Correctness of belief propagation in Gaussian graphical models of arbitrary topology

Yair Weiss
Computer Science Division
UC Berkeley, 485 Soda Hall
Berkeley, CA 94720-1776
Phone: 510-642-5029
yweiss@cs.berkeley.edu

William T. Freeman
Mitsubishi Electric Research Lab
201 Broadway
Cambridge, MA 02139
Phone: 617-621-7527
freeman@merl.com

Abstract

Local "belief propagation" rules of the sort proposed by Pearl [15] are guaranteed to converge to the correct posterior probabilities in singly connected graphical models. Recently, a number of researchers have empirically demonstrated good performance of "loopy belief propagation"– using these same rules on graphs with loops. Perhaps the most dramatic instance is the near Shannon-limit performance of "Turbo codes", whose decoding algorithm is equivalent to loopy belief propagation.

Except for the case of graphs with a single loop, there has been little theoretical understanding of the performance of loopy propagation. Here we analyze belief propagation in networks with arbitrary topologies when the nodes in the graph describe jointly Gaussian random variables. We give an analytical formula relating the true posterior probabilities with those calculated using loopy propagation. We give sufficient conditions for convergence and show that when belief propagation converges it gives the correct posterior means *for all graph topologies*, not just networks with a single loop.

The related "max-product" belief propagation algorithm finds the maximum posterior probability estimate for singly connected networks. We show that, even for non-Gaussian probability distributions, the convergence points of the max-product algorithm in loopy networks are maxima over a particular large local neighborhood of the posterior probability. These results help clarify the empirical performance results and motivate using the powerful belief propagation algorithm in a broader class of networks.

Problems involving probabilistic belief propagation arise in a wide variety of applications, including error correcting codes, speech recognition and medical diagnosis. If the graph is singly connected, there exist local message-passing schemes to calculate the posterior probability of an unobserved variable given the observed variables. Pearl [15] derived such a scheme for singly connected Bayesian networks and showed that this "belief propagation" algorithm is guaranteed to converge to the correct posterior probabilities (or "beliefs").

Several groups have recently reported excellent experimental results by running algorithms

equivalent to Pearl's algorithm on networks with loops [8, 13, 6]. Perhaps the most dramatic instance of this performance is for "Turbo code" [2] error correcting codes. These codes have been described as "the most exciting and potentially important development in coding theory in many years" [12] and have recently been shown [10, 11] to utilize an algorithm equivalent to belief propagation in a network with loops.

Progress in the analysis of loopy belief propagation has been made for the case of networks with a single loop [17, 18, 4, 1]. For these networks, it can be shown that (1) unless all the compatabilities are deterministic, loopy belief propagation will converge. (2) The difference between the loopy beliefs and the true beliefs is related to the convergence rate of the messages — the faster the convergence the more exact the approximation and (3) If the hidden nodes are binary, then the loopy beliefs and the true beliefs are both maximized by the same assignments, although the confidence in that assignment is wrong for the loopy beliefs.

In this paper we analyze belief propagation in graphs of *arbitrary topology*, for nodes describing jointly Gaussian random variables. We give an exact formula relating the correct marginal posterior probabilities with the ones calculated using loopy belief propagation. We show that if belief propagation converges, then it will give the correct posterior means *for all graph topologies*, not just networks with a single loop. We show that the covariance estimates will generally be incorrect but present a relationship between the error in the covariance estimates and the convergence speed. For Gaussian *or* non-Gaussian variables, we show that the "max-product" algorithm, which calculates the MAP estimate in singly connected networks, only converges to points that are maxima over a particular large neighborhood of the posterior probability of loopy networks.

1 Analysis

To simplify the notation, we assume the graphical model has been preprocessed into an undirected graphical model with pairwise potentials. Any graphical model can be converted into this form, and running belief propagation on the pairwise graph is equivalent to running belief propagation on the original graph [18]. We assume each node x_i has a local observation y_i. In each iteration of belief propagation, each node x_i sends a message to each neighboring x_j that is based on the messages it received from the other neighbors, its local observation y_i and the pairwise potentials $\Psi_{ij}(x_i, x_j)$ and $\Psi_{ii}(x_i, y_i)$. We assume the message-passing occurs in parallel.

The idea behind the analysis is to build an unwrapped tree. The unwrapped tree is the graphical model which belief propagation is solving exactly when one applies the belief propagation rules in a loopy network [9, 20, 18]. It is constructed by maintaining the same local neighborhood structure as the loopy network but nodes are replicated so there are no loops. The potentials and the observations are replicated from the loopy graph. Figure 1 (a) shows an unwrapped tree for the diamond shaped graph in (b). By construction, the belief at the root node $\tilde{x_1}$ is identical to that at node x_1 in the loopy graph after four iterations of belief propagation. Each node has a shaded observed node attached to it, omitted here for clarity.

Because the original network represents jointly Gaussian variables, so will the unwrapped tree. Since it is a tree, belief propagation is guaranteed to give the correct answer for the unwrapped graph. We can thus use Gaussian marginalization formulae to calculate the true mean and variances in both the original and the unwrapped networks. In this way, we calculate the accuracy of belief propagation for Gaussian networks of arbitrary topology.

