Approximations and bounds for binary Markov random fields

Haakon M. Austad and Håkon Tjelmeland

Abstract The formulation of discrete Markov random fields (MRFs) include a computationally intractable normalising constant, which limits the applicability of the model class. The normalising constant can in principle be computed by marginalising out each variable in turn, but in practice this is computationally feasible for small lattices only. We propose an approximate marginalisation operation, which can be used to obtain an approximation of the normalisation constant and an approximate probability distribution with an easy to compute normalising constant. In turn these can be used to find an approximation of the maximum likelihood estimators, or can be used in stead of the corresponding exact quantity in a fully Bayesian setting. We also discuss how the approximate marginalisation operation can be modified to give upper and lower bounds for the normalising constant. The same approximation strategy can be used to define an approximate maximisation operation, which in turn can be used to find an approximation of, or lower and upper bounds for, the maximum value of the MRF probability.

1 Introduction

In statistics in general and especially in spatial statistics we often find ourselves with distributions known only up to an unknown normalising constant. Calculating the constant typically involves high dimensional summation or integration. This is the case for the class of discrete Markov random fields (MRF).

Haakon M. Austad
Kongsberg Defence & Aerospace AS, Instituttveien 10, P.O.Box 26, NO-2027 Kjeller, Norway, e-mail: hmaustad@gmail.com

Håkon Tjelmeland
Department of Mathematical Sciences, Norwegian University of Science and Technology, Gløshaugen, NO-7491 Trondheim, Norway, e-mail: haakont@stat.ntnu.no

Ninth International Geostatistics Congress, Oslo, Norway, June 11. – 15., 2012.

A common situation is spatial statistics is that we have some unobserved latent field x for which we have some associated observations y. We model x as an MRF with unknown parameters θ. If we are Bayesians we adopt a prior for θ and study the posterior distribution $p(\theta|y)$. A frequentist approach could involve finding a maximum likelihood estimator for θ from observed data y or from a training image. Independently or in combination of these investigations we might want to perform simulations and generate samples from $p(x|\theta)$ for some values of θ. Without the normalising constant however, all these become non-trivial tasks.

Several techniques have been proposed to overcome this problem. The normalising constant can be estimated by Markov chain Monte Carlo (MCMC) to produce maximum likelihood estimates (Geyer and Thompson, 1995; Gelman and Meng, 1998; Gu and Zhu, 2001). Møller et al. (2006) propose a possible strategy when exact sampling of the latent field is feasible. In the present paper however, we focus on deterministic methods, where by deterministic we mean that repeating the estimation process yields the same estimate. In Reeves and Pettitt (2004) the authors discuss the variable elimination algorithm for a class of models including discrete MRFs. For MRFs defined on a lattice this allows for calculation of the normalisation constant on lattices with up to around 20 rows for models with first order neighbourhoods. In Friel and Rue (2007) and Friel et al. (2009) the authors construct approximations for larger lattices by doing computations for a number of small sub-lattices using the algorithm in Reeves and Pettitt (2004). The variable elimination algorithm is also closely related to the junction tree algorithm (Cowell et al., 2007) and several authors have proposed variants of this algorithm that generates approximations of, or upper or lower bounds for, for the normalisation constant, see for example Jordan et al. (1999) and Mateescu et al. (2010) and references therein.

In the present paper we consider binary MRFs and propose a new scheme for including approximations into the variable elimination algorithm. From this we obtain a computationally feasible approximation of, or upper or lower bound for, the normalising constant. The energy function of a binary MRF is a polynomial of binary variables, also called a pseudo-Boolean function. To develop our approximations we use well-known approximate representations of pseudo-Boolean function, see Hammer and Holzman (1992) and Grabisch et al. (2000). In the following sections we discuss the most important aspects of our approximation strategy, and refer to Austad and Tjelmeland (2011) or Austad (2011) for a more detailed derivation.

