be possible, and computationally reasonably efficient, for all values of θ. The strategy used in the second class requires in practice that the parameter vector θ is low dimensional. The approximation strategy does not have restrictions on the dimension of θ and perfect sampling from $p(x|\theta)$ is not needed. In that sense this approach is more flexible, but of course the approximation quality will typically depend on the value of θ.

In this article we consider the fully Bayesian approach and for simplicity we limit the attention to the case where the components of x are binary. Our focus is on the specification of a prior distribution for θ and on simulation from the associated posterior distribution. The articles discussed above are only considering the Ising model and the closely related autologistic model in their example sections, and very simple prior distributions are adopted. In this article we assume $x|\theta$ to be an MRF with $k \times l$ cliques and allow also higher order interactions. For such a model the number of parameters becomes quite high even for small values of k and l, but to get a flexible prior model which may adapt to the structure of the particular observed image we do not put any absolute restrictions on the parametrization. Instead we limit the effective number of parameters by adopting a prior for θ with discrete probabilities for some parameter values to be equal. To simulate from the resulting posterior distribution we construct a reversible jump MCMC (RJMCMC) algorithm (Green 1995). One should note that with our choice of prior this algorithm effectively act as a model selection procedure. To run the RJMCMC algorithm we have to cope with the computationally intractable normalizing constant of the MRF. In principle any of the approaches discussed above may be used, but the complexity of the parameter space makes the prior estimation of the normalization constant approach impractical. Moreover, the accuracy of the pseudo-

likelihood approximation is known to be quite poor, and in simulation exercises we found that perfect sampling from $p(x|\theta)$ was extremely computational intensive for some values of θ. We are therefore left with the RDA approach and the approximation strategy based on pseudo-Boolean functions. In our simulation examples we adopt the latter of these, but RDA could equally well have been used.

The article has the following organization. In Section 2 we discuss possible parametrization of binary MRFs, and in particular we identify the maximal number of free parameters for a model with $k \times l$ cliques. We define our prior for θ in Section 3, and describe our RJMCMC algorithm for simulating from a posterior distribution in Section 4. In Section 5 we present results for two simulated data examples and for one real data example. Finally, some closing remarks are provided in Section 6.

2. MRF

In this section we give a brief introduction to MRFs, see Cressie (1993) and Hurn et al. (2003) for more details, and in particular we focus on binary MRFs and the parametrization in this case. We close with two examples of binary MRFs. This section provides the theoretical background needed in order to understand the construction of our prior distribution in Section 3, and the RJMCMC algorithm given in Section 4.

2.1 Binary MRF

Assume a rectangular lattice of dimension $n \times m$, and let the nodes be numbered lexicographically from 1 to nm. To each node $i \in S = \{1, ..., nm\}$ associate a binary variable $x_i \in \{0, 1\}$, and let $x = (x_1, ..., x_{nm})$ be the vector of these binary variables. Let $x_\Lambda = (x_i | i \in \Lambda)$ denote

the binary variables with indices belonging to an index set $\Lambda \subseteq S$. We will also use the notation $x_{-i} = x_{S\setminus\{i\}}$. Let $\mathcal{N} = \{\mathcal{N}_1, ..., \mathcal{N}_{nm}\}$ be a neighborhood system where $\mathcal{N}_i \subseteq S \setminus \{i\}$ is the set of neighbor nodes of node i. We assume symmetry so if $i \in \mathcal{N}_j$, then $j \in \mathcal{N}_i$. Now, x is a binary MRF if $p(x) > 0$ for all x, and $p(x_i|x_{-i})$ fulfills the Markov property

$$p(x_i|x_{-i}) = p(x_i|x_{\mathcal{N}_i}) \text{ for all } i.$$

A clique is defined to be a set $\Lambda \subseteq S$, where $i \in \mathcal{N}_j$ for all distinct $i, j \in \Lambda$, and we denote the set of all cliques by $\mathcal{L}$. Note that by this definition sets containing only one node and the empty set are cliques. A maximal clique is defined to be a clique that is not a subset of another clique, and we denote the set of all maximal cliques by $\mathcal{L}_m$. According to the Hammersley-Clifford theorem (Clifford 1990), the most general form the distribution $p(x)$ of an MRF can take is

$$p(x) = c \exp\left(\sum_{\Lambda \in \mathcal{L}_m} U_\Lambda(x_\Lambda, \theta)\right), \tag{1}$$

where c is a computationally demanding normalizing constant, $U_\Lambda(x_\Lambda, \theta)$ is a potential function for a given maximal clique Λ, and θ is a parameter vector.

To simplify the definition of a prior for the parameter vector θ of an MRF we first limit the attention to stationary MRFs defined on an $n \times m$ lattice with torus boundary conditions, and assume the neighborhood system to be such that the set of maximal cliques, $\mathcal{L}_m$, are equal to all $k \times l$ blocks of nodes on the torus. The torus assumption means that nodes close to the boundary have neighbors on the opposite boundary, and we get nm maximal cliques. To obtain this set of maximal cliques the set of neighbors $\mathcal{N}_i$ to any node i must clearly be the set of all nodes, except node i, lying within the $(2k+1) \times (2l+1)$ block of nodes centered at node i. The assumption of stationarity implies that the potential function $U_\Lambda(\cdot, \cdot)$ must be translational invariant in that the function must be equal for all $\Lambda \in \mathcal{L}_m$. We

can thereby simplify the notation by replacing $U_\Lambda(x_\Lambda, \theta)$ with $U(x_\Lambda, \theta)$. Later in the article we consider also the situation when the MRF is defined on a lattice with free boundaries, and in particular we discuss how our prior for θ can be adapted to this situation.

Consider a stationary MRF defined on a torus as defined above. Before defining a prior on θ in Section 3, we will in the following identify a parametric form for $U(x_\Lambda, \theta)$ which makes the corresponding $p(x)$ identifiable, but otherwise is as general as possible given our assumptions about stationarity and torus boundary conditions. For a maximal clique $\Lambda \in \mathcal{L}_m$ there is clearly 2^{kl} possible x_Λ, and we refer to these as configurations. We obtain a naive parametrization of $U(x_\Lambda, \theta)$ by introducing one parameter θ^y to each possible configuration $y \in \{0,1\}^{k\times l}$ and defining

$$U(x_\Lambda, \theta) = \sum_{y\in\{0,1\}^{k\times l}} \theta^y I(x_\Lambda = y) = \theta^{x_\Lambda}, \tag{2}$$

where x_Λ and y are $k \times l$ matrices of zeros and ones, and $I(\cdot)$ is one when the argument is true and zero otherwise. When $k = l = 2$ we have in particular that

$$\theta = \left(\theta^{\begin{smallmatrix}0&0\\0&0\end{smallmatrix}}, \theta^{\begin{smallmatrix}1&0\\0&0\end{smallmatrix}}, ..., \theta^{\begin{smallmatrix}1&1\\1&1\end{smallmatrix}}\right) \in \mathbb{R}^{16}. \tag{3}$$

We refer to the elements of θ as configuration parameters. It is a well known fact that this model is not identifiable, meaning that several different choices of θ give the same model. For example, adding the same value to all configuration parameters will not change the model, as this will be compensated by a corresponding change in the normalizing constant. A less obvious way to change the parameter vector without changing the model, when $k = l = 2$, is for example to add some value to $\theta^{\begin{smallmatrix}1&0\\0&0\end{smallmatrix}}$ and subtract the same value from $\theta^{\begin{smallmatrix}0&1\\0&0\end{smallmatrix}}$, $\theta^{\begin{smallmatrix}0&0\\1&0\end{smallmatrix}}$ or $\theta^{\begin{smallmatrix}0&0\\0&1\end{smallmatrix}}$. In the following we present an alternative representation of $p(x)$ that is clearly always identifiable and use this to find a minimal number of restrictions that needs to be put on θ to make the above parametric model identifiable as well.

Following Tjelmeland and Austad (2012), we note that $U(x_\Lambda, \theta)$ is a pseudo-Boolean function and thereby $p(x)$ can be represented as

$$p(x) = c \exp\left(\sum_{\Lambda \in \mathcal{L}} \beta^\Lambda \prod_{i \in \Lambda} x_i\right), \tag{4}$$

where β^Λ is referred to as the interaction parameter for clique Λ, which is said to be of $|\Lambda|$'th order. More details on pseudo-Boolean functions and their properties can be found in Grabisch et al. (2000) and Hammer and Holzman (1992). Since this representation consists of linearly independent functions of x, it is clear that the model is identifiable when, for example, $\beta^\emptyset$ is fixed. This model represents the most general form of a binary MRF, meaning that all binary MRFs can be represented on this form. To find sufficient restrictions on θ, we first note that the interaction parameters are also translational invariant under the stationary and torus boundary condition assumptions. A proof is included in the supplementary materials of this paper. Next, we establish a one-to-one relation between the θ parameters in (2) and the β parameters in (8).

Let β be the vector of interaction parameters. For instance in the 2×2 clique case we have

$$\beta = \left(\beta^\emptyset, \beta^{\square}, \beta^{\square\square}, \beta^{\begin{smallmatrix}\square\\\square\end{smallmatrix}}, \beta^{\square\,\square}, \beta^{\square\begin{smallmatrix}\square\\\square\end{smallmatrix}}, \beta^{\begin{smallmatrix}\square\\\square\end{smallmatrix}\square}, \beta^{\square\square\begin{smallmatrix}\square\\\square\end{smallmatrix}}, \beta^{\begin{smallmatrix}\square\\\square\end{smallmatrix}\begin{smallmatrix}\square\\\square\end{smallmatrix}}, \beta^{\square\begin{smallmatrix}\square\\\square\end{smallmatrix}\square}, \beta^{\begin{smallmatrix}\square\\\square\end{smallmatrix}\square\square}, \beta^{\begin{smallmatrix}\square\square\\\square\square\end{smallmatrix}}\right),$$

where for instance $\beta^{\square\square}$ denotes the parameter for all horizontally adjacent nodes. Note that we order the elements in this vector by increasing order of $|\Lambda|$. Remembering that one restriction must be put on the β parameters to obtain identifiability we see that 10 free parameters are used in this parametrization, compared to the $2^4 = 16$ configuration parameters in (3). The maximal number of free parameters N_{kl} to be used for maximal cliques of size $k \times l$ for some values of k and l is given in Table 1. In this table we see that the number of free parameters quickly grows as a function of the clique size. Since

$k \times l$	2^{kl}	N_{kl}
1×2	4	2
2×2	16	10
2×3	64	44
3×3	512	400
3×4	4096	3392
4×4	65536	57856

Table 1: The number of free parameters for different $k \times l$ cliques. Also the number of configurations are shown in each case.

the functional spaces of (2) and the exponent of (8) is the same, and the latter model is identifiable, some restrictions must be put on the configuration parameters to make model (7) identifiable as well. In the following we define such a sufficient set of restrictions, and we start, for the 2×2 cliques case, by representing all interaction parameters as functions of the configuration parameters. Later we will see that these restrictions easily generalizes to the $k \times l$ case. In particular we identify the 11×16 matrix A such that $\beta = A\theta$. Finding this relation can be done using a recursive technique where the interaction parameters are calculated in the order they appear in the β vector. That is, we start by calculating $\beta^{\emptyset}$, which can be done by comparing the models (7) and (8) for $x = (0, 0, ..., 0)$,

$$\beta^{\emptyset} = nm\theta^{00}_{00}.$$

Next we calculate $\beta^{\square}$ by evaluating the two models for $x = (0, ..., 1, 0, ..., 0)$, where the position of the 1 is an arbitrary choice in the sense that all choices give the same result. We get

$$\beta^{\emptyset} + \beta^{\square} = \theta^{10}_{00} + \theta^{01}_{00} + \theta^{00}_{10} + \theta^{00}_{01} + (nm - 4)\theta^{00}_{00}$$

and thereby

$$\beta^{\square} = \theta^{10}_{00} + \theta^{01}_{00} + \theta^{00}_{10} + \theta^{00}_{01} - 4\theta^{00}_{00}.$$

Continuing in this way we can establish the rest of the interaction parameters. As already