We assume that the joint mean is zero (the means can be added-in later). The joint distri-

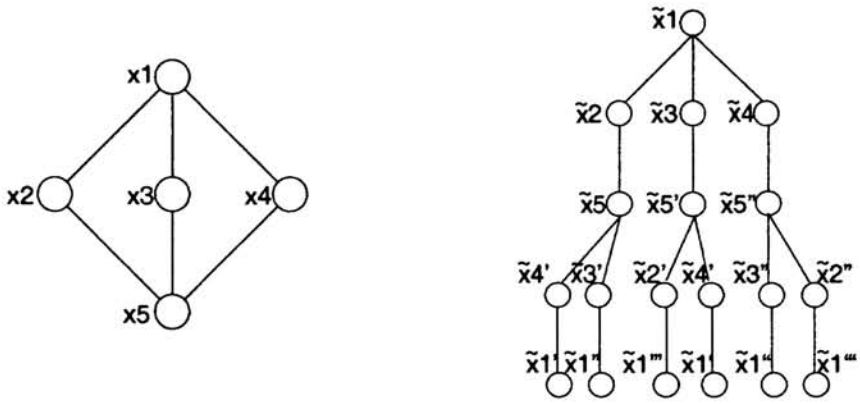


Figure 1: **Left:** A Markov network with multiple loops. **Right:** The unwrapped network corresponding to this structure.

bution of $z = \begin{pmatrix} x \\ y \end{pmatrix}$ is given by $P(z) = \alpha e^{-\frac{1}{2}z^T V z}$, where $V = \begin{pmatrix} V_{xx} & V_{xy} \\ V_{yx} & V_{yy} \end{pmatrix}$. It is straightforward to construct the inverse covariance matrix V of the joint Gaussian that describes a given Gaussian graphical model [3].

Writing out the exponent of the joint and completing the square shows that the mean μ of x, given the observations y, is given by:

$$V_{xx}\mu = -V_{xy}y, \tag{1}$$

and the covariance matrix $C_{x|y}$ of x given y is: $C_{x|y} = V_{xx}^{-1}$. We will denote by $C_{x_i|y}$ the ith row of $C_{x|y}$ so the marginal posterior variance of x_i given the data is $\sigma^2(i) = C_{x_i|y}(i)$.

We will use ˜ for unwrapped quantities. We scan the tree in *breadth first* order and denote by $\tilde{x}$ the vector of values in the hidden nodes of the tree when so scanned. Simlarly, we denote by $\tilde{y}$ the observed nodes scanned in the same order and $\tilde{V}_{xx}, \tilde{V}_{xy}$ the inverse covariance matrices. Since we are scanning in breadth first order the last nodes are the leaf nodes and we denote by L the number of leaf nodes. By the nature of unwrapping, $\tilde{\mu}(1)$ is the mean of the belief at node x_1 after t iterations of belief propagation, where t is the number of unwrappings. Similarly $\tilde{\sigma}^2(1) = \tilde{C}_{x_1|y}(1)$ is the variance of the belief at node x_1 after t iterations.

Because the data is replicated we can write $\tilde{y} = Oy$ where $O(i,j) = 1$ if $\tilde{y}_i$ is a replica of y_j and 0 otherwise. Since the potentials $\Psi(x_i, y_i)$ are replicated, we can write $\tilde{V}_{xy}O = OV_{xy}$. Since the $\Psi(x_i, x_j)$ are also replicated and all non-leaf $\tilde{x}_i$ have the same connectivity as the corresponding x_i, we can write $\tilde{V}_{xx}O = OV_{xx} + E$ where E is zero in all but the last L rows. When these relationships between the loopy and unwrapped inverse covariance matrices are substituted into the loopy and unwrapped versions of equation 1, one obtains the following expression, true for any iteration [19]:

$$\tilde{\mu}(1) = \mu(1) + \tilde{C}_{x_1|y}e \tag{2}$$

where e is a vector that is zero everywhere but the last L components (corresponding to the leaf nodes). Our choice of the node for the root of the tree is arbitrary, so this applies to all nodes of the loopy network. This formula relates, for any node of a network with loops, the means calculated at each iteration by belief propagation with the true posterior means.

Similarly when the relationship between the loopy and unwrapped inverse covariance matrices is substituted into the loopy and unwrapped definitions of $C_{x|y}$ we can relate the

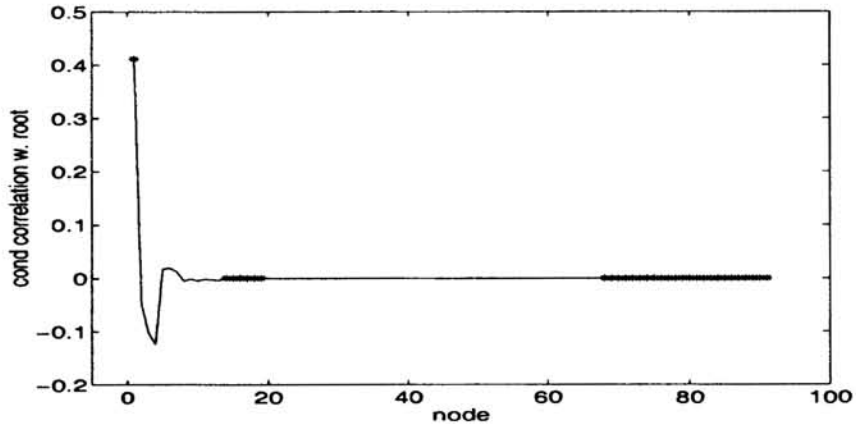


Figure 2: The conditional correlation between the root node and all other nodes in the unwrapped tree of Fig. 1 after eight iterations. Potentials were chosen randomly. Nodes are presented in breadth first order so the last elements are the correlations between the root node and the leaf nodes. We show that if this correlation goes to zero, belief propagation converges and the loopy means are exact. Symbols plotted with a star denote correlations with nodes that correspond to the node x_1 in the loopy graph. The sum of these correlations gives the correct variance of node x_1 while loopy propagation uses only the first correlation.