2 Binary MRFs and the variable elimination algorithm

Assume we have a vector of n binary variables, $x = (x_1, \ldots, x_n) \in \Omega = \{0,1\}^n$. Each of the n variables we associate with a node in a graph. We number the nodes from 1 to n and let $N = \{1, \ldots, n\}$ denote the set of all nodes. We let $\mathcal{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_n\}$ denote a neighbourhood system, where $\mathcal{N}_i \subseteq N \setminus \{i\}$ denotes the set of neighbours of node i. As usual we require a symmetrical neighbourhood system, so $i \in \mathcal{N}_j \Leftrightarrow j \in \mathcal{N}_i$, and write $i \sim j$ whenever $i \in \mathcal{N}_j$. For example we may have a 2D rectangular

lattice where the nodes are numbered from 1 to n in the lexicographical order, where $\mathcal{N}_i$ for an interior node i contains the four or eight nodes closest to it and nodes on the lattice border have correspondingly fewer neighbours.

We use the following standard notations, $x_A = (x_i, i \in A)$ and $x_{-A} = x_{N\setminus A}$ for $A \subseteq N$, and $x_{-i} = x_{N\setminus\{i\}}$ for $i \in N$. A distribution for x, $p(x)$, is said to be a binary MRF with respect to the neighbourhood system $\mathcal{N}$ if $p(x) > 0$ for all $x \in \Omega$ and the full conditionals have the following Markov property,

$$p(x_i|x_{-i}) = p(x_i|x_{\mathcal{N}_i}) \text{ for all } x \in \Omega. \tag{1}$$

We say a set of nodes $\Lambda \subseteq N$ is a clique if $i \in \mathcal{N}_j$ for all distinct pairs of $i, j \in \Lambda$, and Λ is a maximal clique if it is not a subset of another clique. We let $\mathscr{C}$ denote the set of all cliques. The Hammersley-Clifford theorem (Besag, 1974; Clifford, 1990) then states that a distribution $p(x)$ is an MRF if and only if it can be expressed as

$$p(x) = \frac{1}{c}\exp\{U(x)\} \text{ where } U(x) = \sum_{\Lambda\in\mathscr{C}} V_\Lambda(x_\Lambda), \tag{2}$$

for some potential functions $V_C(x_C), C \in \mathscr{C}$. Here c is a normalising constant and $U(x)$ is usually called the energy function associated to $p(x)$.

2.1 Representation of the energy function

A real valued function of binary variables, for example the potential and energy function associated to a binary MRF, is called a pseudo-Boolean function. Hammer and Rudeanu (1968) showed that any pseudo-Boolean function can be expressed uniquely as a binary polynomial, so we have

$$U(x) = \sum_{\Lambda\subseteq N} \beta^\Lambda \prod_{i\in\Lambda} x_i, \tag{3}$$

where β^Λ are real coefficients which we refer to as interactions. For the energy function $U(x)$ it can be shown, see Tjelmeland and Austad (2012), that $\beta^\Lambda = 0$ whenever Λ is not a clique, so most of the 2^n interactions in (3) are equal to zero and a reduced representation of $U(x)$ is possible,

$$U(x) = \sum_{\Lambda\in S} \beta^\Lambda \prod_{i\in\Lambda} x_i, \tag{4}$$

where S is a set of subsets of N at least containing all $\Lambda \subseteq N$ for which $\beta^\Lambda \neq 0$. We say that our representation is dense if for all $\Lambda \in S$ also all subsets of Λ are included in S. The minimal dense representation of $U(x)$ is thereby (4) with

$$S = \{\lambda \subseteq N : \beta^\Lambda \neq 0 \text{ for some } \Lambda \supseteq \lambda\}. \tag{5}$$

In the following we only consider dense representations of $U(x)$.

2.2 The variable elimination algorithm

The variable elimination algorithm calculates the normalising constant c in (2) by summing out from $p(x)$ each variable $x_1,\ldots,x_n$ in turn. As discussed in Reeves and Pettitt (2004) and Friel and Rue (2007) we can perform this summation procedure more efficiently by factorising the unnormalised distribution.