	$\theta^{\substack{00\\00}}$	$\theta^{\substack{10\\00}}$	$\theta^{\substack{01\\00}}$	$\theta^{\substack{00\\10}}$	$\theta^{\substack{00\\01}}$	$\theta^{\substack{11\\00}}$	$\theta^{\substack{00\\11}}$	$\theta^{\substack{10\\10}}$	$\theta^{\substack{01\\01}}$	$\theta^{\substack{10\\01}}$	$\theta^{\substack{01\\10}}$	$\theta^{\substack{01\\11}}$	$\theta^{\substack{10\\11}}$	$\theta^{\substack{11\\01}}$	$\theta^{\substack{11\\10}}$	$\theta^{\substack{11\\11}}$
$\beta^{\emptyset}$	nm															
$\beta^{\square}$	-4	1	1	1	1											
$\beta^{\square\square}$	2	-1	-1	-1	-1	1	1									
$\beta^{\substack{\square\\\square}}$	2	-1	-1	-1	-1			1	1							
$\beta^{\square\,\square}$	1	-1			-1					1						
$\beta^{\square\,\square}$	1		-1	-1							1					
$\beta^{\square\square}$	-1		1	1	1		-1		-1		-1	1				
$\beta^{\square\square}$	-1	1		1	1		-1	-1		-1			1			
$\beta^{\square\square}$	-1	1	1		1	-1			-1	-1				1		
$\beta^{\square\square}$	-1	1	1	1		-1		-1			-1				1	
$\beta^{\square\square}$	1	-1	-1	-1	-1	1	1	1	1	1	1	-1	-1	-1	-1	1

Figure 1: The matrix A between the clique parameters θ and the interaction parameters β in the case with maximal cliques of size 2×2. Empty cells represents the value 0.

mentioned this system of equations can be written as $\beta = A\theta$, and the matrix A for the 2×2 case is shown in Figure 1. The ordering we choose on θ is the number of ones present in each configuration y in θ^y, from no ones to only ones. For configurations with the same number of ones, the ordering is made according to the ordering of the corresponding β parameters. For instance, since $\beta^{\square\square}$ appear before $\beta^{\substack{\square\\\square}}$ we have $\theta^{\substack{11\\00}}$ and $\theta^{\substack{00\\11}}$ before $\theta^{\substack{10\\01}}$. The ordering for the two former is however arbitrary. This ordering give A the lower triangular like shape seen in Figure 1. In order to have identifiability, restrictions must be set on the configuration parameters such that there is a one-to-one relation between θ and β. One possibility is to define a matrix B that constrains θ by a parameter vector ϕ such that $\theta = B\phi$ where B has dimension $2^{kl} \times (N_{kl} + 1)$ and $\phi \in \mathbb{R}^{(N_{kl}+1)}$ in the general case. In the following we illustrate for a 2×2 clique how we construct B making sure that ϕ gives an identifiable model. The generalization to a $k \times l$ clique will follow. To ensure identifiability one can define B such that the matrix AB becomes square and lower triangular. As we can see in Figure 1, as a product of the ordering of the elements we have chosen in β and θ, the system of equations already have a shape that is close to being lower triangular. Starting with the equation for $\beta^{\emptyset}$

in Figure 1 and moving down the rows we see that one or more new configuration parameters are introduced in each row. That is, for each row in the matrix there exists one or more elements from θ that gets non-zero coefficients for the first time. Our strategy is to give all new parameters that are introduced in row i the same value ϕ_i, $i = 0, ..., 10$. We write $\phi = (\phi_0, ..., \phi_{10})$, and by using this constrained parametrization we have defined a square and lower triangular matrix AB. The matrix B which gives this result is easy to define. This matrix simply consists of only 0's except for one entry with the value 1 in each row. For row i this entry picks out the element in ϕ that θ_i should equal. For instance, the rows 2-5 will have a non-zero entry at the second position in order to get $\theta^{10}_{00} = \theta^{01}_{00} = \theta^{00}_{10} = \theta^{00}_{01} = \phi_1$, while the tenth row will have a one at position 5 to obtain $\theta^{10}_{01} = \phi_4$. As we can see the choice we make are in the construction of B, and one could imagine different choices being made here, for instance constraining some of the θ parameters to equal zero. However, the choice made here is intuitive, easy to construct and, as we will see next, easily generalized to a $k \times l$ clique.

One way to obtain this solution also for a $k \times l$ clique is to think of the value 0 as a background color, and focus on the position of the nodes with value 1 in the configuration, which we will think of as an object. If we define the configurations to be translation invariant with respect to the position of the nodes with value 1, we obtain the solution illustrated for the 2×2 clique above. In other words, all configurations where exactly the same object of nodes with value 1 appear but at different positions in the $k \times l$ block will be assigned the same configuration parameter. For instance in the 3×3 case we get

$$\theta^{\substack{110\\100}}_{000} = \theta^{\substack{011\\010}}_{000} = \theta^{\substack{000\\110}}_{100} = \theta^{\substack{000\\011}}_{010}.$$

We may now write the potential function as $U(x_\Lambda, B\phi)$. Note that one more restriction on

$$c_0 = \left\{ \binom{00}{00} \right\},\ c_1 = \left\{ \binom{10}{00}, \binom{01}{00}, \binom{00}{10}, \binom{00}{01} \right\},\ c_2 = \left\{ \binom{11}{00}, \binom{00}{11} \right\},$$
$$c_3 = \left\{ \binom{10}{10}, \binom{01}{01} \right\}\ c_4 = \left\{ \binom{10}{01} \right\},\ c_5 = \left\{ \binom{01}{10} \right\},\ c_6 = \left\{ \binom{11}{10} \right\},$$
$$c_7 = \left\{ \binom{11}{01} \right\},\ c_8 = \left\{ \binom{10}{11} \right\},\ c_9 = \left\{ \binom{01}{11} \right\},\ c_{10} = \left\{ \binom{11}{11} \right\}$$

Figure 2: All the configuration sets for a binary 2×2 clique.

ϕ is still needed in order to make the model identifiable. Our choice for this last restriction is given later.

In the following we refer to all $k \times l$ configurations that are assigned the same configuration parameter as a configuration set. We denote these sets by $c_0, ..., c_{N_{kl}}$. For instance in the 2×2 case $N_{22} = 10$ and all the configuration sets for this case can be seen in Figure 2. We refer to the elements of ϕ as configuration set parameters.

We end this section with a discussion on how the above torus MRF and associated prior can be modified to the free boundary case. One should first note that for a free boundary MRF, a translation invariance property of the potential functions will not be transferred to a corresponding translation invariance for the interaction parameters, and neither will such a model be stationary. Moreover, the restrictions we identified for the θ parameters in the torus boundary condition case no longer apply. However, the extra free θ parameters that may be introduced in the free boundary case will only model properties sufficiently close to a boundary of the lattice. Our strategy in the free boundary case is to keep the same θ parameter vector and translational invariant potential functions $U(x_\Lambda, \theta)$ for all $k \times l$ cliques as in the torus boundary condition case, but to add non-zero potential functions for some (non-maximal) cliques at the boundaries of the lattice. Our motivation for including

non-zero potential functions for some cliques at the boundaries of the lattice is to reduce the boundary effect and, hopefully, get a model which is less non-stationary. To define our non-zero potential functions at the boundaries, imagine that our $n \times m$ lattice is included in a much larger lattice and that this extended lattice also has maximal cliques that are blocks of $k \times l$ nodes. We then include a non-zero potential function for every $k \times l$ clique in the extended lattice which is partly inside and partly outside our $n \times m$ lattice. In such a $k \times l$ clique, let Λ denote the set of nodes that are inside our $n \times m$, and let λ denote the set of nodes outside. As we have assumed that the $k \times l$ clique is partly inside and partly outside our $n \times m$ lattice, Λ and λ are both non-empty and $\Lambda \cup \lambda$ is clearly a maximal clique in the extended lattice. For the (non-maximal) clique Λ we define the potential function

$$U_\Lambda(x_\Lambda, \theta) = \frac{1}{2^{|\lambda|}} \sum_{x_\lambda} U_{\Lambda \cup \lambda}(x_{\Lambda \cup \lambda}, \theta), \tag{5}$$

where $U_{\Lambda \cup \lambda}(x_{\Lambda \cup \lambda}, \theta)$ is the same (translational invariant) potential function we are using for maximal cliques inside our $n \times m$ lattice. One can note that (5) corresponds to averaging over the values in the nodes outside our lattice, assuming them to be independent, and to take the values 0 or 1 with probability a half for each.

2.2 Example 1: The independence model

Consider a model where the variables are all independent of each other and $P(x_i) = p^{x_i}(1-p)^{1-x_i}$ for each i and where p is the probability of x_i being equal to 1. We get

$$p(x) = \prod_{i=1}^{nm} p^{x_i}(1-p)^{1-x_i} \propto \exp\left(\alpha \sum_{i=1}^{nm} x_i\right),$$

where

$$\alpha = \ln\left(\frac{p}{1-p}\right).$$

We use the independence model as an example also later in the paper, and in particular we fit an MRF with 2×2 cliques to data simulated from this model. Therefore it is helpful to know how one can represent the independence model using 2×2 cliques. This can easily be done by first writing the independence model on the form in (8), which we can do using the recursive technique described above. Next one can solve for the configuration set parameters for the configuration sets given in Figure 2 by comparing the interaction parameters for the two models. Starting with the equations for $\beta^{\emptyset}$ we get

$$n\phi_0 = \alpha \cdot 0 = 0 \;\Rightarrow\; \phi_0 = 0.$$

Next, one can solve for ϕ_1 by comparing the equations for $\beta^{\square}$,

$$4\phi_1 - 4\phi_0 = \alpha \;\Rightarrow \phi_1 = \frac{\alpha}{4},$$

and continue in this way until all ϕ_i $i = 0, ..., 10$ are found. As already mentioned adding a constant η to the obtained solution does not change the distribution of interest. The full solution may be written as $\phi_0 = \eta$, $\phi_1 = \alpha/4 + \eta$, $\phi_2 = ... = \phi_5 = \alpha/2 + \eta$, $\phi_6 = ... = \phi_9 = 3\alpha/4 + \eta$ and $\phi_{10} = \alpha + \eta$. How we choose η will be given later in the paper. When $p = 0.5$ we see that $\alpha = 0$, and we get in fact that all configuration set parameters should be equal.

2.3 Example 2: The Ising model

The Ising model (Besag 1986) is given by

$$p(x) = \frac{1}{c} \exp\left\{ -\omega \sum_{i \sim j} I(x_i \neq x_j) \right\},$$

where the sum is over all horizontally and vertically adjacent sites, and ω is a parameter controlling the probability of adjacent sites having the same value. The same strategy as in the previous section can be used in order to represent the Ising model with 2×2 cliques.

The Ising model is obtained by setting the configuration set parameters for the configuration sets in Figure 2 equal to $\phi_0 = \phi_{10} = \eta$, $\phi_1 = \phi_2 = \phi_3 = \phi_6 = \phi_7 = \phi_8 = \phi_9 = -\omega + \eta$, and $\phi_4 = \phi_5 = -2\omega + \eta$.

3. PRIOR SPECIFICATION

In this section we define a generic prior for the parameters of an MRF with $k \times l$ cliques. The first step in the specification of the prior is to choose what parametrization of the MRF to consider. In the previous section we introduced three parametrizations for the MRF, with parameter vectors θ, β and ϕ, respectively. When choosing between these three parametrizations and defining the prior we primarily have the torus version of the MRF in mind. However, as the free boundary version of the model is using the same parameter vectors, the prior we end up with can also be used in that case. As discussed above, the parametrization using θ is grossly overparameterized and it is thereby not natural to focus on this formulation to define a prior. It should also be remembered that the parametrizations using β and ϕ are non-identifiable, but here it is sufficient to add one restriction to make any of these models identifiable. The perhaps easiest way to make the models identifiable is to restrict one of the parameters to equal zero, but other alternatives also exist. We return to this issue below. From Table 1 we see that for the ϕ and β parametrizations the dimension of the parameter vectors grow rapidly with k and l. It is therefore natural to look for prior formulations which include the possibility for a reduced number of free parameters. For the β parametrization the perhaps most natural strategy to obtain this is to assign positive prior probability to the event that one or several of the interaction parameters are exactly zero. The interpretation of the ϕ parameter is different from the interpretation of the β parameter, and

it is not natural to assign positive probability for elements of the ϕ vector to be exactly zero. A more reasonable scheme here is to set positive prior probability for the event that groups of configuration parameters have exactly the same value. In addition to specify the probabilities for some elements in β to be exactly zero, or the probabilities for groups of elements in ϕ to be equal, one also needs to specify a prior density for the (non-zero) parameter values. As mentioned above the interpretation of the β and ϕ parameter differ substantially, and thereby a prior for their value should also differ. We find it natural to assume apriori that all elements in ϕ are on the same scale and, unless particular prior information is available and suggests the opposite, we find it natural to choose a prior where the elements of the ϕ vector are exchangeable. The interpretation of the β parameter is more complex than for the ϕ parameters. In particular we think higher order interaction parameters apriori should tend to take smaller values than lower order interaction parameters. This makes it more difficult to specify a reasonable prior for β than for ϕ. In the following we therefore focus on specifying a prior for ϕ. We first introduce the notation necessary to define the groups of configuration parameters that should have the same value and thereafter discuss possibilities for how to define the prior.