marginalized covariances calculated by belief propagation to the true ones [19]:

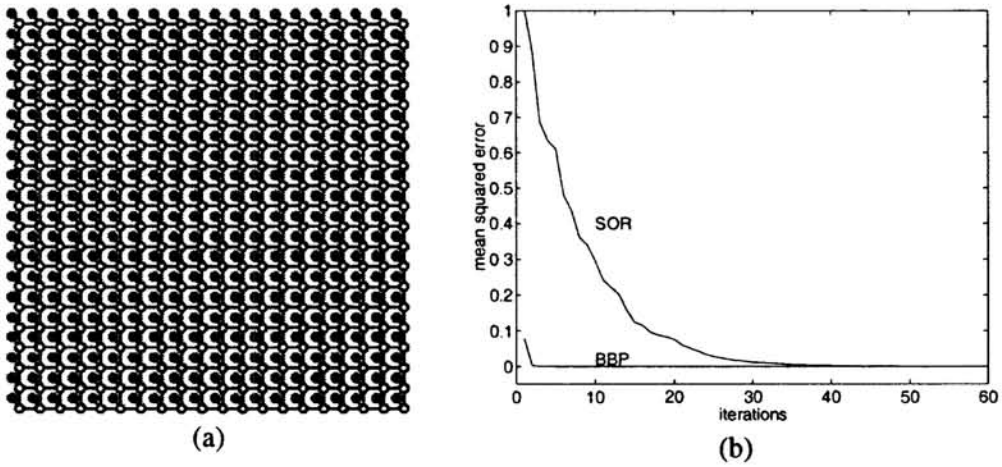
$$\tilde{\sigma}^2(1) = \sigma^2(1) + \tilde{C}_{x_1|y} e_1 - \tilde{C}_{x_1|y} e_2 \tag{3}$$

where e_1 is a vector that is zero everywhere but the last L components while e_2 is equal to 1 for all nodes in the unwrapped tree that are replicas of x_1 except for $\tilde{x}_1$. All other components of e_2 are zero,

Figure 2 shows $\tilde{C}_{x_1|y}$ for the diamond network in Fig. 1. We generated random potential functions and observations and calculated the conditional correlations in the unwrapped tree. Note that the conditional correlation decreases with distance in the tree — we are scanning in breadth first order so the last L components correspond to the leaf nodes. As the number of iterations of loopy propagation is increased the size of the unwrapped tree increases and the conditional correlation between the leaf nodes and the root node decreases.

From equations 2–3 it is clear that if the conditional correlation between the leaf nodes and the root nodes are zero for all sufficiently large unwrappings then (1) belief propagation converges (2) the means are exact and (3) the variances may be incorrect. In practice the conditional correlations will not actually be equal to zero for any finite unwrapping. In [19] we give a more precise statement: if the conditional correlation of the root node and the leaf nodes decreases rapidly enough then (1) belief propagation converges (2) the means are exact and (3) the variances may be incorrect. We also show sufficient conditions on the potentials $\Psi(x_i, x_j)$ for the correlation to decrease rapidly enough: the rate at which the correlation decreases is determined by the ratio of off-diagonal and diagonal components in the quadratic form defining the potentials [19].

How wrong will the variances be? The term $\tilde{C}_{x_1|y} e_2$ in equation 3 is simply the sum of many components of $\tilde{C}_{x_1|y}$. Figure 2 shows these components. The correct variance is the sum of all the components while the belief propagation variance approximates this sum with the first (and dominant) term. Whenever there is a positive correlation between the root node and other replicas of x_1 the loopy variance is strictly less than the true variance — the loopy estimate is overconfident.



(a) (b)

Figure 3: (a) 25×25 graphical model for simulation. The unobserved nodes (unfilled) were connected to their four nearest neighbors and to an observation node (filled). (b) The error of the estimates of loopy propagation and successive over-relaxation (SOR) as a function of iteration. Note that belief propagation converges much faster than SOR.

Note that when the conditional correlation decreases rapidly to zero two things happen. First, the convergence is faster (because $\tilde{C}_{x_1|y}e_1$ approaches zero faster). Second, the approximation error of the variances is smaller (because $\tilde{C}_{x_1|y}e_2$ is smaller). Thus we have shown, as in the single loop case, quick convergence is correlated with good approximation.

2 Simulations

We ran belief propagation on the 25×25 $2D$ grid of Fig. 3 a. The joint probability was:

$$P(x,y) = exp(-\sum_{ij} w_{ij}(x_i - x_j)^2 - \sum_{i} w_{ii}(x_i - y_i)^2) \tag{4}$$

where $w_{ij} = 0$ if nodes x_i, x_j are not neighbors and 0.01 otherwise and w_{ii} was randomly selected to be 0 or 1 for all i with probability of 1 set to 0.2. The observations y_i were chosen randomly. This problem corresponds to an approximation problem from sparse data where only 20% of the points are visible.