Assume we have a dense representation of $U(x)$ as discussed above and assume we want to sum out x_i. Clearly we can then always split the set S into $S_{\{i\}} = \{\Lambda \in S : i \in \Lambda\}$ and $S_{-\{i\}} = \{\Lambda \in S : i \notin \Lambda\}$. Correspondingly we can split the sum in (4) in a sum of two sums,

$$U(x) = \sum_{\Lambda \in S_{-\{i\}}} \beta^{\Lambda} \prod_{i \in \Lambda} x_i + \sum_{\Lambda \in S_{\{i\}}} \beta^{\Lambda} \prod_{i \in \Lambda} x_i. \tag{6}$$

Note that the first of these two sums is not a function of x_i, so for $p(x_{-i}) = \sum_{x_i} p(x)$ we get

$$p(x_{-i}) = \frac{1}{c} \exp\left\{ \sum_{\Lambda \in S_{-\{i\}}} \beta^{\Lambda} \prod_{i \in \Lambda} x_i \right\} \sum_{x_i \in \{0,1\}} \exp\left\{ \sum_{\Lambda \in S_{\{i\}}} \beta^{\Lambda} \prod_{i \in \Lambda} x_i \right\}. \tag{7}$$

The sum over x_i can be expressed as a binary polynomial, i.e.

$$\sum_{x_i \in \{0,1\}} \exp\left\{ \sum_{\Lambda \in S_{\{i\}}} \beta^{\Lambda} \prod_{i \in \Lambda} x_i \right\} = \exp\left\{ \sum_{\Lambda \subseteq \mathcal{N}_i} \check{\beta}^{\Lambda} \prod_{i \in \Lambda} x_i \right\}, \tag{8}$$

where the interactions $\check{\beta}^{\Lambda}$ can be sequentially calculated as detailed in Tjelmeland and Austad (2012). Thus, $\beta^{\Lambda}, \Lambda \in S_{-\{i\}}$ and $\check{\beta}^{\Lambda}, \Lambda \subseteq \mathcal{N}_i$ together give a representation of the energy function $U(x_{-i})$ associated to $p(x_{-i})$ in a form corresponding to (4). Note that $p(x_{-i})$ is also a binary MRF, with a new neighbourhood system and set of cliques. The above summation is thereby the first step in an sequential procedure for calculating the normalising constant c. In each step we sum over one of the remaining variables by splitting the energy function as above. Repeating the procedure until we have summed out all variables naturally yields the normalising constant.

The computational bottleneck for the above algorithm is when computing the sum in (8). The number of new interactions that have to be calculated is two to the power of the number of neighbours of the node to be summed out. For the procedure to be computationally feasible for a binary MRF on a rectangular lattice with a first order neighbourhood this in practice restricts the number of rows of the lattice to be < 20. In the next section we define an approximation operation of pseudo-Boolean

functions that can be used to define an approximate variable elimination algorithm that is computationally feasible also for larger lattices.

3 The approximate variable elimination algorithm

Consider a pseudo-Boolean function, $U(x), x = (x_1, \ldots, x_n) \in \Omega$ say, with a dense representation (4). Assume we are in the variable elimination algorithm and next should sum out x_i, but that the number of neighbours of node i is so high that this summation operation is computationally infeasible. To cope with this problem we propose before doing the summation to approximate $U(x)$ with a pseudo-Boolean function $\widetilde{U}(x)$ where the number of neighbours of node i is reduced to a feasible number. More precisely, we sequentially reduce the number of neighbours of node i with one at a time until we have reached a predefined maximum number of neighbours, ν. Assuming we define the approximate energy function by minimising the error sum of squares, Austad and Tjelmeland (2011) show that an upper bound on the approximation error by redefining two neighbour nodes i and j not to be neighbours any more, is

$$\frac{1}{4} \sum_{\Lambda \in S_{\{i,j\}}} \left| \beta^{\Lambda} \right|, \tag{9}$$

where $S_{\{i,j\}} = \{\Lambda \in S : i, j \in S\}$. Thus, we first find the value of j that minimise (9), redefine the neighbourhood system so that i and j no longer are neighbours, and approximate $\widetilde{U}(x)$ by minimising the error sum of squares. Denoting all resulting approximated values by a tilde, we have $\widetilde{S} = S \setminus S_{\{i,j\}}$, for $\Lambda \in S_{\{i,j\}}$ and Austad and Tjelmeland (2011) show that we get

$$\widetilde{\beta}^{\Lambda \setminus \{i,j\}} = \beta^{\Lambda \setminus \{i,j\}} - \frac{1}{4} \beta^{\Lambda}, \tag{10}$$

$$\widetilde{\beta}^{\Lambda \setminus \{i\}} = \beta^{\Lambda \setminus \{i\}} + \frac{1}{2} \beta^{\Lambda}, \tag{11}$$

$$\widetilde{\beta}^{\Lambda \setminus \{j\}} = \beta^{\Lambda \setminus \{j\}} + \frac{1}{2} \beta^{\Lambda}. \tag{12}$$