To define groups of configuration set parameters that should have the same value, let $C_1, \ldots, C_r$ be a partition of the set of all configuration sets, with $C_i \neq \emptyset$ for $i = 1, \ldots, r$. Thus, $C_i \cap C_j = \emptyset$ for $i \neq j$ and $C_1 \cup \ldots \cup C_r = \{c_0, \ldots, c_{N_{kl}}\}$. Let φ_i denote the common value for ϕ_j for all $c_j \in C_i$, and let $z = \{(C_i, \varphi_i), i = 1, \ldots, r\}$. Thus, for $i = 0, \ldots, N_{kl}$ we have

$$\phi_i = \sum_{(C,\varphi)\in z} \varphi I(c_i \in C).$$

Let Φ denote the function so that $\phi = \Phi(z)$. We define a prior on ϕ by specifying a prior for z. An alternative to this construction would be to build up $\{C_1, ..., C_r\}$ in a non-random fashion,

constraining ϕ according to properties like symmetric and rotational invariance. However with our prior distribution such properties can be inferred from data. The potential function may now be written as $U(x_\Lambda, B\phi) = U(x_\Lambda, B\Phi(z))$.

Given all configuration sets in a $k \times l$ clique, we want to assign positive probability to the event that groups of configuration sets have exactly the same parameter value. For instance, the 3 groups indicated in Section 2.3 is an example of such a grouping for a 2×2 clique. Since no groups can be empty, the maximum number of groups one can get is $N_{kl} + 1$. Our prior distribution for z will be on the form

$$p(z) = p(\{C_1, ..., C_r\})p(\{\varphi_1, ..., \varphi_r\}|r)$$

where $p(\{C_1, ..., C_r\})$ is the prior for the grouping of the configuration sets, while $p(\{\varphi_1, ..., \varphi_r\}|r)$ is the prior distribution on the group parameters given the number of groups r. Two possibilities for the prior distribution for $\{C_1, ..., C_r\}$ immediately comes to mind. The first alternative is to assume a uniform distribution on the groupings,

$$p_1(\{C_1, ..., C_r\}) \propto const,$$

meaning that each grouping is apriori equally likely. However for $p(r)$, the marginal probability of the number of groups, this means that most of the probability is put on groupings with approximately $(N_{kl} + 1)/2$ groups. In fact the probability $p(r)$ becomes equal to

$$p(r) = \frac{g(N_{kl} + 1, r)}{\sum_{i=1}^{N_{kl}+1} g(N_{kl} + 1, i)},$$

where $g(N_{kl} + 1, r)$ is the number of ways $N_{kl} + 1$ configuration sets can be organized into r unordered groups, remembering that no empty groups are allowed. The function $g(N_{kl}+1, r)$ is easily deduced to be

$$g(N_{kl} + 1, r) = \frac{1}{r!} \sum_{i=0}^{r} \binom{r}{i} (-1)^{r-i} i^{N_{kl}+1},$$

where each term in the sum is equal to the number of ways $N_{kl}+1$ configuration sets can be organized into r ordered groups allowing for empty groups. For the 2×2 clique this means for instance that $p(r=1) = p(r=11) \approx 10^{-6}$ while $p(r=5) = 0.36$. The second alternative for $p(\{C_1, ..., C_r\})$ is to make the distribution for the number of groups uniform. This can be done by defining the probability distribution

$$p_2(\{C_1, ..., C_r\}) = \frac{1}{(N_{kl}+1)g(N_{kl}+1, r)}.$$

With this prior a particular grouping with many or few groups will have a larger probability than a particular grouping with approximately $(N_{kl}+1)/2$ groups. For instance in the 2×2 case the probability of the grouping where all configuration sets are assigned to the same group or the grouping with 11 groups is $p(\{C_1\}) = p(\{C_1, ..., C_{11}\}) = 0.09$, while the probability of a particular grouping with 5 groups is $p(\{C_1, ..., C_5\}) \approx 10^{-7}$. Observe however, that with both prior distributions we have the property that given the number of groups the grouping is uniformly distributed. As a compromise between these two prior distributions we propose

$$p(\{C_1, ..., C_r\}) \propto p_1(\{C_1, ..., C_r\})^{1-\gamma} p_2(\{C_1, ..., C_r\})^{\gamma},$$

where $0 \leq \gamma \leq 1$. Given z we assume in the reminder of the paper that

$$\sum_{(C,\varphi) \in z} \varphi = 0, \tag{6}$$

in order to obtain identifiability. This choice also gives us an exchangeable model in opposite to for instance fixing one configuration set parameter to a constant. Restricted to this sum-to-zero property we assume independent zero mean normal priors with variance σ_φ^2 for all the group parameters. This fully defines the prior for z, except that we have not specified values for the two hyper-parameters γ and σ_φ^2.

4. MCMC SAMPLING FROM A POSTERIOR DISTRIBUTION

In this section we assume that an observed binary $n \times m$ image is available. We consider this image as a realization from our MRF with free boundary conditions defined in Section 2. As prior for the MRF parameters we adopt the prior specified in Section 3. The focus in this section is then on how to sample from the resulting posterior distribution. Letting x denote the observed image, the posterior distribution we want to sample from is given by

$$p(z|x) \propto p(x|B\Phi(z))p(z),$$

where $p(x|B\Phi(z))$ and $p(z)$ are the MRF from Section 2 and the prior from Section 3, respectively. To simulate from this posterior distribution we adopt a reversible jump Markov chain Monte Carlo (RJMCMC) algorithm (Green 1995) with three types of updates. The detailed proposal mechanisms are specified in the supplementary materials, here we just give a brief description of our proposal strategies.

The first proposal in our algorithm is simply first to propose a change in an existing φ parameter by a random walk proposal with variance σ^2, and thereafter to subtract the same value from all φ parameters to commit with the sum-to-zero constraint. In the second proposal we draw a pair of groups and propose to move one configuration set from the first group to the second group, ensuring that the two groups are still non-empty. In the last proposal type, we propose a new state by either increasing or decreasing the number of groups with one. When increasing the number of groups by one we randomly choose a configuration set from a randomly chosen group and propose this configuration set to be a new group. When proposing to reduce the number of parameters with one, we randomly choose a group with only one configuration set and propose to merge this group into another

group. In the trans-dimensional proposals we ensure that the proposed parameters commit with the sum-to-zero constrain by subtracting the same value from all φ parameters.

5. SIMULATION EXAMPLES

In this section we first present two examples based on simulated data sets, and thereafter present results for a data set of census counts of red dear in the Grampians Region of north-east Scotland. In all the simulation experiments we use the prior distribution as defined in Section 3. In this prior the values of the two hyper-parameters σ_φ and γ must be specified. We have fixed $\sigma_\varphi = 10$ and tried $\gamma = 0$, 0.5 and 1. When discussing simulation results we first present results for $\gamma = 0.5$. As likelihood function we use the MRF discussed in Section 2 and we use 2×2 cliques except in the last part of the real data example where we also discuss results for 3×3 cliques. To cope with the computationally intractable normalizing constant of the MRF likelihoods, we adopt the approximation strategy of Tjelmeland and Austad (2012). The MRF is then approximated with a partially ordered Markov model (POMM), see Cressie and Davidson (1998), where the conditional distribution of one variable given all previous values is allowed to depend on maximally ν previous values. We have tried different values for ν and found that in all our examples $\nu = 7$ is sufficient to obtain very good approximations, so all the results presented here are based on this value of ν. To simulate from posterior distributions we use the reversible jump MCMC algorithm defined in Section 4. In our sampling algorithm we have an algorithmic tuning parameter σ^2 as the variance in Gaussian proposals. Based on the results of some preliminary runs we set $\sigma = 0.3$. Note also that one iteration of our sampling algorithm is defined to be one of each proposal type. Lastly we note that parallel computing was used in order to reduce computational time, and

the technique that is used is explained in the supplementary materials.

5.1 The independence model

We generate a realization from the independence model in Section 2.2 with $p = 0.3$ on a 100×100 lattice, consider this as our observed data x, and simulate by the MCMC algorithm defined in Section 4 from the resulting posterior distribution. Using the notation for the configuration sets in a 2×2 clique defined in Figure 2 and the results from Section 2.2, we ideally want our algorithm to produce realizations with the grouping $\{c_0\}$, $\{c_1\}$, $\{c_2, c_3, c_4, c_5\}$, $\{c_6, c_7, c_8, c_9\}$, $\{c_{10}\}$. Note that due to our identifiablity restriction in (6) the configuration set parameters should be close to the solution from Section 2.2 with $\eta = -\alpha/2$. To check convergence we investigated trace plots of various statistics, see the supplementary material, and these investigations show that the algorithm converges very quickly. The acceptance rate for the parameter value proposals is 24%, whereas the acceptance rates for the other two types of proposals are both around 2%. We run our sampling algorithm for 20000 iterations, and estimate the posterior probability of the number of groups. The configuration sets are organized into 4 (77%), 5 (21%) or 6 (2%) groups, so for these data the grouping tends to be a little bit too strong compared to the correct number of groups. This can also be seen from the estimated posterior probability of two configuration sets being assigned to the same group, shown in Figure 3. This figure suggests the four groups $\{c_0\}, \{c_1\}, \{c_2, c_3, c_4, c_5, c_7\}$, $\{c_6, c_8, c_9, c_{10}\}$ which is also calculated to be the most probable grouping estimated by counting the number of occurences. In fact the posterior probability for this grouping is as high as 55%. In Figure 3 we also see how the most probable grouping differ from the correct model grouping, shown in grey. The group $\{c_6, c_7, c_8, c_9\}$ in the correct model is split in the most probable grouping, and the subsets $\{c_7\}$ and $\{c_6, c_8, c_9\}$ are inserted