We found the exact posterior by solving equation 1. We also ran belief propagation and found that when it converged, the calculated means were identical to the true means up to machine precision. Also, as predicted by the theory, the calculated variances were too small — the belief propagation estimate was overconfident.

In many applications, the solution of equation 1 by matrix inversion is intractable and iterative methods are used. Figure 3 compares the error in the means as a function of iterations for loopy propagation and successive-over-relaxation (SOR), considered one of the best relaxation methods [16]. Note that after essentially five iterations loopy propagation gives the right answer while SOR requires many more. As expected by the fast convergence, the approximation error in the variances was quite small. The median error was 0.018. For comparison the true variances ranged from 0.01 to 0.94 with a mean of 0.322. Also, the nodes for which the approximation error was worse were indeed the nodes that converged slower.

3 Discussion

Independently, two other groups have recently analyzed special cases of Gaussian graphical models. Frey [7] analyzed the graphical model corresponding to factor analysis and gave conditions for the existence of a stable fixed-point. Rusmevichientong and Van Roy [14] analyzed a graphical model with the topology of turbo decoding but a Gaussian joint density. For this specific graph they gave sufficient conditions for convergence and showed that the means are exact.

Our main interest in the Gaussian case is to understand the performance of belief propagation in general networks with multiple loops. We are struck by the similarity of our results for Gaussians in arbitrary networks and the results for single loops of arbitrary distributions [18]. First, in single loop networks with binary nodes, loopy belief at a node and the true belief at a node are maximized by the same assignment while the confidence in that assignment is incorrect. In Gaussian networks with multiple loops, the mean at each node is correct but the confidence around that mean may be incorrect. Second, for both single-loop and Gaussian networks, fast belief propagation convergence correlates with accurate beliefs. Third, in both Gaussians and discrete valued single loop networks, the statistical dependence between root and leaf nodes governs the convergence rate and accuracy.

The two models are quite different. Mean field approximations are exact for Gaussian MRFs while they work poorly in sparsely connected discrete networks with a single loop. The results for the Gaussian and single-loop cases lead us to believe that similar results may hold for a larger class of networks.

Can our analysis be extended to non-Gaussian distributions? The basic idea applies to arbitrary graphs and arbitrary potentials: belief propagation is performing exact inference on a tree that has the same local neighbor structure as the loopy graph. However, the linear algebra that we used to calculate exact expressions for the error in belief propagation at any iteration holds only for Gaussian variables.

We have used a similar approach to analyze the related "max-product" belief propagation algorithm on arbitrary graphs with arbitrary distributions [5] (both discrete and continuous valued nodes). We show that if the max-product algorithm converges, the max-product assignment has greater posterior probability then any assignment in a particular large region around that assignment. While this is a weaker condition than a global maximum, it is much stronger than a simple local maximum of the posterior probability.

The sum-product and max-product belief propagation algorithms are fast and parallelizable. Due to the well known hardness of probabilistic inference in graphical models, belief propagation will obviously not work for arbitrary networks and distributions. Nevertheless, a growing body of empirical evidence shows its success in many networks with loops. Our results justify applying belief propagation in certain networks with multiple loops. This may enable fast, approximate probabilistic inference in a range of new applications.

References

[1] S.M. Aji, G.B. Horn, and R.J. McEliece. On the convergence of iterative decoding on graphs with a single cycle. In *Proc. 1998 ISIT*, 1998.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes. In *Proc. IEEE International Communications Conference '93*, 1993.

[3] R. Cowell. Advanced inference in Bayesian networks. In M.I. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1998.

[4] G.D. Forney, F.R. Kschischang, and B. Marcus. Iterative decoding of tail-biting trellisses. preprint presented at 1998 Information Theory Workshop in San Diego, 1998.

[5] W. T. Freeman and Y. Weiss. On the fixed points of the max-product algorithm. Technical Report 99–39, MERL, 201 Broadway, Cambridge, MA 02139, 1999.

[6] W.T. Freeman and E.C. Pasztor. Learning to estimate scenes from images. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Adv. Neural Information Processing Systems 11*. MIT Press, 1999.

[7] B.J. Frey. Turbo factor analysis. In *Adv. Neural Information Processing Systems 12*. 2000. to appear.

[8] Brendan J. Frey. *Bayesian Networks for Pattern Classification, Data Compression and Channel Coding*. MIT Press, 1998.

[9] R.G. Gallager. *Low Density Parity Check Codes*. MIT Press, 1963.

[10] F. R. Kschischang and B. J. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal on Selected Areas in Communication*, 16(2):219–230, 1998.

[11] R.J. McEliece, D.J.C. MackKay, and J.F. Cheng. Turbo decoding as as an instance of Pearl's 'belief propagation' algorithm. *IEEE Journal on Selected Areas in Communication*, 16(2):140–152, 1998.

[12] R.J. McEliece, E. Rodemich, and J.F. Cheng. The Turbo decision algorithm. In *Proc. 33rd Allerton Conference on Communications, Control and Computing*, pages 366–379, Monticello, IL, 1995.

[13] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of Uncertainty in AI*, 1999.

[14] Rusmevichientong P. and Van Roy B. An analysis of Turbo decoding with Gaussian densities. In *Adv. Neural Information Processing Systems 12*. 2000. to appear.

[15] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[16] Gilbert Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge, 1986.

[17] Y. Weiss. Belief propagation and revision in networks with loops. Technical Report 1616, MIT AI lab, 1997.

[18] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, to appear, 2000.