For sets $\Lambda \in \widetilde{S}$ where the the interaction β^{Λ} is not defined by (10), (11) or (12), we have $\widetilde{\beta}^{\Lambda} = \beta^{\Lambda}$, and clearly $\widetilde{\beta}^{\Lambda} = 0$ for all $\lambda \notin \widetilde{S}$. The approximate variable elimination algorithm is summarised in Figure 1.

On should note that as a side effect of the approximate variable elimination algorithm we get an approximation $\widetilde{p}(x)$ of $p(x)$ given as

$$\widetilde{p}(x) = p(x_1|x_2, \ldots, x_n) \widetilde{p}(x_2|x_3, \ldots, x_n) \cdots \widetilde{p}(x_{n-1}|x_n) \widetilde{p}(x_n), \tag{13}$$

where $\widetilde{p}(x_i|x_{i+1}, \ldots, x_n)$ is the conditional distribution corresponding to the approximate distribution we have for $x_i, \ldots, x_n$ after we have (approximately) summed out $x_1, \ldots, x_{i-1}$.

1. For $i = 1, \ldots, n$:
   a. While $|\mathcal{N}_i| > \nu$
      i. Find $j = \arg\min_j \sum_{\Lambda \in S_{\{i,j\}}} |\beta^\Lambda|$.
      ii. Approximate the energy function by
         A. for all $\Lambda \in S_{\{i,j\}}$ set $\beta^{\Lambda\setminus\{i,j\}} := \beta^{\Lambda\setminus\{i,j\}} - \frac{1}{4}\beta^\Lambda$, $\beta^{\Lambda\setminus\{i\}} := \beta^{\Lambda\setminus\{i\}} + \frac{1}{2}\beta^\Lambda$ and $\beta^{\Lambda\setminus\{j\}} := \beta^{\Lambda\setminus\{j\}} + \frac{1}{2}\beta^\Lambda$.
         B. Set $S := S \setminus S_{\{i,j\}}$, i.e. set $\beta^\Lambda := 0$ for all Λ not in the new set S.
   b. Sum out variable x_i as defined in Section 2.2.
2. Calculate the approximate normalising constant $\widetilde{c} = \exp\{\beta^\emptyset\}$.

Fig. 1 The approximate variable elimination algorithm for finding an approximation, $\widetilde{c}$, to the normalising constant c.

4 Bounds for the MRF normalising constant

When approximating an energy function $U(x)$ with $\widetilde{U}(x)$ by minimising the error sum of squares as discussed in Section 3 it is also possible to find the associated error, i.e. for any state x analytical expressions for $\widetilde{U}(x) - U(x)$ are available. In particular these analytical expressions can be used to define lower and upper bounds for $U(x)$ represented on the same dense set S as $\widetilde{U}(x)$, for details again see Austad and Tjelmeland (2011). Thereby a lower (or upper) bound on the normalising constant can be found by replacing item 1.a.ii.A in Algorithm 1 by expressions for the lower (or upper) bound for $U(x)$. On should note that having a lower or upper bound for c we immediately also have a corresponding lower or upper bound for the probability distribution $p(x)$.

5 Approximations and bounds for maximum of a binary MRF

The Viterbi algorithm, which finds $\max_x p(x)$, is similar to the variable elimination algorithm except that in stead of summing out x_i it takes the maximum over x_i. Thus, in stead of (7) one gets

$$\max_x p(x) = \frac{1}{c} \max_{x_{-i}} \left[\exp\left\{ \sum_{\Lambda \in S_{-\{i\}}} \beta^\Lambda \prod_{i \in \Lambda} x_i \right\} \max_{x_i} \exp\left\{ \sum_{\Lambda \in S_{\{i\}}} \beta^\Lambda \prod_{i \in \Lambda} x_i \right\} \right]. \quad (14)$$

The inner maximisation in this expression becomes computationally too expensive if the number of neighbours of i is too high, so just like the variable elimination algorithm the Viterbi algorithm is computationally feasible for MRFs defined on reasonably small lattices only. However, an approximate Viterbi algorithm can be defined correspondingly to how we introduced an approximation to the variable elimination algorithm in Section 3. Alternatively lower or upper bounds for $\max_x p(x)$ can

be found by replacing the approximation $U(x)$ with a corresponding lower or upper bound, corresponding to what we did for the variable elimination algorithm in Section 4.