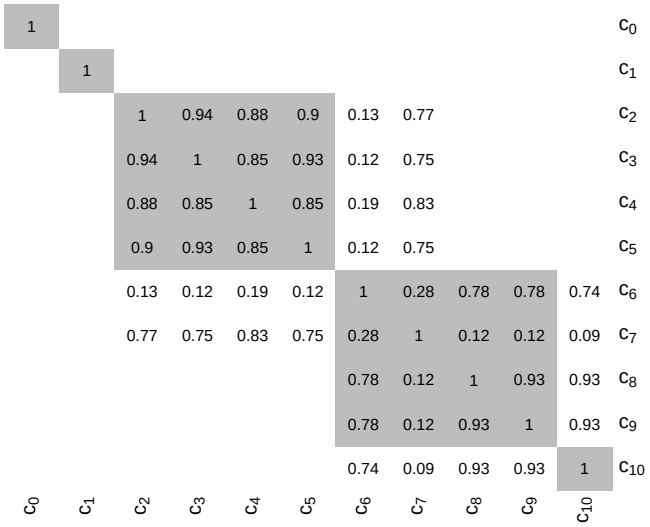
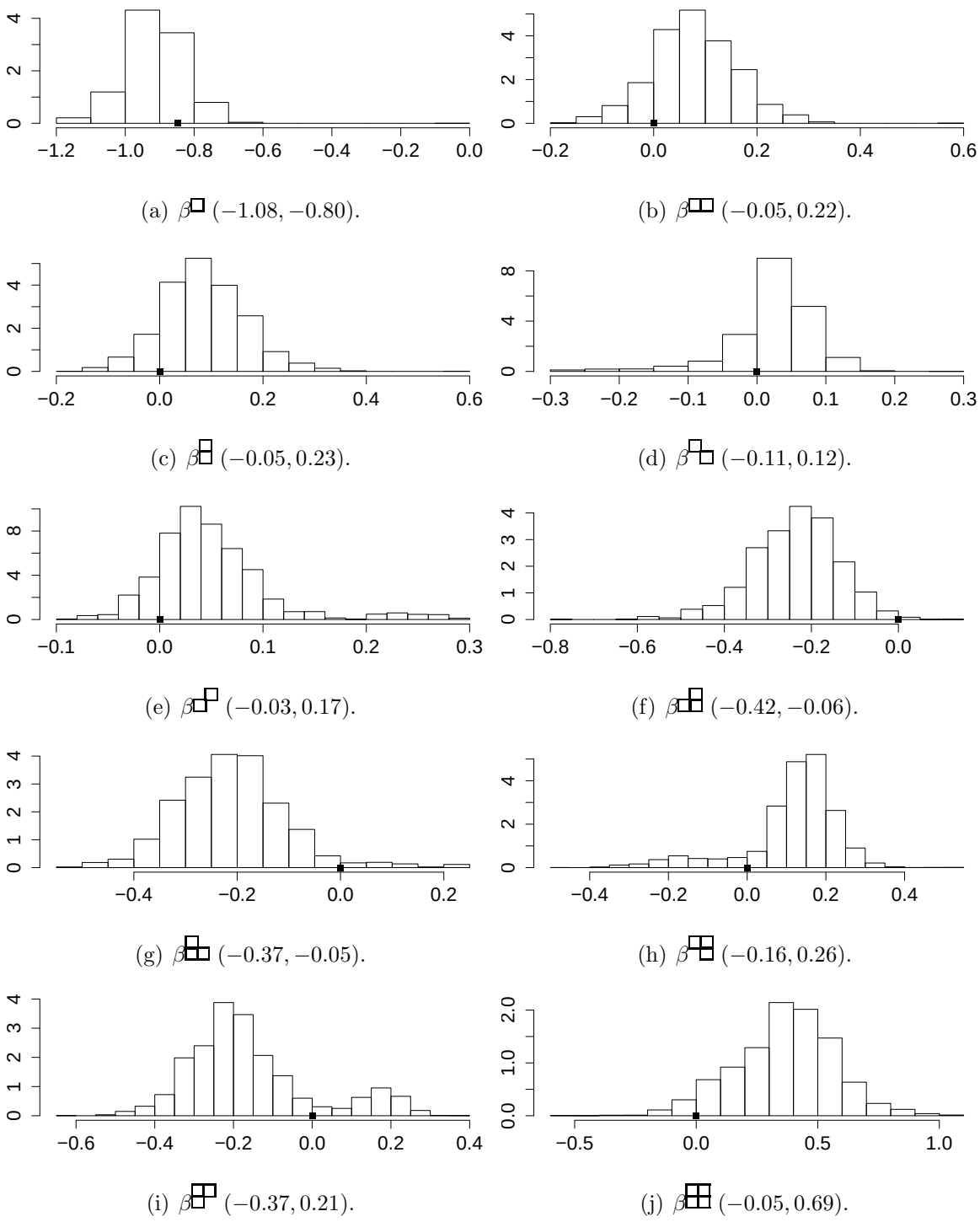


Figure 3: Independence model example: Estimated posterior probabilities for two configuration sets to be grouped together. The correct grouping is shown in grey, and only probabilities larger than 5% are given.

into the correct model groups $\{c_2, c_3, c_4, c_5\}$ and $\{c_{10}\}$, respectively.

One informative way to look at the result of the simulation is to estimate the posterior distribution for the interaction parameters β. Histograms and estimated 95% credibility intervals for each of the parameters are given in Figure 4. As we can see, the true value of the interaction parameters are mostly within the credibility intervals, but the tendency to group the configurations too much is in this case forcing some of the true values into a tail of the marginal posterior distributions.

To study the properties of the MRF $p(\cdot|B\Phi(z))$ when z is a sample from the posterior $p(z|x)$ we take 5000 samples from the MCMC run for $p(z|x)$ and generate for each of these a corresponding realisation from the MRF $p(\cdot|B\Phi(z))$. To analyze these 5000 images we use six statistics describing local properties of the images. The statistics used and resulting



(a) $\beta^{\square}$ $(-1.08, -0.80)$.

(b) $\beta^{\square\square}$ $(-0.05, 0.22)$.

(c) $\beta^{\square\!\square}$ $(-0.05, 0.23)$.

(d) $\beta^{\square\,\square}$ $(-0.11, 0.12)$.

(e) $\beta^{\square\!\square}$ $(-0.03, 0.17)$.

(f) $\beta^{\square\!\square}$ $(-0.42, -0.06)$.

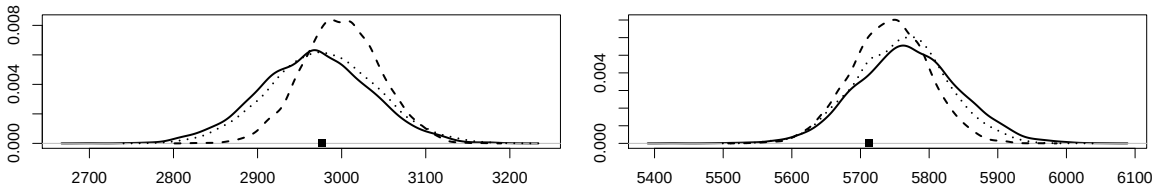
(g) $\beta^{\square\!\square}$ $(-0.37, -0.05)$.

(h) $\beta^{\square\!\square}$ $(-0.16, 0.26)$.

(i) $\beta^{\square\!\square}$ $(-0.37, 0.21)$.

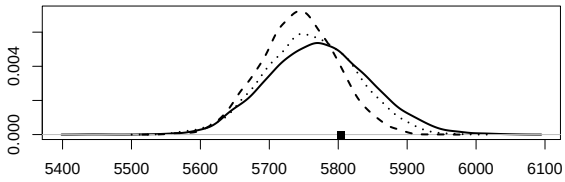
(j) $\beta^{\square\!\square}$ $(-0.05, 0.69)$.

Figure 4: Independence model example: Estimated marginal posterior distribution of the interaction parameters. True values are shown with a black point and estimated 95% credibility interval is given for each parameter.

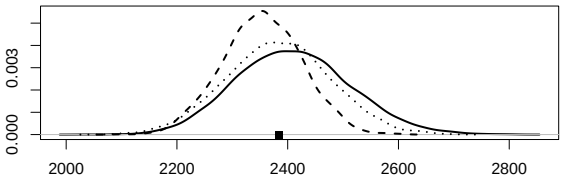


(a) $g(x) = \sum_i x_i$

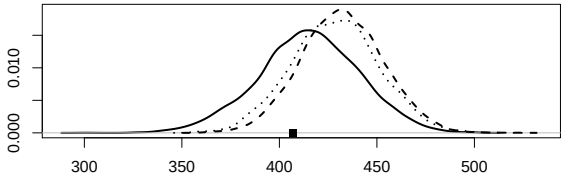
(b) $g(x) = \sum_{i,j:\text{vertical adjacent sites}} I(x_i = x_j)$



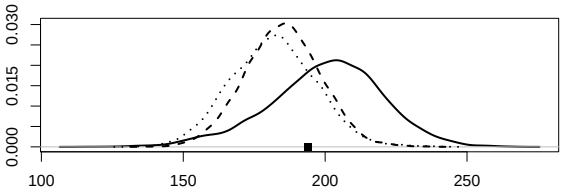
(c) $g(x) = \sum_{i,j:\text{horizontal adjacent sites}} I(x_i = x_j)$



(d) $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \binom{00}{00}\right)$



(e) $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \binom{01}{10}\right)$



(f) $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \binom{11}{10}\right)$

Figure 5: Independence model example: Distribution of six statistics of realizations from our 2×2 model with posterior samples of z (solid), the independence model with correct parameter value (dashed), and the independence model with posterior samples of the parameter value (dotted). The data evaluated with each statistic is shown with a black point.

density estimates (solid) of the distribution of these statistics are shown in Figure 5. In the same figure we also show density estimates of the same statistics when images are generated from the independence model with the true parameter value (dashed), and when images are generated from the independence model with parameter value α generated from the posterior distribution given our observed image x (dotted). In this last case, a zero mean Gaussian prior distribution with standard deviation equal to 10 is used for α. As we can see, our model captures approximately the correct distribution of the chosen statistics. It is interesting to

note that for some statistics the realizations from the independence model with simulated α values follows our model tightly whereas for the other statistics it is close to the correct model.

All the above results are for $\gamma = 0.5$, but as mentioned in the introduction of this section we also investigate the results for $\gamma = 0$ and 1. For $\gamma = 0$ the configuration sets are organized into 4 (75%), 5 (23%) or 6 (2%) groups, and for $\gamma = 1$ we get 4 (93%), 5 (6%), 6 (1%) groups. From these number we see the effect of varying γ. Particularly when increasing γ from 0.5 to 1.0 for this data set, the tendency to group more configuration sets together becomes stronger.

We also did experiments were the value of p was changed. If the value of p is close to 0.5 the tendency to group the configurations too much becomes stronger. This makes perfectly sense, since the correct grouping for $p = 0.5$ is to put all configuration sets into only one group. In the other end, choosing p closer to 0 or 1 gives a stronger tendency to group the configurations according to the correct solution. This illustrates the fact that the algorithm tries to find a good model for the data using as few groups as possible, but as the difference between the true parameter values of the groups becomes larger the price to pay for choosing a model with fewer parameters increases.

5.2 The Ising model

We then repeat the same simulation exercise as above for an Ising model with $\omega = 0.4$. Thus, we generate a realization from the Ising model with $\omega = 0.4$ on a 100×100 lattice, consider this as our observed data x and simulate by the MCMC algorithm from the resulting posterior distribution. The x was obtained using the perfect sampler presented in Propp and Wilson (1996). From the calculations in Section 2.3 we ideally want the correct grouping, $\{c_0, c_{10}\}$

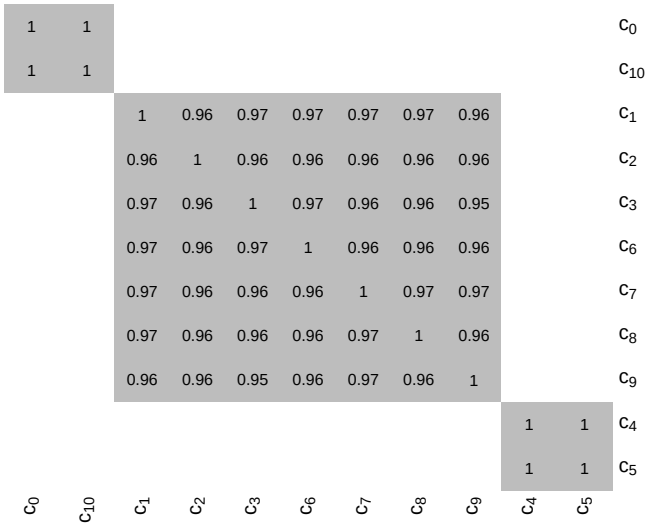


Figure 6: Ising model example: Estimated posterior probabilities for two configuration sets to be grouped together. The true grouping is shown in grey, and only probabilities larger than 5% are given. Note the permutation done to the ordering of the configuration sets c_i.

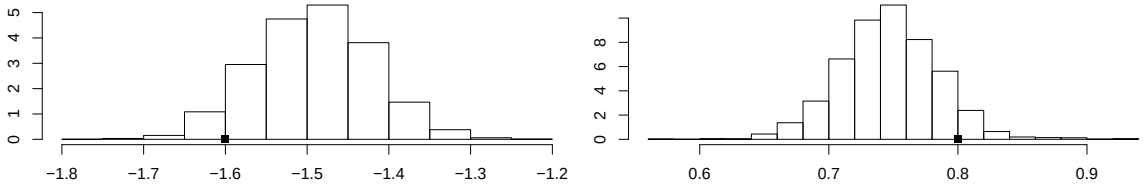
$\{c_1, c_2, c_3, c_6, c_7, c_8, c_9\}$, and $\{c_4, c_5\}$, to be visited frequently by our sampler. Again we run our sampler for 20000 iterations and study the simulation results after convergence. The acceptance rate for the parameter value proposals is 19%, whereas the acceptance rates for the other two types of proposals are both around 1%. The estimated distribution for the number of groups is 94%, 5% and 1%, for 3, 4 and 5 groups respectively.