[19] Y. Weiss and W. T. Freeman. Loopy propagation gives the correct posterior means for Gaussians. Technical Report UCB.CSD-99-1046, Berkeley Computer Science Dept., 1999. www.cs.berkeley.edu yweiss/.

[20] N. Wiberg. *Codes and decoding on general graphs*. PhD thesis, Department of Electrical Engineering, U. Linkoping, Sweden, 1996.

Gaussian Fields for Approximate Inference in Layered Sigmoid Belief Networks

David Barber*
Stichting Neurale Netwerken
Medical Physics and Biophysics
Nijmegen University, The Netherlands
barberd@aston.ac.uk

Peter Sollich
Department of Mathematics
King's College, University of London
London WC2R 2LS, U.K.
peter.sollich@kcl.ac.uk

Abstract

Layered Sigmoid Belief Networks are directed graphical models in which the local conditional probabilities are parameterised by weighted sums of parental states. Learning and inference in such networks are generally intractable, and approximations need to be considered. Progress in *learning* these networks has been made by using variational procedures. We demonstrate, however, that variational procedures can be inappropriate for the equally important issue of *inference* - that is, calculating marginals of the network. We introduce an alternative procedure, based on assuming that the weighted input to a node is approximately Gaussian distributed. Our approach goes beyond previous Gaussian field assumptions in that we take into account correlations between parents of nodes. This procedure is specialized for calculating marginals and is significantly faster and simpler than the variational procedure.

1 Introduction

Layered Sigmoid Belief Networks [1] are directed graphical models [2] in which the local conditional probabilities are parameterised by weighted sums of parental states, see fig(1). This is a graphical representation of a distribution over a set of binary variables $s_i \in \{0, 1\}$. Typically, one supposes that the states of the nodes at the bottom of the network are *generated* by states in previous layers. Whilst, in principle, there is no restriction on the number of nodes in any layer, typically, one considers structures similar to the "fan out" in fig(1) in which higher level layers provide an "explanation" for patterns generated in lower layers. Such graphical models are attractive since they correspond to layers of information processors, of potentially increasing complexity. Unfortunately, learning and inference in such networks is generally intractable, and approximations need to be considered. Progress in learning has been made by using variational procedures [3, 4, 5]. However, another crucial aspect remains inference [2]. That is, given some evidence (or none), calculate the marginal of a variable, conditional on this evidence. This assumes that we have found a suitable network from some learning procedure, and now wish

*Present Address: NCRG, Aston University, Birmingham B4 7ET, U.K.

to query this network. Whilst the variational procedure is attractive for learning, since it generally provides a bound on the likelihood of the visible units, we demonstrate that it may not always be equally appropriate for the inference problem.

A directed graphical model defines a distribution over a set of variables $\mathbf{s} = (s_1 \dots s_n)$ that factorises into the local conditional distributions,

$$p(s_1 \dots s_n) = \prod_{i=1}^{n} p(s_i|\pi_i) \tag{1}$$

where π_i denotes the parent nodes of node i. In a layered network, these are the nodes in the proceeding layer that feed into node i. In a sigmoid belief network the local probabilities are defined as

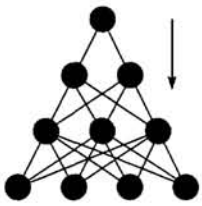


Figure 1: A Layered Sigmoid Belief Network

$$p(s_i = 1|\pi_i) = \sigma\left(\sum_j w_{ij}s_j + \theta_i\right) = \sigma(h_i) \tag{2}$$

where the "field" at node i is defined as $h_i = \sum_j w_{ij}s_j + \theta_i$ and $\sigma(h) = 1/(1+e^{-h})$. w_{ij} is the strength of the connection between node i and its parent node j; if j is not a parent of i we set $w_{ij} = 0$. θ_i is a bias term that gives a parent-independent bias to the state of node i.

We are interested in inference - in particular, calculating marginals of the network for cases with and without evidential nodes. In section (2) we describe how to approximate the quantities $p(s_i = 1)$ and discuss in section (2.1) why our method can improve on the standard variational mean field theory. Conditional marginals, such as $p(s_i = 1|s_j = 1, s_k = 0)$ are considered in section (3).

2 Gaussian Field Distributions

Under the 0/1 coding for the variables s_i, the mean of a variable, m_i is given by the probability that it is in state 1. Using the fact from (2) that the local conditional distribution of node i is dependent on its parents *only* through its field h_i, we have

$$m_i = p(s_i = 1) = \int p(s_i = 1|h_i)\, p(h_i)\, dh_i \equiv \langle \sigma(h_i) \rangle_{p(h_i)} \tag{3}$$

where we use the notation $\langle(\cdot)\rangle_p$ to denote an average with respect to the distribution p. If there are many parents of node i, a reasonable assumption is that the distribution of the field h_i will be Gaussian, $p(h_i) \approx N(\mu_i, \sigma_i^2)$. Under this Gaussian Field (GF) assumption, we need to work out the mean and variance, which are given by

$$\mu_i = \langle h_i \rangle = \sum_j w_{ij} \langle s_j \rangle + \theta_i = \sum_j w_{ij} m_j + \theta_i \tag{4}$$

$$\sigma_i^2 = \left\langle (\Delta h_i)^2 \right\rangle = \sum_{j,k} w_{ij} w_{ik} R_{jk} \tag{5}$$

where $R_{jk} = \langle \Delta s_j \Delta s_k \rangle$. We use the notation $\Delta(\cdot) \equiv (\cdot) - \langle(\cdot)\rangle$.