6 Examples

In this section we present approximation results and lower and upper bounds for an Ising model. More examples can be found in Austad and Tjelmeland (2011), including an example demonstrating how the approximation can be used to fit a fully Bayesian model to a given data set.

Assume we have an Ising model on a 100×100 lattice with model parameter θ. Thus we have $n = 10\,000$ nodes and two nodes i and j are neighbours if and only if they are located next to eachother in the horizontal or vertical direction. The energy function, which is now of course also a function of θ, is given as

$$U(x) = \theta \sum_{i \sim j} I(x_i = x_j), \tag{15}$$

where $I(\cdot)$ is the indicator function. In the approximate algorithms we sum or maximise out the variables in the lexicographical order.

To evaluate the performance of the approximation algorithms we first sample a perfect sample from the Ising model using coupling from the past (Propp and Wilson, 1996) for a given parameter value θ_{true}. Then, treating our realisation as data and θ as an unknown parameter we approximate the posterior distribution for θ, $p(\theta|x) \propto p(\theta)p(x|\theta)$, by replacing the likelihood $p(x|\theta)$ with the corresponding approximation in (13) and using an improper uniform prior on $(-\infty, \infty)$. Results for $\theta_{\text{true}} = 0.4$, 0.6 and 0.8 are shown in Figure 2. Figure 3 shows corresponding lower and upper bounds for the log-likelihood function. As clearly $\max_\theta p(x|\theta)$ must at least be as large as the maximum of the lower bound (indicated by horizontal dashed lines in the figure), one can from lower and upper bounds for the likelihood function easily identify an interval which must contain the maximum likelihood estimator.

7 Closing remarks

We have discussed approximations and bounds for binary MRFs. The approximations and bounds are valid for a MRF with any neighbourhood structure, and can in particular be used for MRFs with a local neighbourhood system defined on a rectangular lattice. The results can be generalised to a situation with more than two possible values in each node, but this require the introduction of a more general notation. Moreover, we expect the computational resources necessary to obtain reasonably accurate approximation results to grow rapidly with the number of possible values in each node.

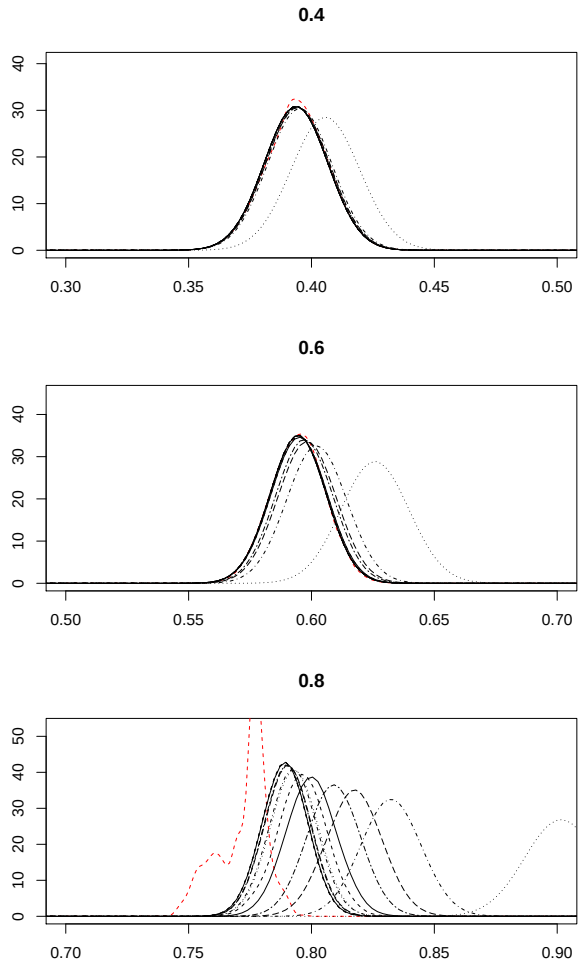

Fig. 2 The dashed curves are approximate posterior distributions for θ for $\nu = 2$ up to 13, moving from left to right for increasing value of ν. The dashed red curve is the results for an alternative approximation defined in Tjelmeland and Austad (2012). The upper, middle and lower plots show results for $\theta_{\text{true}} = 0.4$, 0.6 and 0.8, respectively.