In Figure 6 we have plotted the matrix representing the estimated posterior probability of two configuration sets being assigned to the same group. As we can see in this figure, the configuration sets are separated into 3 groups, and these groups correspond to the correct grouping shown in grey. About 94% of the realizations is assigned to this particular grouping, and almost all other groupings that are simulated correspond to groupings where the middle group is split in various ways, while some very few are splits of the groups $\{c_1, c_{10}\}$ and

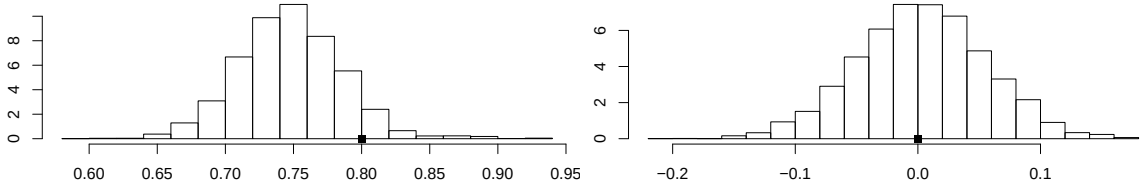
$\{c_4, c_5\}$. Every one of these alternative groupings have an estimated posterior probability of less than 0.5%.

As in the previous example we estimate the posterior distribution for the interaction parameters, see Figure 7. As we can see, all the true values of the interaction parameters are within the estimated credibility intervals, however the mode of the distribution for the pairwise horizontal and vertical second order interaction, see Figure 7(b) and 7(c), seems to be somewhat lower than the correct value. As in the first example we compare the distribution of the same six statistics from simulations from our 2×2 model with posterior samples of z, the Ising model with correct parameter value, and the Ising model with parameter value obtained by posterior sampling, see Figure 8. In this figure we also see that the data we use for posterior sampling (black points) of z is a realization from the Ising model with low values for the number of equal horizontal and vertical adjacent sites, see Figure 8(b) and 8(c), which causes, as already observed above, our simulations of the second order interactions between horizontal and vertical adjacent sites to be somewhat lower than the true value, see Figure 7(b) and 7(c). In fact we can see that the simulations from the Ising model using posterior samples for the parameter value closely follows that of our 2×2 model. This means that the results from our model is as accurate as the result one gets when knowing that the true model is the Ising model without knowing the model parameter.

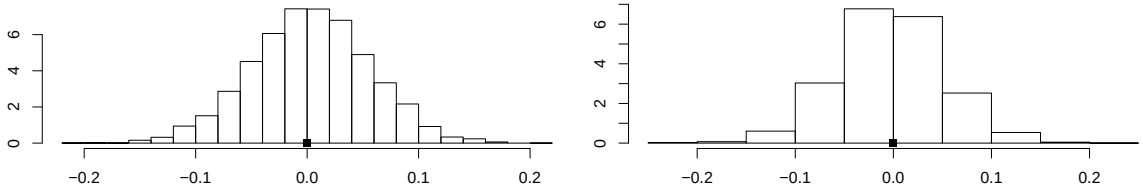
Also for this data set we ran our sampling algorithm in the cases where $\gamma = 0$ and 1. For $\gamma = 0$ the configuration sets are organized into 3 (66%), 4 (31%) or 5 (3%) groups, and for $\gamma = 1$ we get 3 (96%), 4 (4%) groups. As expected we again see the tendency towards stronger grouping when γ is increased.



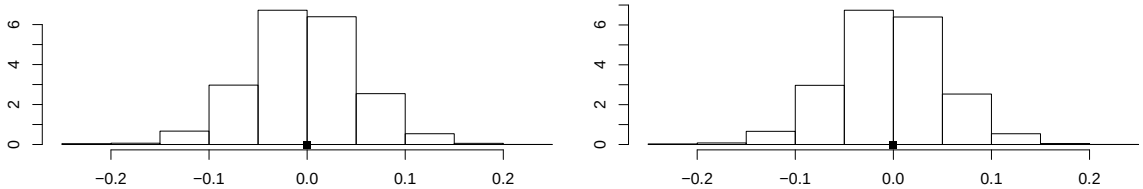
(a) $\beta^{\square}$ $(-1.62, -1.38)$. (b) $\beta^{\square\square}$ $(0.69, 0.81)$.



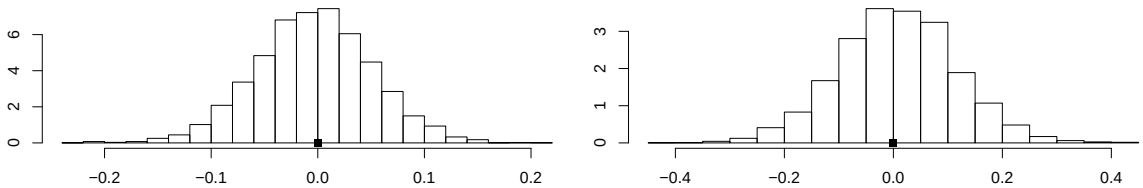
(c) $\beta^{\square}$ $(0.69, 0.81)$. (d) $\beta^{\square}$ $(-0.09, 0.09)$.



(e) $\beta^{\square}$ $(-0.09, 0.09)$. (f) $\beta^{\square}$ $(-0.09, 0.09)$.

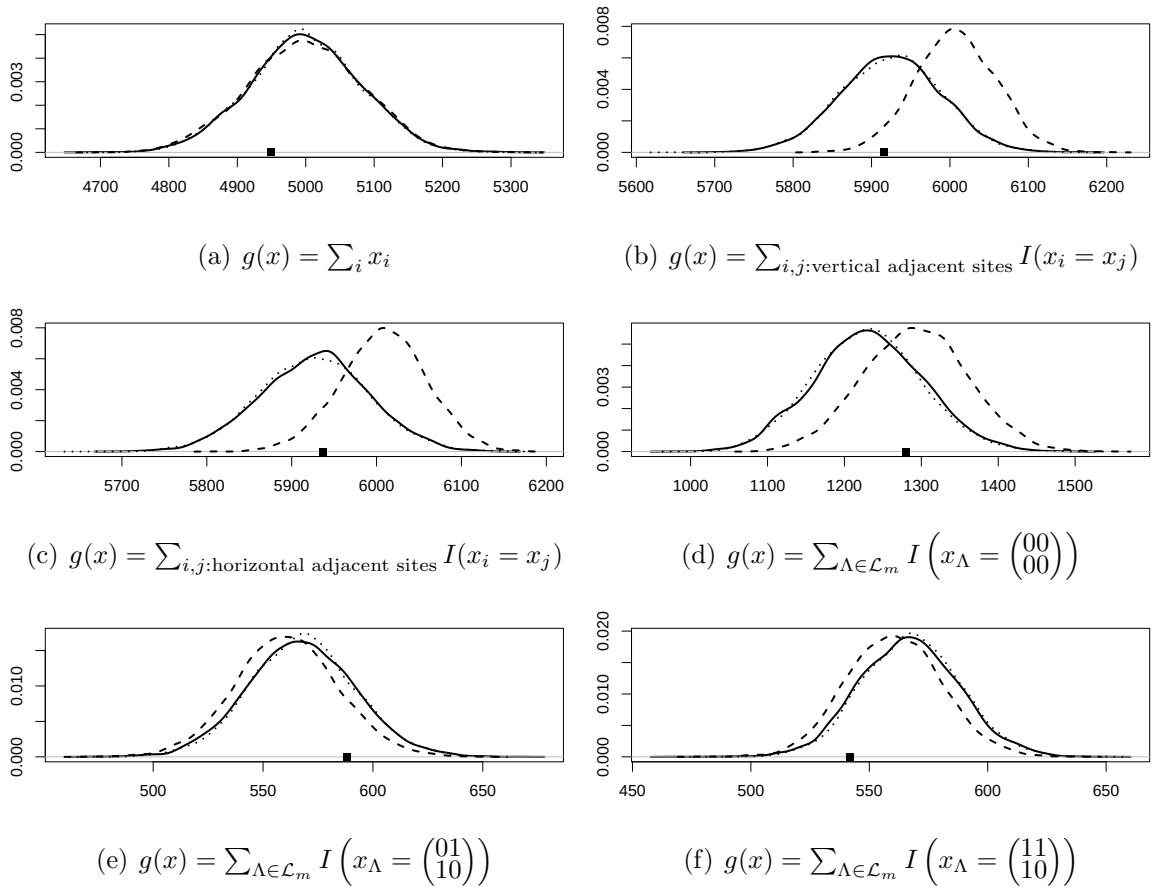


(g) $\beta^{\square}$ $(-0.10, 0.09)$. (h) $\beta^{\square}$ $(-0.09, 0.09)$.



(i) $\beta^{\square}$ $(-0.10, 0.08)$. (j) $\beta^{\square}$ $(-0.17, 0.19)$.

Figure 7: Ising model example: Estimated marginal posterior distribution for the interaction parameters. True values are shown with a black point and estimated 95% credibility interval is given for each parameter.



(a) $g(x) = \sum_i x_i$

(b) $g(x) = \sum_{i,j:\text{vertical adjacent sites}} I(x_i = x_j)$

(c) $g(x) = \sum_{i,j:\text{horizontal adjacent sites}} I(x_i = x_j)$

(d) $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \binom{00}{00}\right)$

(e) $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \binom{01}{10}\right)$

(f) $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \binom{11}{10}\right)$

Figure 8: Ising model example: Distribution of six statistics of realizations from our 2×2 model with posterior samples of z (solid), the Ising model with correct parameter value (dashed), and the Ising model with posterior samples of the parameter value (dotted). The data evaluated with each statistic is shown with a black point.

5.3 Red deer census count data

In this section we analyse a data set of census counts of red deer in the Grampians Region of north-east Scotland. A full description of the data set is found in Augustin et al. (1996) and Buckland and Elston (1993). The data is obtained by dividing the region of interest into $n = 1277$ grid cells on a lattice and observing the presence or absence of red deer in each cell. In our notation this is our observed image x, but in this example we also have the four covariates altitude, mires, north coordinate and east coordinate available in each

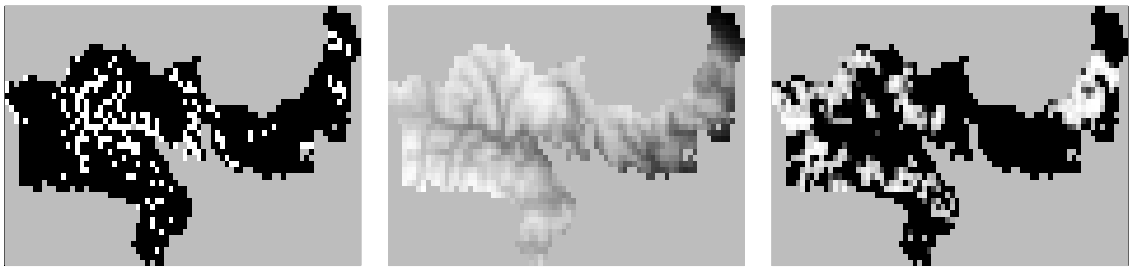


Figure 9: Red deer example: The presence/absence of red deer (left), altitude (middle), and mires (right) in the Grampians Region of north-east Scotland.

grid cell. The binary data x and the two first covariates are shown in Figure 9. We denote the covariate j at each location i by $y_{ij},\ j = 1, 2, 3, 4$, and model them into the likelihood function in the following way

$$p(x|B\Phi(z), \theta^C, y) = \frac{1}{c} \exp\left(\sum_{\Lambda \in \mathcal{L}_m} U(x_\Lambda, B\Phi(z)) + \sum_{i=1}^{n} x_i \sum_{j=1}^{4} \theta_j^C y_{ij} \right),$$

where $\theta^C = (\theta_1^C, ..., \theta_4^C)$ are the parameters for the covariates.

We put independent zero mean Gaussian prior distribution with standard deviation equal to 10 on θ_j^C, $j = 1, ..., 4$. In the sampling algorithm these covariates are updated using random walk, i.e. we uniformly choose one of the four covariates to update and propose a new value using a Gaussian distribution with the old parameter value as the mean and a standard deviation of 0.1.