The diagonal terms of the node covariance matrix are $R_{ii} = m_i(1 - m_i)$. In contrast to previous studies, we include off diagonal terms in the calculation of R [4]. From

(5) we only need to find correlations between parents i and j of a node. These are easy to calculate in the layered networks that we are considering, because neither i nor j is a descendant of the other:

$$R_{ij} = p(s_i = 1, s_j = 1) - m_i m_j \tag{6}$$

$$= \int p(s_i = 1|h_i)p(s_j = 1|h_j)p(h_i, h_j)dh - m_i m_j \tag{7}$$

$$= \langle \sigma(h_i)\, \sigma(h_j) \rangle_{p(h_i,h_j)} - m_i m_j \tag{8}$$

Assuming that the joint distribution $p(h_i, h_j)$ is Gaussian, we again need its mean and covariance, given by

$$\boldsymbol{\mu}^T = (\langle h_i \rangle, \langle h_j \rangle) = \left(\sum_k w_{ik} m_k + \theta_i, \sum_l w_{jl} m_l + \theta_j \right) \tag{9}$$

$$\Sigma_{ij} = \langle \Delta h_i \Delta h_j \rangle = \sum_{kl} w_{ik} w_{jl} \langle \Delta s_k \Delta s_l \rangle = \sum_{kl} w_{ik} w_{jl} R_{kl} \tag{10}$$

Under this scheme, we have a closed set of equations, (4,5,8,10) for the means m_i and covariance matrix R_{ij} which can be solved by forward propagation of the equations. That is, we start from nodes without parents, and then consider the next layer of nodes, repeating the procedure until a full sweep through the network has been completed. The one and two dimensional field averages, equations (3) and (8), are computed using Gaussian Quadrature. This results in an extremely fast procedure for approximating the marginals m_i, requiring only a single sweep through the network.

Our approach is related to that of [6] by the common motivating assumption that each node has a large number of parents. This is used in [6] to obtain actual bounds on quantities of interest such as joint marginals. Our approach does not give bounds. Its advantage, however, is that it allows fluctuations in the fields h_i, which are effectively excluded in [6] by the assumed scaling of the weights w_{ij} with the number of parents per node.

2.1 Relation to Variational Mean Field Theory

In the variational approach, one fits a tractable approximating distribution Q to the SBN. Taking Q factorised, $Q(\mathbf{s}) = \prod_i m_i^{s_i}(1 - m_i)^{1-s_i}$ we have the bound

$$\ln p(s_1 \ldots s_n) \geq \sum_i \{-m_i \ln m_i - (1 - m_i) \ln (1 - m_i)\}$$

$$+ \sum_i \left\{ \sum_j m_i w_{ij} m_j + \theta_i m_i - \left\langle \ln \left(1 + e^{h_i}\right) \right\rangle_Q \right\} \tag{11}$$

The final term in (11) causes some difficulty even in the case in which Q is a factorised model. Formally, this is because this term does not have the same graphical structure as the tractable model Q. One way around around this difficulty is to employ a further bound, with associated variational parameters [7]. Another approach is to make the Gaussian assumption for the field h_i as in section (2). Because Q is factorised, corresponding to a diagonal correlation matrix R, this gives [4]

$$\left\langle \ln \left(1 + e^{h_i}\right) \right\rangle_Q \approx \left\langle \ln \left(1 + e^{h_i}\right) \right\rangle_{N(\mu_i, \sigma_i^2)} \tag{12}$$

where $\mu_i = \sum_j w_{ij} m_j + \theta_i$ and $\sigma_i^2 = \sum_j w_{ij}^2 m_j (1 - m_j)$. Note that this is a one dimensional integral of a smooth function. In contrast to [4] we therefore evaluate this quantity using Gaussian Quadrature. This has the advantage that no extra variational parameters need to be introduced. Technically, the assumption of a Gaussian field distribution means that (11) is no longer a bound. Nevertheless, in practice it is found that this has little effect on the quality of the resulting solution. In our implementation of the variational approach, we find the optimal parameters m_i by maximising the above equation for each component m_i separately, cycling through the nodes until the parameters m_i do not change by more than 10^{-10}. This is repeated 5 times, and the solution with the highest bound score is chosen. Note that these equations cannot be solved by forward propagation alone since the final term contains contributions from all the nodes in the network. This is in contrast to the GF approach of section (2). Finding appropriate parameters m_i by the variational approach is therefore rather slower than using the GF method.

In arriving at the above equations, we have made two assumptions. The first is that the intractable distribution is well approximated by a factorised model. The second is that the field distribution is Gaussian. The first step is necessary in order to obtain a bound on the likelihood of the model (although this is slightly compromised by the Gaussian field assumption). In the GF approach we dispense with this assumption of an effectively factorised network (partially because if we are only interested in inference, a bound on the model likelihood is less relevant). The GF method may therefore prove useful for a broader class of networks than the variational approach.

2.2 Results for unconditional marginals

We compared three procedures for estimating the conditional values $p(s_i = 1)$ for all the nodes in the network, namely the variational theory, as described in section (2.1), the diagonal Gaussian field theory, and the non-diagonal Gaussian field theory which includes correlation effects between parents. Results for small weight values w_{ij} are shown in fig(2). In this case, all three methods perform reasonably well, although there is a significant improvement in using the GF methods over the variational procedure; parental correlations are not important (compare figs(2b) and (2c)). In fig(3) the weights and biases are chosen such that the exact mean variables m_i are roughly 0.5 with non-trivial correlation effects between parents. Note that the variational mean field theory now provides a poor solution, whereas the GF methods are relatively accurate. The effect of using the non-diagonal R terms is beneficial, although not dramatically so.