In Section 6 we show some approximation results for an Ising model on a 100×100 lattice. Other demonstrations on how the approximations and bounds can be used are given in Austad and Tjelmeland (2011) and in Toftaker and Tjelmeland (2012).

Acknowledgements We acknowledge support from The Research Council of Norway, from Statoil and from ENI.

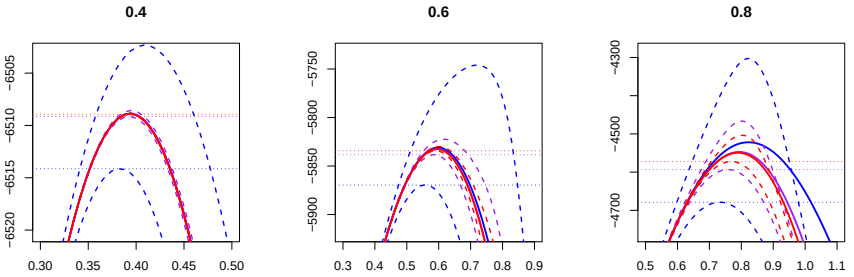

Fig. 3 The solid curves are approximate log-likelihood functions for $\nu = 6$ (blue), $\nu = 10$ (purple) and $\nu = 13$ (red). The corresponding dashed curves are corresponding lower and upper bounds. The left, middle and right plots show results for $\theta_{\text{true}} = 0.4$, 0.6 and 0.8, respectively.

References

Austad, H. M. (2011). *Approximations of Binary Markov random fields*, PhD 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.

Austad, H. and Tjelmeland, H. (2011). An approximate forward-backward algorithm applied to binary Markov random fields, *Technical report 11/2011*, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway. Available from http://www.math.ntnu.no/preprint/statistics/.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems, *Journal of the Royal Statistical Society, Series B* **36**: 192–225.

Clifford, P. (1990). Markov random fields in statistics, *in* G. R. Grimmett and D. J. A. Welsh (eds), *Disorder in Physical Systems*, Oxford University Press, pp. 19–31.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. and Spiegelhalter, D. J. (2007). *Probabilistic Networks and Expert Systems, Exact Computational Methods for Bayesian Networks*, Springer, London.

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, *Journal of Computational and Graphical Statistics* **18**: 243–261.

Friel, N. and Rue, H. (2007). Recursive computing and simulation-free inference for general factorizable models, *Biometrika* **94**: 661–672.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: from importance sampling to bridge sampling to path sampling, *Statistical Science* **13**: 163–185.

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference, *Journal of American Statistical Association* **90**: 909–920.

Grabisch, M., Marichal, J. L. and Roubens, M. (2000). Equivalent representations of set functions, *Mathematics of Operations Research* **25**: 157–178.

Gu, M. G. and Zhu, H. T. (2001). Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation, *Journal of the Royal Statistical Society, Series B* **63**: 339–355.

Hammer, P. L. and Holzman, R. (1992). Approximations of pseudo-Boolean functions; applications to game theory, *Methods and Models of Operation Research* **36**: 3–21.

Hammer, P. L. and Rudeanu, S. (1968). *Boolean Methods in Operation Research and Related Areas*, Springer, Berlin.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. and Saul, L. K. (1999). An introduction to variational methods for graphical models, *Machine Learning* **37**: 183–233.

Mateescu, R., Kask, K., Gogate, V. and Dechter, R. (2010). Join-graph propagation algorithms, *Journal of Artificial Intelligence Research* **37**: 279–328.

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.

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

Reeves, R. and Pettitt, A. N. (2004). Efficient recursions for general factorisable models, *Biometrika* **91**: 751–757.

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

Toftaker, H. and Tjelmeland, H. (2012). Construction of binary multi-grid markov random field prior models from training images, *Technical report 1/2012*, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway. Available from http://www.math.ntnu.no/preprint/statistics/.