We ran our algorithm for 50000 iterations, and the acceptance rates for the parameter random walk proposal is 42 %, the group changing proposal is 33%, the trans-dimensional proposal is 5 %, and the covariate proposal is 48 %. The posterior most probable grouping becomes $\{c_0\}$, $\{c_1, ..., c_9\}$, $\{c_{10}\}$ with probability 33.2%. In total more than 2500 different groupings are visited, and except for the posterior most probable grouping the posterior probabilities of all other groupings are less than 5%. The estimated posterior probability distribution for the number of groups becomes 43% for 3 groups, 48% for 4 groups, 8% for

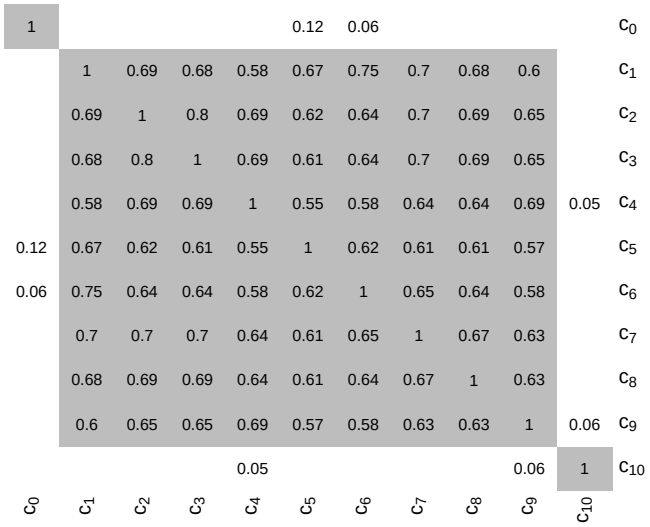
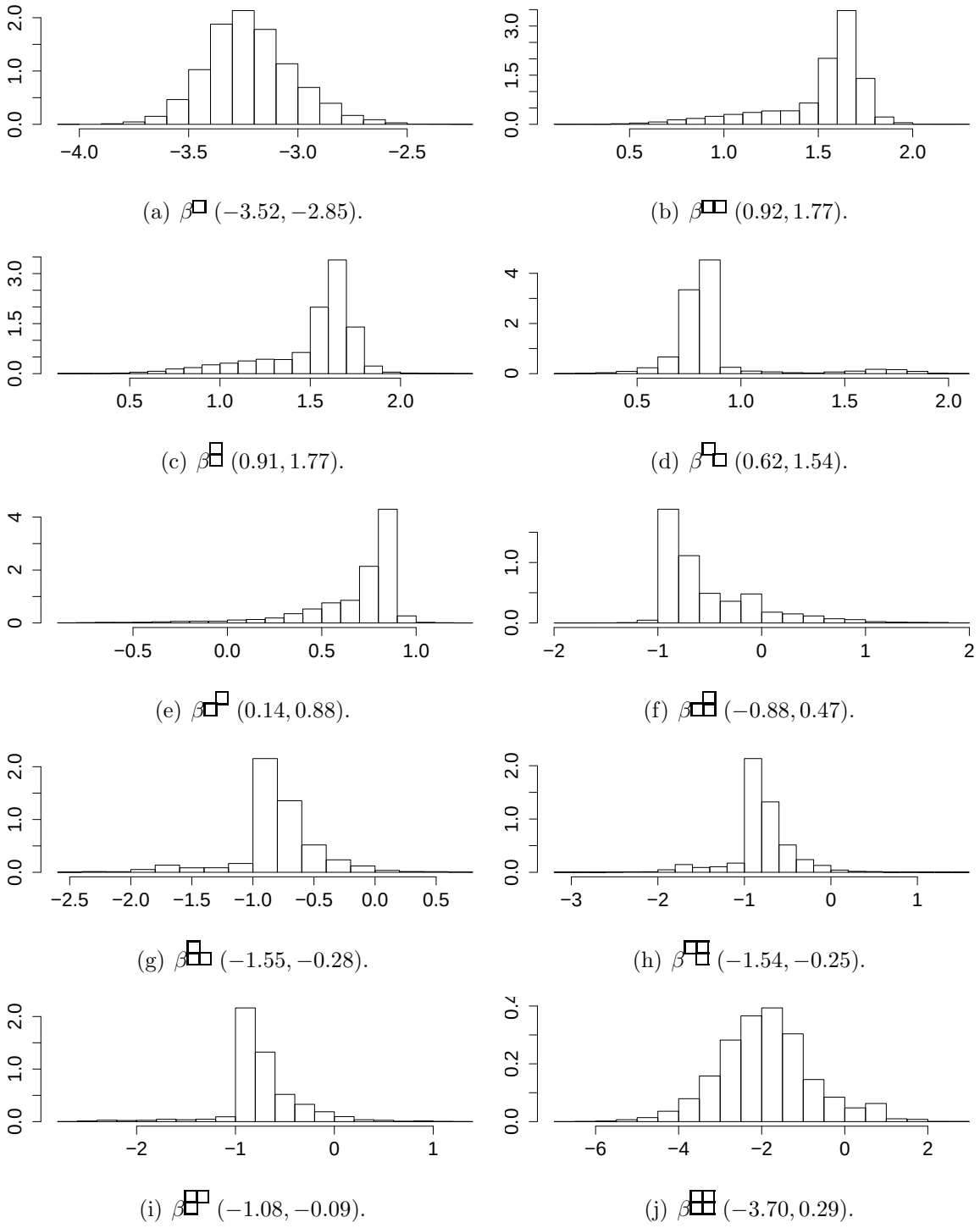


Figure 10: Red deer example: Estimated posterior probabilities for two configuration sets to be grouped together for the red deer data set. The estimated most probable grouping is shown in grey, and only probabilities larger than 5 % are given.

5 groups and 1% for 6 groups. In particular, the realizations with four or more groups are mostly groupings where the set $\{c_1, ..., c_9\}$ are split in various ways. This can also be seen in Figure 10, which shows the estimated posterior probability of two configuration sets being assigned to the same group. The grey blocks in this figure show the estimated posterior most probable grouping described above. Next we estimate the posterior density for the interaction parameters, see Figure 11. As we can see, most of the higher order interaction parameters becomes significantly different from zero, suggesting that a 2×2 clique system is needed for this data set. Figure 12 shows the estimated posterior density for the covariate parameters. As we can see from the credibility intervals, all these parameters are significantly different from zero, which justifies the need to include them.

Simulations of $p(x|z, \theta^C, y)$ for three randomly chosen posterior samples of z and θ^C are



(a) $\beta^{\square}$ $(-3.52, -2.85)$.

(b) $\beta^{\square\square}$ $(0.92, 1.77)$.

(c) $\beta^{\begin{smallmatrix}\square\\\square\end{smallmatrix}}$ $(0.91, 1.77)$.

(d) $\beta^{\begin{smallmatrix}\square&\\&\square\end{smallmatrix}}$ $(0.62, 1.54)$.

(e) $\beta^{\begin{smallmatrix}\square&\\\square&\end{smallmatrix}}$ $(0.14, 0.88)$.

(f) $\beta^{\begin{smallmatrix}\square&\square\\&\end{smallmatrix}}$ $(-0.88, 0.47)$.

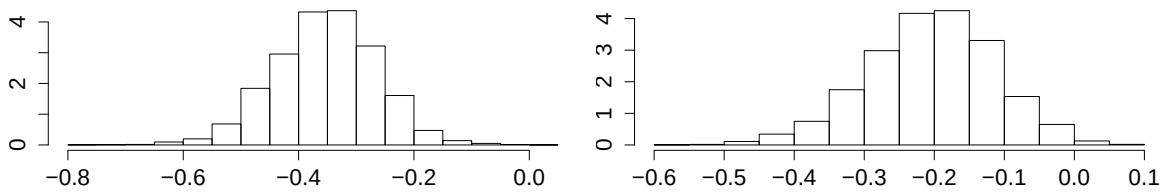
(g) $\beta^{\begin{smallmatrix}\square&\\\square&\square\end{smallmatrix}}$ $(-1.55, -0.28)$.

(h) $\beta^{\begin{smallmatrix}\square&\square\\\square&\end{smallmatrix}}$ $(-1.54, -0.25)$.

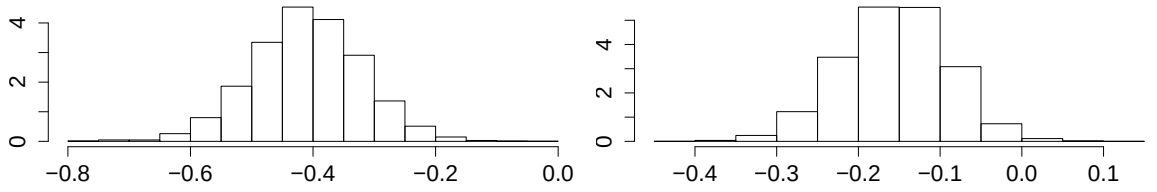
(i) $\beta^{\begin{smallmatrix}\square&\square\\\square&\end{smallmatrix}}$ $(-1.08, -0.09)$.

(j) $\beta^{\begin{smallmatrix}\square&\square\\\square&\square\end{smallmatrix}}$ $(-3.70, 0.29)$.

Figure 11: Red deer example: Estimated marginal posterior distribution for the interaction parameters. Estimated 95% credibility interval is given for each parameter.



(a) θ_1^C $(-0.50, -0.21)$. (b) θ_2^C $(-0.36, -0.07)$.



(c) θ_3^C $(-0.55, -0.27)$. (d) θ_4^C $(-0.26, -0.05)$.

Figure 12: Red deer example: Estimated marginal posterior distributions for the parameters of the covariates. Estimated 95 % credibility interval is given for each parameter.

Figure 13: Red deer example: Three realizations from the likelihood for three random samples of z from the posterior distribution.

shown in Figure 13. As we can see the spatial dependency in these realizations looks similar to the data which indicates that the features of this data set are captured with this model.

Using $\gamma = 0$ for this data set gives the estimated posterior probability distribution 24%, 63%, 11% and 2% for 3, 4, 5 and 6 groups respectively, whereas for $\gamma = 1$ we obtain 60%, 35% and 5% for 3, 4 and 5 groups respectively. Again we see that higher values of γ results in more realizations with fewer number of groups. For all the three values of γ the estimated most probable grouping is the same.

We end our disscussion of this data set by mentioning that some results of this data set when assuming a clique size of 3×3 is included in the supplementary material of this paper. These results indicate that no more significant structure is introduced in the 3×3 case for this data set.

6. CLOSING REMARKS

Our main fucus in this paper is to design a generic prior distribution for the parameters of an MRF. This is done by assuming a maximal $k \times l$ clique, but as the number of free parameters grows quickly as a function of k and l we construct our prior distribution such that it gives a positive probability for groups of parameters to have exactly the same value. In that way we reduce the effective number of parameters, still keeping the complexity a higher order neighbourhood provides. Proposal distributions that enables us to simulate from the resulting posterior distributed is also presented. However, to evaluate the likelihood we use a previously defined approximation to MRFs (Austad 2011), and the trade off between accuracy and computational complexity limits in practice the size of the cliques that can be assumed. An alternative to approximations is perfect sampling (Propp and Wilson 1996), but this was in all our examples too computational intensive. A third alternative would be to use an MCMC sample of x instead of a perfect sample, as described in for instance Everitt (2012). An issue with this approach is the need to set a burn in period for the sampler of x, where a too long burn in period would make the parameter sampler too intensive. Lastly, we illustrate the effect of our prior distribution and sampling algorithm on three examples.

Our focus in this paper is on binary MRFs. It is however possible to generalize our framework to dicrete MRFs, i.e. where $x_i \in \{0, 1, ..., K\}$ for $K \geq 2$. An indentifiable

parameterization of a dicrete MRF using clique potentials can with a small effort be defined in a similar way to what is done in the binary case, and ones this parameterization is established, the prior distribution presented in this paper can be used unchanged. The same apply to or sampling strategy.

With our prior distribution the size of the maximal cliques, and thereby the number of configuration sets, act as a hyper parameter and must be set prior to any sampling algorithm. One could imagine also putting prior distribution on these variables, introducing the need to construct algorithms for trans-dimensional sampling also for these quantities. Another way to avoid the need to set the number of configuration sets would be to construct a prior distribution for the β parameters. A natural choice would be to construct a positive prior probability for these parameters to be exactly zero, and in this way the significant interactions of a MRF can be inferred from data. However, it is not clear to us how to design prior distributions for the values of these interaction parameters, as higher order interactions intuitively would be different from lower order interaction. Also, grouping β parameters together in order to reduce the number of parameters would, for the same reason as above, make little sense. An ideal solution would be somehow to draw strength from both of the two parametrizations in order to assign a prior distribution to both the appearance of different cliques and the number of free parameters. This idea is currently work in progress.