3 Calculating Conditional Marginals

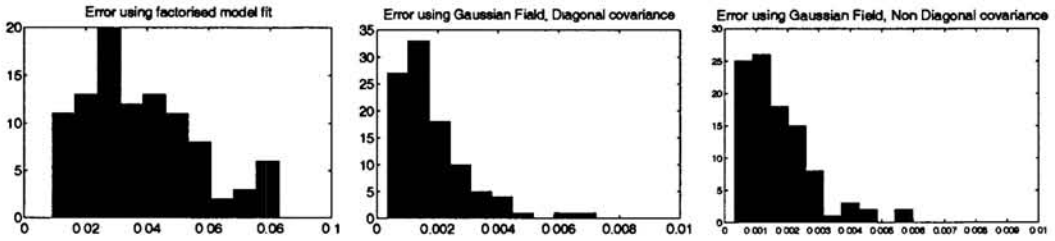
We consider now how to calculate conditional marginals, given some evidential nodes. (In contrast to [6], any set of nodes in the network, not just output nodes, can be considered evidential.) We write the evidence in the following manner

$$E = \{s_{c_1} = S_{c_1}, \ldots s_{c_n} = S_{c_n}\} = \{E_{c_1} \ldots E_{c_n}\}$$

The quantities that we are interested in are conditional marginals which, from Bayes rule are related to the joint distribution by

$$p(s_i = 1|E) = \frac{p(s_i = 1, E)}{p(s_i = 0, E) + p(s_i = 1, E)} \tag{13}$$

That is, provided that we have a procedure for estimating joint marginals, we can obtain conditional marginals too. Without loss of generality, we therefore consider



(a) Mean error = 0.0377 (b) Mean error = 0.0018 (c) Mean error = 0.0017

Figure 2: Error in approximating $p(s_i = 1)$ for the network in fig(1), averaged over all the nodes in the network. In each of 100 trials, weights were drawn from a zero mean, unit variance Gaussian; biases were set to 0. Note the different scale in (b) and (c). In (a) we use the variational procedure with a factorised Q, as in section (2.1). In (b) we use the Gaussian field equations, assuming a diagonal covariance matrix R. This procedure was repeated in (c) including correlations between parents.

$E^+ = E \cup \{s_i = 1\}$, which then contains $n+1$ "evidential" variables. That is, the desired marginal variable is absorbed into the evidence set. For convenience, we then split the nodes into two sets, those containing the evidential or "clamped" nodes, C, and the remaining "free" nodes F. The joint evidence is then given by

$$p(E^+) = \sum_{s_F} p\left(E_{c_1}, \ldots E_{c_{n+1}}, s_{f_1}, \ldots s_{f_m}\right) \tag{14}$$

$$= \sum_{s_F} p\left(E_{c_1}|\pi^*_{c_1}\right) \ldots p\left(E_{n+1}|\pi^*_{c_{n+1}}\right) p\left(s_{f_1}|\pi^*_{f_1}\right) \ldots p\left(s_{f_m}|\pi^*_{f_m}\right) \tag{15}$$

where π^*_i are the parents of node i, with any evidential parental nodes set to their values as specified in E^+. In the sigmoid belief network

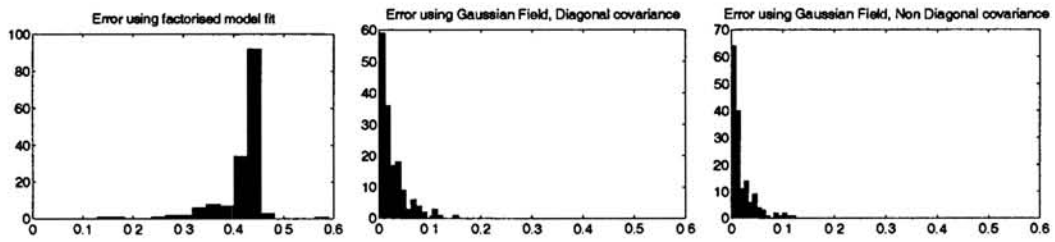
$$p(E_k|\pi^*_k) = \sigma\left((2S_k - 1)\left(\sum_i w_{ki} s^*_i + \theta_k\right)\right), \quad s^*_i = \begin{cases} S_i, & \text{if } i \text{ is an evidential node} \\ s_i, & \text{otherwise} \end{cases} \tag{16}$$

$p(E_k|\pi^*_k)$ is therefore determined by the distribution of the field $h^*_k = \sum_i w_{ki} s^*_i + \theta_k$. Examining (15), we see that the product over the "free" nodes defines a SBN in which the local probability distributions are given by those of the original network, but with any evidential parental nodes clamped to their evidence values. Therefore,

$$p\left(E^+\right) = \left\langle \prod_{i=1}^{n+1} \sigma\left((2S_{c_i} - 1)h^*_{c_i}\right) \right\rangle_{p\left(h^*_{c_1} \ldots h^*_{c_{n+1}}\right)} \tag{17}$$

Consistent with our previous assumptions, we assume that the distribution of the fields $\mathbf{h}^* = \left(h^*_{c_1} \ldots h^*_{c_{n+1}}\right)$ is jointly Gaussian. We can then find the mean and covariance matrix for the distribution of $\mathbf{h}^*$ by repeating the calculation of section (2) in which evidential nodes have been clamped to their evidence values. Once this Gaussian has been determined, it can be used in (17) to determine $p(E^+)$. Gaussian averages of products of sigmoids are calculated by drawing 1000 samples from the Gaussian over which we wish to integrate[1]. Note that if there are evidential nodes

[1] In one and two dimensions ($n = 0, 1$), or $n = 1$, we use Gaussian Quadrature.