SUPPLEMENTARY MATERIALS

Additional .pdf file: Proof of translational invariance for the interaction parameters, details for the MCMC sampling algorithm, trace plot for the independence model example, reed deer census count data with 3×3 clique, and parallelisation of the sampling

algorithm.

REFERENCES

Augustin, N. H., Mugglestone, M. A., and Buckland, S. T. (1996). "An Autologistic Model for the Spatial Distribution of Wildlife." *Journal of Applied Ecology*, 33, 339–347.

Austad, H. M. (2011). "Approximations of binary Markov random fields." Ph.D. thesis, Norwegian University of Science and Technology. Thesis number 292:2011. Available from http://urn.kb.se/resolve?urn=urn:nbn:no:ntnu:diva-14922.

Besag, J. (1986). "On the statistic analysis of dirty pictures." *Journal of the Royal Statistical Society. Series B (Methodological)*, 48, 259–302.

Buckland, S. T. and Elston, D. A. (1993). "Empirical Models for the Spatial Distribution of Wildlife." *Journal of Applied Ecology*, 30, 478–495.

Clifford, P. (1990). "Markov Random Fields in Statistics." In *Disorder in Physical Systems, A Volume in Honour of John M.Hammersley*, eds. G. Grimmett and D. J. Welsh. Oxford University Press.

Cressie, N. and Davidson, J. (1998). "Image analysis with partially ordered Markov models." *Computational Statistics and Data Analysis*, 29, 1–26.

Cressie, N. A. (1993). *Statistics for spatial data*. 2nd ed. New York: John Wiley.

Everitt, R. G. (2012). "Bayesian Parameter Estimation for Latent Markov Random Fields and Social Networks." *Journal of Computational and Graphical Statistics*, 21, 940–960.

Friel, N., Pettitt, A. N., Reeves, R., and Wit, E. (2009). "Bayesian inference in hidden Markov random fields for binary data defined on large lattices." 18, 243–261.

Grabisch, M., Marichal, L.-L., and Roubens, M. (2000). "Equivalent representation of set function." *Mathematics of operations reasearch*, 25, 157–178.

Green, P. J. (1995). "Reversible jump MCMC computation and Bayesian model determination." *Biometrika*, 82, 711–732.

Hammer, P. and Holzman, R. (1992). "Approximations of pseudo-boolean functions; application to game theory." *Methods and models of operations research*, 36, 3–21.

Heikkinen, J. and Högmander, H. (1994). "Fully Bayesian approach to image restoration with an application in biogeography." *Applied Statistics*, 43, 569–582.

Higdon, D. M., Bowsher, J. E., Johnsen, V. E., Turkington, T. G., Gilland, D. R., and Jaszczak, R. J. (1997). "Fully Bayesian estimation of Gibbs hyperparameters for emission computed tomography data." *IEEE Transactions on medical imaging*, 16, 516–526.

Hurn, M., Husby, O., and Rue, H. (2003). "A tutorial on image analysis." In *Spatial Statistics and Computational Methods*, ed. J. Møller, vol. 173 of *Lecture Notes in Statistics*, 87–141. Springer Verlag.

Martins, T. G., Simpson, D., Lindgren, F., and Rue, H. (2013). "Bayesian computing with INLA: new features." *Computational Statistics and Data Analysis*, 67, 68–83.

McGrory, C. A., Pettitt, A. N., Reeves, R., Griffin, M., and Dwyer, M. (2012). "Variational Bayes and the Reduced Dependence Approximation for the Autologistic Model on an

Irregular Grid With Applications." *Journal of Computational and Graphical Statistics*, 21, 781–796.

Møller, J., Pettitt, A., Reeves, R., and Berthelsen, K. (2006). "An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants." *Biometrika*, 93, 451–458.

Murray, I., Ghahramani, Z., and MacKay, D. (2006). "MCMC for doubly-intractable distributions." In *Proceedings of the Twenty-Second Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, 359–366. Arlington, Virginia: AUAI Press.

Propp, J. G. and Wilson, D. B. (1996). "Exact sampling with coupled Markov chains and applications to statistical mechanics." *Random Structures & Algorithms*, 9, 223–252.

Rue, H., Martino, S., and Chopin, N. (2009). "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations." *Journal of the Royal Statistical Society, Series B*, 71, 319–392.

Tjelmeland, H. and Austad, H. M. (2012). "Exact and approximate recursive calculations for binary Markov random fields defined on graphs." *Journal of Computational and Graphical Statistics*, 21, 758–780.

,

Supplemental materials to the paper

Fully Bayesian binary Markov random field models: Prior specification and posterior simulation

Petter Arnesen and Håkon Tjelmeland

PROOF OF TRANSLATIONAL INVARIANCE FOR THE INTERACTION PARAMETERS

As explained in the paper an MRF with torus boundary condition is given by

$$p(x) = c \exp\left(\sum_{\Lambda \in \mathcal{L}_m} U_\Lambda(x_\Lambda, \theta)\right), \tag{7}$$

where $U_\Lambda(x_\Lambda, \theta)$ is a potential function for a given maximal clique Λ, $\mathcal{L}_m$ is the set of maximal cliques, and θ is a parameter vector. If we assume the MRF to be stationary the potential function $U_\Lambda(\cdot, \cdot)$ must be translational invariant in that the function must be equal for all $\Lambda \in \mathcal{L}_m$. We can thereby simplify the notation by replacing $U_\Lambda(x_\Lambda, \theta)$ with $U(x_\Lambda, \theta)$. Alternatively $p(x)$ can be expressed by

$$p(x) = c \exp\left(\sum_{\Lambda \in \mathcal{L}} \beta^\Lambda \prod_{i \in \Lambda} x_i\right), \tag{8}$$

where β^Λ is referred to as the interaction parameter for clique Λ, which is said to be of $|\Lambda|$'th order, and where $\mathcal{L}$ is the set of all cliques. In this section we prove by induction that also β^Λ is translational invariant under the given assumptions. First we assume all interactions up to order $|\Lambda| = o$ to be translational invariant and then prove that then an interaction parameter of order $o+1$ must also be translational invariant. Since $\beta^\emptyset$ obviously is translational invariant this is enough to complete the proof. Thus, now assume all interaction parameters up to order o to be translational invariant. Assume x to be such that only one interaction parameter of order $o+1$ appear in the sum in (8), and let this interaction parameter be $\beta^\Lambda = \beta^{\{i_1,...,i_{o+1}\}}$, where $i_1, ..., i_{o+1}$ are the positions of the nodes in this interaction. One example of such an x is $x_i = I(i \in \Lambda)$ for $i \in S$. Next, we let $x' = (x'_1, ..., x'_{nm})$ be a translation of x such that $x'_i = x_{t_{l_1,l_2}(i)}$ for $i \in S$, where $t_{l_1,l_2}(i)$ is a translation that takes the position of node i and moves it l_1 positions upwards and l_2 positions leftwards in the lattice, correcting for the torus boundary condition, i.e.

$$t_{l_1,l_2}(i) = 1 + \left[\left(\left\lfloor \frac{i-1}{m} \right\rfloor + l_1\right) \mod n\right] m + \left(\left(\frac{i-1}{m} - \left\lfloor \frac{i-1}{m} \right\rfloor\right) m + l_2\right) \mod m.$$

The assumed stationarity clearly gives that we must have $p(x) = p(x')$, i.e.

$$\beta^{\{i_1,...,i_{o+1}\}} + \sum_{\Lambda \in S:|\Lambda|\leq o} \beta^\Lambda \prod_{i\in\Lambda} x_i = \beta^{\{t_{l_1,l_2}(i_1),...,t_{l_1,l_2}(i_{o+1})\}} + \sum_{\Lambda \in S:|\Lambda|\leq o} \beta^\Lambda \prod_{i\in\Lambda} x'_i.$$

Using the assumption that all interaction parameters up to order o are translational invariant we get

$$\beta^{\{i_1,...,i_{o+1}\}} = \beta^{\{t_{l_1,l_2}(i_1),...,t_{l_1,l_2}(i_{o+1})\}}.$$

Thus, by induction all interaction parameters are translational invariant.

DETAILS FOR THE MCMC SAMLING ALGORITHM

In this Section we provide details of the proposal distributions that we use when sampling from the posterior distribution

$$p(z|x) \propto p(x|B\Phi(z))p(z),$$

where $p(x|B\Phi(z))$ and $p(z)$ are the MRF and the prior given in the paper, respectively.

To simulate from this posterior distribution we adopt a reversible jump Markov chain Monte Carlo (RJMCMC) algorithm with three types of updates. The first update type uses a random walk proposal for one of the φ parameters, the second proposes to move one configuration set to a new group, and the third proposes to change the number of groups, r, in the partition of the configuration sets. In the following we describe the proposal mechanisms for each of the three update types. The corresponding acceptance probabilities are given by standard formulas. It should be noted that only the last type of proposal produces a change in the dimension of the parameter space.

Random walk proposal for parameter values

The first proposal in our algorithm is simply to propose a new value for an already existing parameter using a random walk proposal, but correcting for the fact the parameters should sum to zero. More precisely, we first draw a change $\varepsilon \sim \mathrm{N}(0, \sigma^2)$, where σ^2 is an algorithmic tuning parameter. Second, we uniformly draw one element from the current state $z = \{(C_i, \varphi_i), i = 1, \ldots, r\}$, (C_i, φ_i) say, and define the potential new state as

$$z^* = \left\{ \left(C_j, \varphi_j - \frac{1}{r}\varepsilon \right), j = 1, \ldots, i-1, i+1, \ldots, r \right\} \cup \left\{ \left(C_i, \varphi_i + \varepsilon - \frac{1}{r}\varepsilon \right) \right\}.$$

Proposing to change the group for one configuration set

Letting the current state be $z = \{(C_i, \varphi_i), i = 1, \ldots, r\}$, we start this proposal by drawing a pair of groups, C_i and C_j say, where the first set C_i is restricted to include at least two configuration sets. We draw C_i and C_j so that the difference between the corresponding parameter values, $\varphi_i - \varphi_j$, tend to be small. More precisely, we draw (i, j) from the joint distribution

$$q(i,j) \propto \begin{cases} \exp\left(-(\varphi_i - \varphi_j)^2\right) & \text{if } i \neq j \text{ and group } C_i \text{ contains at least two configuration sets,} \\ 0 & \text{otherwise.} \end{cases}$$

Thereafter we draw uniformly at random one of the configuration sets in C_i, c say. Our potential new state is then obtained by moving c from C_i to C_j. Thus, our potential new state becomes

$$z^* = \{z \setminus \{(C_i, \varphi_i), (C_j, \varphi_j)\}\} \cup \{(C_i \setminus c, \varphi_i), (C_j \cup c, \varphi_j)\}.$$

Trans-dimensional proposals

Let again the current state be $z = \{(C_i, \varphi_i), i = 1, \ldots, r\}$. In the following we describe how we propose a new state by either increasing or reducing the number of groups, r, with one. There will be a one-to-one transition in the proposal, meaning that the opposite proposal, going from the new state to the old state has a non-zero probability. We make no attempt to jump between states where the difference between the dimensions are larger than one.

First we draw whether to increase or to decrease the number of groups. If the number of groups are equal to the number of configurations sets, no proposal to increase the number of groups can be made due to the fact that empty groups have zero prior probability. In that

case we propose to decrease the number of dimensions with probability 1. In our proposals we also make the restriction that only groupings containing at least one group with only one configuration set can be subject to a dimension reducing proposal. In a case where no such group exists, a proposal of increasing the number of dimensions are made with probability 1. In a case where both proposals are allowed we draw at random which to do with probability 1/2 for each. Note that at least one of the two proposals is always valid.

We now explain how to propose to increase the number of groups by one. We start by drawing uniformly at random one of the groups with more than one configuration set, C_i say, which we want to split into two new groups. Thereafter we draw uniformly at random one of the configuration sets in C_i, c say, and form a new partition of the configuration sets by extracting c from C_i and adding a new group containing only c. Next we need to draw a parameter value for the new group $\{c\}$, and the parameter values for the other groups also need to be modified for the proposal to conform with the requirement that the sum of the (proposed) parameters should equal zero. We do this by first drawing a change $\varepsilon \sim \mathrm{N}(0, \sigma^2)$ in the parameter value for c, where σ^2 is the same tuning parameter as in the random walk proposal. We then define the potential new state as

$$\begin{aligned} z^* &= \left\{ \left(C_j, \varphi_j - \frac{1}{r+1}(\varphi_i + \varepsilon) \right), j = 1, \ldots, i-1, i+1, \ldots, r \right\} \cup \\ &\quad \left\{ \left(C_i \setminus c, \varphi_i - \frac{1}{r+1}(\varphi_i + \varepsilon) \right), \left(\{c\}, \varphi_i + \varepsilon - \frac{1}{r+1}(\varphi_i + \varepsilon) \right) \right\}. \end{aligned}$$

Next we explain the proposal we make when the dimension is to be decreased by one. Since we need a one-to-one transition in our proposals, we get certain restrictions for these proposals. In particular, the fact that only groupings containing at least one group with only one configuration set are possible outcomes from a dimension increasing proposal dictates that dimension decreasing proposals only can be made from such groupings. Assume again

our current model to be $z = \{(C_i, \varphi_i), i = 1, \ldots, r\}$, where at least one group contains only one configuration set. The strategy is to propose to merge one group consisting of only one configuration set into another group. As in Section 6, we draw the two configuration sets to be merged so that the difference between the corresponding parameter values tend to be small. More precisely, we let the two groups be C_i and C_j where (i, j) is sampled according to the joint distribution

$$q(i,j) \propto \begin{cases} \exp\left(-(\varphi_i - \varphi_j)^2\right) & \text{if } i \neq j \text{ and } C_i \text{ consists of only one configuration set,} \\ 0 & \text{otherwise.} \end{cases}$$

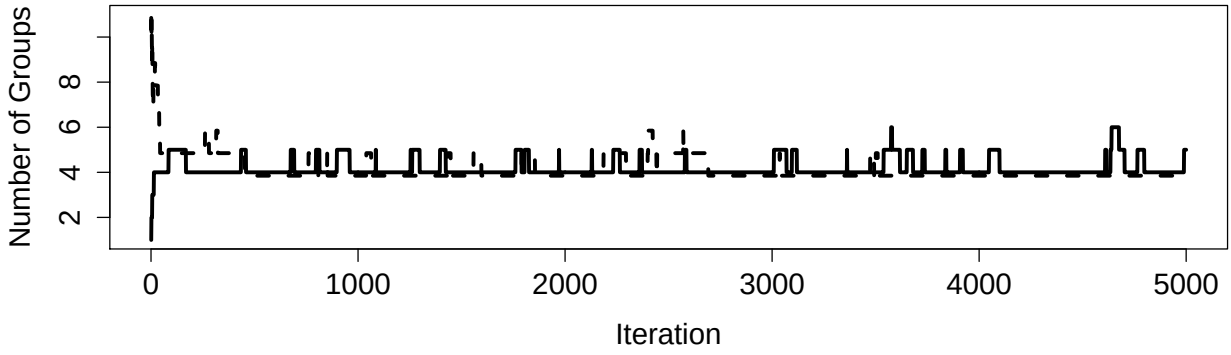
Next we need to specify potential new parameter values. As these must conform with how we generated potential new values in the split proposal, we have no freedom left in how to do this. The potential new state must be

$$z^* = \left\{ \left(C_k, \varphi_k + \frac{1}{r-1}\varphi_i \right), k \in \{1, \ldots, r\} \setminus \{i, j\} \right\} \cup \left\{ \left(C_j \cup C_i, \varphi_j + \frac{1}{r-1}\varphi_i \right) \right\}.$$

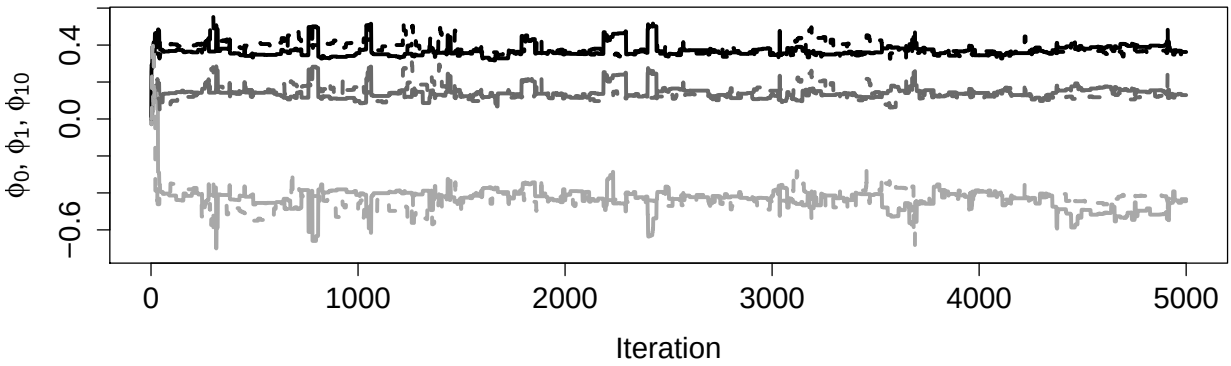
The split and merge steps produce a change in the dimension of the parameter space, so to calculate the acceptance probabilities for such proposals we need corresponding Jacobi determinants. It is straightforward to show that the Jacobi determinants for the merge and split proposals become $\frac{r}{r-1}$ and $\frac{r}{r+1}$, respectively.

TRACE PLOT FOR THE INDEPENDENCE MODEL EXAMPLE

To check for convergence of our sampling algorithm we investigate different trace plots. One example for the independence model is shown in Figure 14. As we can see from this figure the algorithm converges quickly.



(a) Trace plot for the number of groups.



(b) Trace plot for ϕ_0(black), ϕ_1(dark grey) and ϕ_{10}(light grey).

Figure 14: Independence model example: Trace plots for the first quarter of the posterior simulation run. Solid curves are the result from a simulation where the initial number of groups is 1, and dashed curves are from a run with an initial value of 11 (maximal) number of groups.

RED DEER CENSUS COUNT DATA WITH 3×3 CLIQUE

In this section we present some results when assuming a clique size of 3×3 for the red deer data set presented in Section 5.3 in the paper. The main drawback with our approach is computational time, which is very dependent on the approximation parameter ν. One also needs to keep in mind that even data from simple models will need many groups in the 3×3 case to be modeled correctly. For instance, for the independence model the 401 configuration

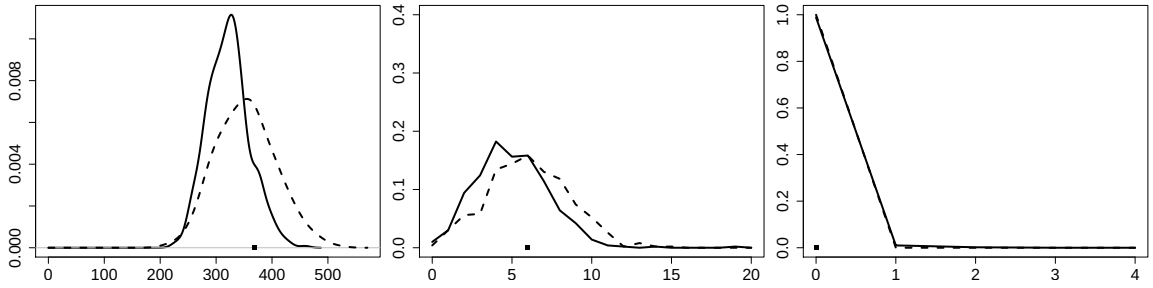
sets would need to be separated into 10 groups, while for the Ising model one would need 11 groups to get the correct model grouping. Similarly, the posterior most probable grouping found for the 2×2 case for the reed deer example would need 38 groups to be modeled in the 3×3 case. Thus it is important not to assume bigger cliques than needed. However for this data set it is possible to run the sampling algorithm with 3×3 clique, even though this is computationally expensive.

In these simulations we use $\nu = 7$, since this is the largest value of ν that gives a reasonable computational time. Also, to get convergence we need a small generalization to the proposal distribution for the trans-dimensional sampling step presented in Section 6. In particular we allow for several configuration sets to be split out into a new group at a single proposal, and correspondingly allow for the possibility of several configuration sets to be merge into another group in one single proposal. The estimated marginal distribution of the number of groups is 1%, 65%, 33%, and 1% for 29, 30, 31 and 32 groups respectively. Three realizations from the likelihood for three randomly chosen realization of z is shown in Figure 15(a), and comparing with the realizations for the 2×2 case, see Figure 13 in the paper, it is hard to see any differences in the spatial structure of the realisations. We also investigated the distribution of four statistics for 5000 realization from the likelihood of each of the two clique sizes, see Figure 15(b), and it appears to be little difference also here. These results indicate that 2×2 cliques might have sufficient complexity to explain this data set.

PARALLELISATION OF THE SAMPLING ALGORITHM

Most of the computing time for running our sampling algorithm is used to evaluate the likelihood in (7). In order to reduce the running time we adopt a scheme that do multiple

(a) Three realizations from the likelihood for three random samples of z from the posterior distribution.



(b) Distribution of three functions of realizations from the likelihood with 3×3 cliques (solid) and 2×2 cliques (dashed). The three functions are $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \begin{pmatrix} 000 \\ 000 \\ 000 \end{pmatrix}\right)$ (left), $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \begin{pmatrix} 000 \\ 000 \\ 011 \end{pmatrix}\right)$ (middle), and $g(x) = \sum_{\Lambda \in \mathcal{L}_m} I\left(x_\Lambda = \begin{pmatrix} 111 \\ 110 \\ 100 \end{pmatrix}\right)$ (right).

Figure 15: Red deer 3×3 example: Posterior results with $\gamma = 0.5$.

updates of the Markov chain by evaluating likelihoods in parallel.

Assume we are in a state z and propose to split/merge into a new state z_1. Now there are two possible outcomes for this proposal. Either we reject the proposal, which result in state z, or we accept the proposal, which result in state z_1. Either way we always propose a parameter update in the next step, and proposing this step from both the two states z and z_1 before evaluating the acceptance probability for the split/merge step is possible. The possible outcomes for these three proposals are z, z_1, z_2 and z_{12}, where z is the outcome where neither the split/merge proposal nor the following parameter proposal is accepted, z_1 is the outcome where the split/merge proposal is accepted but not the following parameter

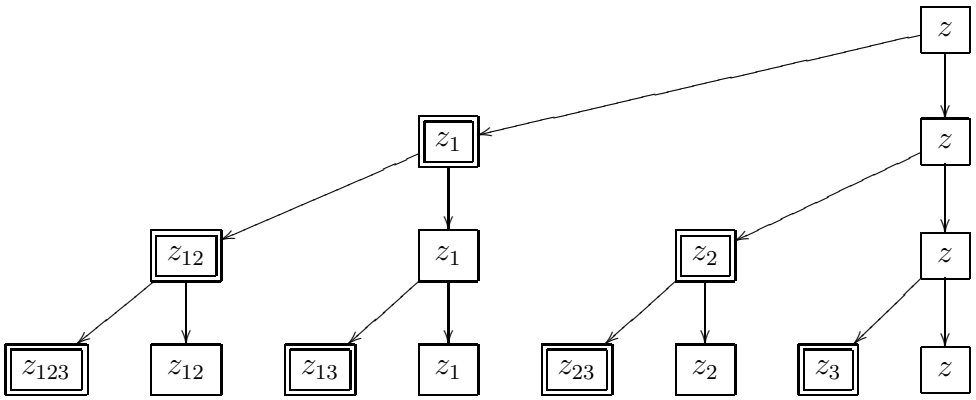


Figure 16: Proposal scheme for parallel likelihood evaluations. Starting in model z, proposals are made down the graph. Arrows pointing straight down represents rejection of proposal while arrow pointing down and left represent acceptance. Double squares are used to represents states where a new likelihood evaluation is needed.

proposal, z_2 is the outcome where the split/merge proposal is not accepted but the parameter proposal is, and z_{12} is the outcome where both the split/merge proposal and the following parameter proposal are accepted. If we continue the argument we can do the same to propose updates where configurations are moved from one group to another group, and in the red deer example we even include a level where updates of covariates are proposed. After making all proposals we evaluate the likelihood for each possible state in parallel. The result is that we do need to evaluate too many likelihoods, but if the number of CPUs that are available is larger than or equal to the number of likelihoods we need to evaluate, a computational gain close to the number of levels is obtained. The updating scheme is illustrated in Figure 16.