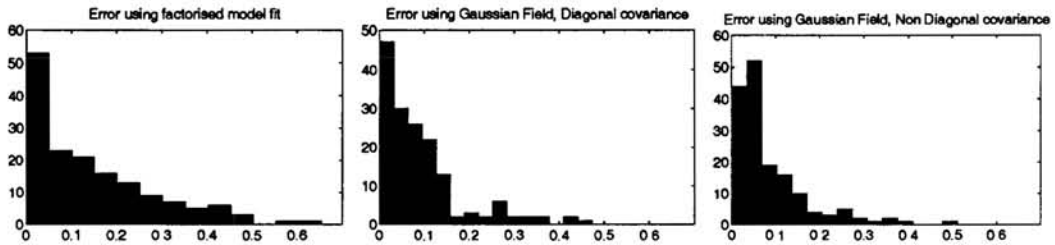


(a) Mean error = 0.4188 (b) Mean error = 0.0253 (c) Mean error = 0.0198

Figure 3: All weights are set to uniformly from 0 to 50. Biases are set to -0.5 of the summed parental weights plus a uniform random number from -2.5 to 2.5. The root node is set to be 1 with probability 0.5. This has the effect of making all the nodes in the exact network roughly 0.5 in mean, with non-negligible correlations between parental nodes. 160 simulations were made.

in different layers, we require the correlations between their fields h to evaluate (17). Such 'inter-layer' correlations were not required in section (2), and to be able to use the same calculational scheme we simply neglect them. (We leave a study of the effects of this assumption for future work.) The average in (17) then factors into groups, where each group contains evidential terms in a particular layer.

The conditional marginal for node i is obtained from repeating the above procedure in which the desired marginal node is clamped to its opposite value, and then using these results in (13). The above procedure is repeated for each conditional marginal that we are interested in. Although this may seem computationally expensive, the marginal for each node is computed quickly, since the equations are solved by one forward propagation sweep only.



(a) Mean error = 0.1534 (b) Mean error = 0.0931 (c) Mean error = 0.0865

Figure 4: Estimating the conditional marginal of the top node being in state 1, given that the four bottom nodes are in state 1. Weights were drawn from a zero mean Gaussian with variance 5, with biases set to -0.5 the summed parental weights plus a uniform random number from -2.5 to 2.5. Results of 160 simulations.

3.1 Results for conditional marginals

We used the same structure as in the previous experiments, as shown in fig(1). We are interested here in calculating the probability that the top node is in state 1,

given that the four bottom nodes are in state 1. Weights were chosen from a zero mean Gaussian with variance 5. Biases were set to negative half of the summed parent weights, plus a uniform random value from -2.5 to 2.5. Correlation effects in these networks are not as strong as in the experiments in section (2.2), although the improvement of the GF theory over the variational theory seen in fig(4) remains clear. The improvement from the off diagonal terms in R is minimal.

4 Conclusion

Despite their appropriateness for learning, variational methods may not be equally suited to inference, making more tailored methods attractive. We have considered an approximation procedure that is based on assuming that the distribution of the weighted input to a node is approximately Gaussian. Correlation effects between parents of a node were taken into account to improve the Gaussian theory, although in our examples this gave only relatively modest improvements.

The variational mean field theory performs poorly in networks with strong correlation effects between nodes. On the other hand, one may conjecture that the Gaussian Field approach will not generally perform catastrophically worse than the factorised variational mean field theory. One advantage of the variational theory is the presence of an objective function against which competing solutions can be compared. However, finding an optimum solution for the mean parameters m_i from this function is numerically complex. Since the Gaussian Field theory is extremely fast to solve, an interesting compromise might be to prime the variational solution with the results from the Gaussian Field theory.

Acknowledgments

DB would like to thank Bert Kappen and Wim Wiegerinck for stimulating and helpful discussions. PS thanks the Royal Society for financial support.

[1] R. Neal. Connectionist learning of Belief Networks. *Artificial Intelligence*, 56:71–113, 1992.

[2] E. Castillo, J. M. Gutierrez, and A. S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer, 1997.

[3] M. I. Jordan, Z. Gharamani, T. S. Jaakola, and L. K. Saul. An Introduction to Variational Methods for Graphical Models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 105–161. Kluwer, 1998.

[4] L. Saul and M. I. Jordan. A mean field learning algorithm for unsupervised neural networks. In M. I. Jordan, editor, *Learning in Graphical Models*, 1998.

[5] D. Barber and W Wiegerinck. Tractable variational structures for approximating graphical models. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems NIPS 11*. MIT Press, 1999.

[6] M. Kearns and L. Saul. Inference in Multilayer Networks via Large Deviation Bounds. In *Advances in Neural Information Processing Systems NIPS 11*, 1999.

[7] L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean Field Theory for Sigmoid Belief Networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.